



# **TRANSYT-7F™**

Traffic Network Study Tool



---

# USER GUIDE

# Contents

<b>Introduction</b>	<b>6</b>
License Agreement.....	6
Acknowledgements.....	9
Trademarks and Copyrights.....	9
 <b>Chapter 1: TRANSYT-7F Software Package</b>	 <b>11</b>
Scope and Qualities.....	11
History of TRANSYT-7F.....	12
TRANSYT-7F Graphical User Interface (GUI).....	16
Record Type Editor.....	18
HCS Import/Export.....	19
CORSIM Import/Export/Processing/Optimization.....	20
Formatted Report.....	23
Sample Data Files.....	25
Technical Support and Problem Reporting.....	26
 <b>Chapter 2: TRANSYT-7F Applications</b>	 <b>27</b>
Macroscopic Simulation of Traffic Flow.....	27
Traffic Signal Timing Elements and Characteristics.....	32
Traffic Signal Timing Optimization.....	34
Traffic-Actuated Control.....	43
Permitted Movements.....	46
Shared Lanes and Lane-by-Lane Analysis.....	48
Multi-Cycle and Multi-Period Simulation.....	49
Uncoordinated Operations.....	52
Sign Control.....	52
Mid-Block Sources.....	53
Pedestrians, Bicycles, and Exclusive Pedestrian Phases.....	53
Bus and Carpool Operations.....	54
Bottlenecks.....	55
 <b>Chapter 3: TRANSYT-7F Input Data</b>	 <b>57</b>
TRANSYT-7F Input Screens.....	57
Input Data Screen Overview.....	57
New TRANSYT Input File.....	59
Map View (associated with record types 70 and 71).....	62
Link Numbering Scheme.....	77
Lane Configuration Screen (associated with record types 28, 29, and 291).....	80
Traffic Screen (associated with record types 28 and 29).....	86
Timing View Screen (associated with record types 1X, 18, 2Y, and 2Z).....	91
Dual-Ring Timing Screen (associated with record types 1X, 18, 2Y, and 2Z).....	99
Satflow View Screen (associated with record types 28 and 292).....	106
Feeders Screen (associated with record type 28).....	110
Analysis Screen (associated with record types 1, 1X, 10, 5X, and 9X).....	112
Global Screen (associated with record types 1-10).....	120

Route Screen (associated with record types 40-43, and 60-61) .....	132
Timing Screen (associated with record types 1X, 18, 2Y, and 2Z).....	136
Queuing Screen (associated with record types 28 and 291).....	143
Actuated Screen (associated with record type 291) .....	144
Permitted Screen (associated with record type 29) .....	146
Weighting Screen (associated with record types 30-39) .....	150
Saturation Flow Rate Calculator (associated with record types 28 and 292) .....	152
Intersection Names Screen.....	155
Preferences Screen .....	156
Record Type Format .....	159
Record Type Format.....	159
Run Title Record .....	165
Record Type 0 .....	166
Record Type 1 .....	167
Record Type 2 .....	168
Record Type 3 .....	169
Record Type 4 .....	170
Record Type 5 .....	171
Record Type 6 .....	172
Record Type 7 .....	173
Record Type 8 .....	174
Record Type 10 .....	175
Record Type 1X .....	176
Record Type 18 .....	177
Record Type 2Y .....	178
Record Type 2Z .....	179
Record Type 28 .....	180
Record Type 29 .....	181
Record Type 291 .....	182
Record Type 292 .....	183
Record Type 30 .....	184
Record Type 31 .....	185
Record Type 32 .....	186
Record Type 33 .....	187
Record Type 34 .....	188
Record Type 35 .....	189
Record Type 36 .....	190
Record Type 37 .....	191
Record Type 38 .....	192
Record Type 39 .....	193
Record Type 40 .....	194
Record Type 41 .....	195
Record Type 42 .....	196
Record Type 43 .....	197
Record Type 5X .....	198
Record Type 60 .....	199
Route Title Record .....	200
Record Type 61 .....	201
Record Type 70 .....	202
Record Type 71 .....	203
Record Type 9X .....	204
Intersection Name Record .....	205
Link Name Record .....	206
Header Information Record.....	207

## **Chapter 4: TRANSYT-7F Model 209**

Actuated Estimation Methodology .....	209
Arrival Type .....	216
Artplan Import Methodology .....	217
Attainability .....	217
Automatic Flow Balancing .....	218
Average System Speed .....	225
Average Travel Time .....	225
Back of Queue Algorithm .....	225
CORSIM Processing Methodology .....	228
Degree of Saturation .....	236
Delay and Level of Service .....	237
Disutility Index .....	242
Double Cycling .....	243
Effective Green Time .....	244
Efficiency .....	244
Excess Operating Cost .....	245
Flow .....	245
Force-Off and Maximum Green .....	245
Fuel Consumption .....	246
HCS Import Methodology .....	248
Interference .....	251
Macroscopic Simulation Methodology .....	251
Maximum Back of Queue .....	255
One-Touch CORSIM Animation .....	257
Operating Cost .....	260
Overlap Phasing .....	260
Permitted Movement Methodology .....	262
Platoon Dispersion .....	269
Profile View .....	274
PROS .....	276
Percent of Vehicles Arriving on Green .....	279
Queue Service Time and Green Extension Time .....	279
Queue Spillback and Queue Spillover .....	280
Sensitivity Parameters .....	282
Shared Lanes Methodology .....	283
Sign Control Methodology .....	287
Spyglass Internal Data Viewer .....	288
Stops .....	290
Throughput .....	293
Time Full and Critical Link .....	293
Time Jammed .....	293
Time-Space Diagram .....	294
Total Operating Cost .....	298
Total Travel and Total Travel Time .....	299
Uncoordinated Operations Methodology .....	300

## **Chapter 5: TRANSYT-7F Optimization Process 303**

Initial Timing Model .....	303
Optimization Node List .....	307
Hill-Climb Optimization .....	307
Cycle Length Optimization .....	314
PROS Optimization .....	318
Optimization Under Congested Conditions .....	320

Grouped Nodes .....	322
Genetic Algorithm Optimization .....	325
Genetic Algorithm Optimization .....	325
Recommended G.A. Parameter Values .....	331
Genetic Algorithm Options .....	332
Phasing Sequence Optimization .....	335
Multi-Period Optimization .....	342
Uncoordinated Optimization .....	344
Direct CORSIM Optimization .....	345
<b>Chapter 6: Tutorial Examples</b>	<b>351</b>
Creating a Brand-New Input Data File .....	351
Creating a New Input Data File by Importing HCS Files .....	373
Editing Input Data (Sensitivity Analysis) .....	376
Progression Analysis Examples .....	380
Queue Spillback and Queue Spillover Examples .....	389
Actuated Estimation Examples .....	399
Multi-Period Analysis Examples .....	405
Five Approaches or More .....	412
Sign Control Examples .....	413
Lane-by-Lane Analysis Example .....	418
CORSIM Processing and Optimization Examples .....	421
<b>Appendix A: TRR Paper</b>	<b>435</b>
Appendix A: TRR Paper .....	435
<b>Appendix B: Traffic Network Study Principles</b>	<b>453</b>
Calibration .....	453
Empirical Model .....	453
Analytical Model .....	453
Simulation Model .....	453
Macroscopic Model .....	453
Microscopic Model .....	454
Deterministic Model .....	454
Stochastic Model .....	454
Network Partitioning .....	454
Phase Failures .....	454
Red Light Running .....	454
Internal Links .....	455
External Links .....	455
Peak Hour Factor .....	455
Analysis Period .....	455
Control Delay .....	456
Objective Function .....	456
Verification and Validation .....	456
<b>References</b>	<b>459</b>
References .....	459
<b>Index</b>	<b>463</b>

# Introduction

## License Agreement

HIGHWAY CAPACITY SOFTWARE™, TRANSYT-7F™, AND HIGHWAY SAFETY SOFTWARE™

HCS2022™ END USER LICENSE AGREEMENT

Copyright © 2021 University of Florida. All rights reserved.

This is an agreement between you and the University of Florida Board of Trustees for the benefit of the McTrans Center.

BY USING THE PROGRAMS WITHIN THE ELECTRONIC TRANSMITTED PACKAGE OR DIRECT DOWNLOAD (“SOFTWARE”), YOU AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT. DO NOT USE THE SOFTWARE WITHOUT FIRST READING AND UNDERSTANDING THE TERMS OF THIS LICENSE AGREEMENT.

If you do not agree with the terms of this license, discontinue use, delete and/or return the unused Software immediately to the distributor from which it was obtained.

1. **LICENSE:** In consideration of the required license fee, the University of Florida McTrans Center (hereinafter called “McTrans”) hereby grants you (hereinafter called “End User”) a nonexclusive license to use the enclosed HIGHWAY CAPACITY SOFTWARE, TRANSYT-7F, and HIGHWAY SAFETY SOFTWARE in HCS2022 (hereinafter called “Software”) as described in the applicable portion of section 3 for a period in which the End User pays the subscription fee established by McTrans.

2. **OWNERSHIP:** This license is not a sale of the Software; it is a license to use the Software under the terms of this License Agreement.

3. **USE:** The Software is licensed to End Users for a particular use, depending on the particular license that is designated on order and described in this Section 3.

a. **Single Office License.** The End User may use the Software on any number of computers within End User’s immediate office, which is defined as a single organizational unit, having a unique function, comprising one or more contiguous rooms in a single location. All persons using the Software are also End Users. End Users may also use the Software on portable computers while on official travel and on home computers, as long as ALL such use is on behalf of the office for which this Software License was purchased. This Software License does not authorize the use of the Software at any offices that are related to that single organizational unit but are not located at the same geographic location, even if the Software is stored on a portable computer. If you have any questions as to the validity of your End User

License, contact McTrans.

b. Academic License. The End User may use the Software only within a laboratory within End User's immediate school, which is defined as a single school unit, having a unique function, comprising one or more contiguous rooms in a single location, up to the permitted (purchased) number of copies, provided the use is supervised in a classroom setting for teaching or research purposes, and appropriate protection is in place against students copying the Software and taking it out of the lab. The responsibility for ensuring this protection of the Software rests with the End User. Multiple laboratories require multiple licenses. The Software may not be installed or used on any computer outside of the licensed laboratory under the Academic License, including (but not limited to) any office, home, networked or portable computer. If you have any questions as to the type or validity of your End User License, contact McTrans.

WARNING! If this license is a Single Office License for use in traffic and transportation applications, the Software may not be used for classroom or workshop instruction, demonstrations, conferences or other similar purposes without the express, written consent of McTrans. (Contact McTrans for an Academic License.) If this license is an Academic License for use in training or educational applications, the Software may not be used for traffic and transportation applications or other similar purposes without the express, written consent of McTrans. (Contact McTrans for an Applications License.)

4. NETWORK INCOMPATIBILITIES: McTrans is not able to provide technical support for issues resulting from network incompatibilities.

5. GENERAL: The Software is not designed for network use, and such use is not supported. No network use is permitted, including but not limited to, using the Software either directly or through commands, data or instructions from or to a computer not already licensed and part of the internal network, for Internet or web hosting services or by any user not licensed to use this copy of the Software through a valid license from McTrans.

6. COPY RESTRICTIONS: Unauthorized copying, distribution, or dissemination of the Software is expressly prohibited. End User may be held legally responsible for any copyright infringement that is caused, encouraged, or enabled by failure to abide by the terms of this license. Subject to these restrictions, End User may make copies of the Software for backup purposes and for use of the Software on any computers located within its office complex, as described in paragraph 3. Under no circumstances may the Software or associated printed documentation be copied for sale, distribution, or dissemination, except as provided in paragraph 3, without the express, written consent of McTrans. End User may reproduce the associated documentation as necessary to use within the geographic location for which the license is granted. End User may not modify, adapt, translate, reverse engineer, decompile, disassemble, or create derivative work from the Software or any associated written materials without the prior written consent of McTrans. ANY SUCH ACTIVITY OR PRODUCT BECOMES THE PROPERTY OF McTRANS.

7. TRANSFER RESTRICTIONS: The Software is licensed only to the End User, and this License Agreement may not be assigned or transferred to anyone without the prior written consent of McTrans. Any authorized transferee of this License Agreement shall be bound by the terms of this License Agreement. End User may not transfer, assign, rent, lease, sell, or otherwise dispose of the Software except as expressly provided in this License Agreement.

8. REGISTRATION AND SUPPORT: This license is automatically registered to the person or entity that purchased it. If the End User wishes to change the registration, he or she must fill out and return the registration card contained in this package. McTrans may, from time to time, revise or update the Software to correct bugs or add minor improvements. McTrans will supply, at its discretion, minor updates to the End User without additional charge and will provide reasonable telephone or email support to the End User during the normal business hours of McTrans for the period for which End User pays the subscription fee established by McTrans. McTrans reserves the right to determine the extent of the support. McTrans reserves the right to make substantial revisions and enhancements to the Software and to market such substantially revised versions as separate products. End User shall not be entitled to receive such separate products without additional charge. Upon release of any such revised version of

the Software, the obligation to provide the above described technical support services to End User shall cease, unless End User chooses to pay to license the enhanced version of the Software, whether or not the End User has previously requested or received support on the prior version.

9. LIMITED WARRANTY AND LIMITATIONS OF REMEDIES: McTrans warrants the magnetic or optical medium on which the Software is furnished to be free from defects in material and workmanship under normal use for a period of thirty (30) days from the date of delivery to the End User as evidenced by a copy of invoice. The entire liability and End User's exclusive remedy shall be the replacement of any diskette found to be defective. Defective diskettes should be returned with a copy of the End User's invoice to the distributor from which the Software was obtained. End User agrees that it is the responsibility of End User to understand fully the limitations of and assumptions underlying the Software and not to make use of the Software without the assistance of personnel with appropriate expertise. End User accepts all responsibility for review, application, and use of the program output and accepts all responsibility for assurance that the output meets provisions of any applicable design code, standard, or any project specification.

EXCEPT AS PROVIDED ABOVE, THE PRODUCT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. McTRANS DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET END USER'S REQUIREMENTS OR THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PRODUCT IS WITH THE END USER. IN NO EVENT WILL McTRANS BE LIABLE TO END USER FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE EVEN IF McTRANS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. THE ABOVE WARRANTY GIVES END USER SPECIFIC LEGAL RIGHTS AND END USER MAY ALSO HAVE OTHER RIGHTS IN THE UNITED STATES WHICH VARY FROM STATE TO STATE. USE OF THE SOFTWARE WHERE THE END USER BELIEVES OR HAS REASON TO BELIEVE THAT DAMAGE MAY OCCUR IS A VIOLATION OF ANY AND ALL WARRANTIES. McTRANS HAS NO KNOWLEDGE OF THE HARDWARE OR SOFTWARE CONFIGURATIONS OF THE END USER AND ASSUMES NO LIABILITY FOR THEIR DAMAGE.

10. TERMINATION: This license is effective from the date End User receives the Software until this License Agreement is terminated. End User may terminate it by returning or destroying all copies of the Software and associated documentation in End User's possession or control. If any terms of this license are breached by End User, the license shall terminate immediately.

11. GOVERNING LAW: This License Agreement shall be governed by the laws of the State of Florida, U.S.A. Venue for all disputes related to this License Agreement shall be in the United States District Court of the Northern District of Florida, Gainesville Division, or in the Circuit Court of the Eighth Judicial Circuit for Alachua County Florida.

12. U.S. GOVERNMENT RESTRICTED RIGHTS: The Software and associated documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subdivision (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at 52.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48CFR52.227-19. The contractor is McTrans, University of Florida, PO Box 116585, Gainesville, FL 32611-6585.

13. EXPORT REGULATION: The Software may be subject to U.S. export control laws, including the U.S. Export Control Reform Act and its associated regulations. End User will not directly or indirectly, export, re-export, or release the Software to, or make the Software accessible from, any country, jurisdiction or person to which export, re-export, or release is prohibited by applicable law. End User will comply with all applicable laws and complete all required undertakings (including obtaining any necessary export license



or other governmental approval) prior to exporting, re-exporting, releasing, or otherwise making the Software available outside the U.S.

14. **INTEGRATION:** This License Agreement is the complete and exclusive statement of the agreement between End User and McTrans and supersedes any proposal or prior agreement, oral or written, and any other communications between End User and McTrans relating to the subject matter of this License Agreement. This License Agreement may only be modified by a written agreement made subsequent to the date hereof and signed by both parties.

15. **SEVERABILITY:** If any portion of this License Agreement shall be declared void or a nullity by a court of competent jurisdiction then the remainder shall survive and remain in full force. Should you have any questions concerning this license, you may contact the McTrans Center, University of Florida, PO Box 116585, Gainesville, FL 32611-6585, U.S.A., [1] (352) 392-0378.

## Acknowledgements

McTrans Center, University of Florida Transportation Institute  
PO Box 116585, Gainesville FL 32611-6585  
Telephone: 1-800-226-1013 Fax: 352-392-6629  
Web: [mctrans.ce.ufl.edu](http://mctrans.ce.ufl.edu) Email: [mctrans@ce.ufl.edu](mailto:mctrans@ce.ufl.edu)

We acknowledge the Transportation Research Board (TRB), the USDOT's Federal Highway Administration (FHWA), the American Traffic Safety Services Association (ATSSA), the Institute of Transportation Engineers (ITE), and the American Association of State Highway and Transportation Officials (AASHTO) for all text, figures, and references included in HCS and express our sincere appreciation for permission to include this information.

## Trademarks and Copyrights

- Microsoft(R), MS-DOS(R), and Windows(TM) are registered trademarks of Microsoft Corporation.
- McTrans(TM), HCS(TM) and TRANSYT-7F(TM) are registered trademarks of the University of Florida.
- PASSER(R) is a registered trademark of the Texas Transportation Institute.
- HCS, HCS2022, HCS7, HCS2010, HCS+, HSS, TRANSYT-7F, and T7F are copyright University of Florida.
- PASSER II, and PASSER III are copyright Texas Transportation Institute.
- TSIS, CORSIM, NETSIM, FRESIM, TRAFED, TRAFVU, and TShell are copyright University of Florida, with portions copyright ITT Industries, Inc., Systems Division.



# Chapter 1: TRANSYT-7F Software Package

## Scope and Qualities

TRANSYT-7F is a traffic simulation and signal timing optimization program. The primary application of TRANSYT-7F is signal timing design and optimization.

TRANSYT-7F features genetic algorithm optimization of cycle length, phasing sequence, splits, and offsets. TRANSYT-7F is the only software package available that combines a state-of-the-art optimization process (including genetic algorithm, multi-period, and CORSIM optimization) with a state-of-the-art macroscopic simulation model (including queue spillback, platoon dispersion, and actuated control simulation).

The primary qualities of the underlying macroscopic traffic model within TRANSYT-7F include detailed simulation of platoon dispersion, queue spillback, queue spillover, traffic-actuated control, and the flexibility to perform lane-by-lane analysis. In addition, while other models are limited to analyzing 4 or 5 intersection approaches, there is no practical limitation to the number of approaches that can be simulated by TRANSYT-7F. Explicit handling of both right-hand and left-hand driving, as well as both English and metric units, allow TRANSYT-7F to be used worldwide.

The primary qualities of the TRANSYT-7F optimization process include the availability of multiple search techniques (hill-climb and genetic algorithm), numerous optimization objective functions (e.g., involving combinations of progression opportunities, delay, stops, fuel consumption, throughput, and queuing), extensive ability to customize the optimization process, and the ability to optimize all signal settings (cycle length, phasing sequence, splits, and offsets).

The current version of TRANSYT-7F is dimensioned to accommodate a maximum of 99 intersections per data file. Although many cities contain more than 99 coordinated intersections, they are typically subdivided into much smaller coordinated sections, with small groups (or "clusters") of intersections coordinated with one another. The software can analyze a maximum of 7 "single-ring" signal phases per intersection, which is sufficient for modeling virtually any pre-timed or traffic-actuated control plan.

In addition to signalized intersections, TRANSYT-7F can explicitly simulate two-way stop-controlled (TWSC) intersections, and yield-controlled intersections. All-way stop-controlled (AWSC) intersections and roundabouts can be modeled implicitly through their effects on platoon dispersion and signalized intersections, but measures of effectiveness (MOEs) for these intersections are not estimated or provided.

TRANSYT-7F is considered an "offline" optimizer because the signal timing plan is developed in the engineer's office, well in advance of field implementation. Before performing any optimization, a certain amount of engineering expertise and judgment are required for 1) coding the input data, and 2) calibrating the model. For a re-timing project, signal timing programs should ideally be made to reasonably replicate existing conditions in the field before proceeding with any optimization. If the model cannot reasonably replicate existing conditions in the field, optimization results become less reliable. Software packages are not smart enough to code their own input data, and are not smart enough to calibrate themselves to match field conditions.

Pre-timed (or fixed-time) control is a relatively old form of signal control in which the green times always remain constant from cycle to cycle. Timing plans can be easily implemented in the field, and detectors are not needed. Traffic-actuated control is a more modern form of signal control in which green time is terminated soon after the queue has been served. If congestion is not too severe, actuated control can be helpful in minimizing delay, because green time does not get wasted on intersection approaches with no vehicles present. The most efficient use of green time is to always be serving queues, with the possible exception of a progression design where the platoon is about to arrive. Many cities employ actuated control extensively due to its favorable ratio of cost savings versus extra effort for field implementation. TRANSYT-7F can handle both pre-timed and actuated control with relative ease.

Adaptive control is an advanced form of signal control in which cycle length and phasing sequence optimization occurs in "real time" in the field. Numerous software/hardware packages (not listed here) are considered "online" optimizers, and are available for performing adaptive control optimization. Although TRANSYT-7F is not specifically designed for adaptive control, some adaptive control solutions have employed TRANSYT for automatically performing numerous optimization runs throughout the day.

## History of TRANSYT-7F

The TRANSYT-7F package is a microcomputer-based system. Mainframe support of the TRANSYT-7F model itself has been discontinued, due to the considerable productivity improvements of PCs. TRANSYT-7F is an acronym for TRAFFIC NETWORK STUDY Tool, version 7F. The original TRANSYT model was developed by the Transport Research Laboratory (formerly Transport and Road Research Laboratory) in the United Kingdom. TRANSYT, version 7 was "Americanized" for the Federal Highway Administration (FHWA); thus the "7F." The TRANSYT-7F program and the original TRANSYT-7F manual were developed for the Federal Highway Administration (FHWA) under the National Signal Timing Optimization (NSTOP) Project by the University of Florida Transportation Research Center (TRC).

The TRANSYT model was originally developed by Dr. Dennis I. Robertson of the United Kingdom in 1967, while he was employed with the Plessey Automation Company. Shortly after his announcement of TRANSYT, Dr. Robertson joined the Road Research Laboratory, later the Transport and Road Research Laboratory (TRRL) in Crowthorne, Berkshire, in the United Kingdom. The first TRANSYT report [2] was also issued as a TRRL Report, LR 253 [3], which is one of the primary TRANSYT references. This first version was written in machine code for a Myriad computer.

Dr. Robertson and his colleagues converted the program to FORTRAN and continued to make improvements over the years. The list below indicates the improvements that were made.

1. TRANSYT, 1967. This was the original version.
2. TRANSYT/2, 1968. This was the first FORTRAN version and included provisions for more than three controller phases.
3. TRANSYT/3, 1970. Improvements were made to the inputs and data checking.

4. TRANSYT/4, 1971. A routine was added to calculate initial timings, the flow profile plots were added, and a provision was made for shared stoplines (i.e., buses and cars could be represented separately even though they use the same roadway).
5. TRANSYT/5, 1972. The shared-stopline facility was expanded to allow multiple links, and bus progression and bus stop delay were added. This was the first version used widely in the United States.
6. TRANSYT/6, 1975. An improved model of stops was added, and other improvements enabled the program to be used on smaller computers [4]. This version was also widely used and has been modified extensively by others to add other capabilities.
7. TRANSYT/7, 1978. Data preparation and run time were significantly reduced by elimination of the need to code a "link list" and a change in the optimization process, respectively. Modeling of bottlenecks was also added. This is the version of TRANSYT [5] originally modified to become TRANSYT-7F.
8. TRANSYT/8, 1981. This version includes estimates of fuel consumption, modeling of yield control, inspection of a range of cycle lengths and other modifications.
9. TRANSYT/9, 1987. This update [6] to TRANSYT/8 runs on microcomputers and has supporting software for data entry, cycle selection, phase sequencing and graphic displays.

#### Development of the TRANSYT-7F Package

Since the original issue of TRANSYT-7F, there have been a number of improvements in the program. The various releases are summarized briefly below:

1. Release 1 (January 1981) was the original program made available to the project cities participating in the National Signal Timing Optimization (NSTOP) project [7].
2. Release 2 (July 1981) was the first general public release. Several bugs found in release 1 were fixed.
3. Release 3 (February 1983) had improved outputs and error checking and added capabilities for a "punch" data file (or "deck"), cycle range evaluation and other improvements. The original TRANSYT7 input option was deleted.
4. Release 4 (June 1984) was updated to the FORTRAN 77 standard and the program was converted to run on microcomputers. The Platoon Progression Diagram (PPD) capability was added, and as a result of its incorporation into the AAP, an 80-column output option was added.
5. Release 5 (June 1987) added four significant new modeling capabilities to the simulation submodel:
  - a permitted-movement model for permitted, conflicting movements,
  - explicit treatment of "sneakers",
  - explicit treatment of stop-sign control, and
  - modeling of shared (or optional) lane movements.
6. Release 6 (October 1988) added a number of new modeling capabilities and implemented several changes that many users had suggested, including:
  - improved calculation of average phase durations for actuated controllers [based on research and development by the University of California, Berkeley, sponsored by the California Department of Transportation (CALTRANS)],
  - a bandwidth constraint which preserves a user supplied thru green band (based on

research by FHWA),

- expansion of the optimization objective function to optionally include excess queue backup (spillback) and/or operating cost,
- improved link-to-link weighting to prioritize specific routes,
- expanded use of the former Time-Space Diagram input card for "route summary" outputs;
- optimization of grouped node splits,
- recommendation of nodes which are candidates for double cycling, and
- improved Flow Profile Diagrams.

7. Release 7 (June 1992) added a number of new modeling capabilities, including:

- a new TRANSYT-7F Executive menu and file processing program for the microcomputer version, called T7FEX,
- explicit optimization of progression opportunities (PROS),
- new split optimization algorithm, based on degrees of saturation,
- explicit handling of overlap phases,
- replaced the random delay estimate with the one from the HCM [12],
- added a user-selectable family of permitted movement-models and changed the default model to one developed in Australia,
- revised stops algorithm, for better accuracy near or at saturation,
- use the "punch" file to capture the intermediate "best" timing plan(s) in a cycle length optimization run,
- allow for optimizing specific nodes more than once in each optimization step,
- permit coding of initial offsets while requesting the calculation of initial splits by TRANSYT-7F,
- network size increased to 100 nodes and 600 links,
- allow 5-digit node and link numbers,
- optional NEMA or user-defined standard link numbering schemes,
- improved output reports, and
- miscellaneous other improvements.

8. Release 8 (March 1998) added significant new modeling capabilities, including:

- developed a new simulation modeling process to permit:
  - ◆ explicit modeling of saturated (and oversaturated) conditions including spillback effects,
  - ◆ more accurate calculation of the back of queue, placing arrivals at the back of queue (i.e., horizontal queue) rather than at the stopline (i.e., vertical queue),
  - ◆ different cycle lengths among intersections for uncoordinated operations,
  - ◆ simulation of the complete time period with multiple cycles (results from previous releases were based on only one cycle of simulation),
  - ◆ more accurate intra-link travel time, and
  - ◆ more accurate estimates of stops and queue delay;
- added new optimization objective functions that emphasize vehicle throughput and queue management for saturated conditions;
- allows multiple time period inputs;
- increased the simulation time base (steps) to permit up to one second per step;
- added a Windows 95/NT version of TRANSYT-7F (and McT7F);
- modified the delay model to be consistent with the latest version of the Highway Capacity Manual (HCM) [12];
- added level of service (LOS) for signalized intersections as an explicit output;
- added performance measures for saturated conditions;
- implemented a random-plus-oversaturation term for stops, but with the same proviso as for delay;

- allow non-integer lost time, extension of effective green, and change intervals;
- updated the operating cost model;
- converted program messages from the "all upper-case" format to the more readable lower-case format, and
- miscellaneous other improvements.

9. Release 9 (February 2001) added additional new features, including:

- genetic algorithm optimization
- phasing sequence optimization
- HCS-style input data screens and a map view
- alternative methodology for evaluation of traffic-actuated control
- "Spyglass" utility program for viewing internal simulation data
- ability to read HCS data files and convert these into TRANSYT-7F networks
- separate module for CORSIM processing and offset optimization

10. Release 10 (January 2004) added additional new features, including:

- map view enhancements
- interactive time space diagram screen
- timing view screen enhancements
- "direct" CORSIM optimization of cycle length, splits, and offsets
- ability to read CORSIM data files and convert these into TRANSYT-7F networks
- ability to export TRANSYT timings to CORSIM
- graphical coding of lane configuration and peak hour factor (PHF)
- saturation flow rate calculator screen
- improved modeling of shared lanes
- GUI support for multi-period simulation
- multi-period optimization
- "profile view" static graphics
- automatic coding of preset phasing sequences
- summary output reports and formatted reports
- miscellaneous other improvements

11. Release 11 (January 2008) added additional new features, including:

- rewritten GUI features Microsoft "dot-net" architecture
- one-touch CORSIM animation, one-touch HCS analysis
- new Dual-Ring Timing input screen provides familiar NEMA 8-phase data entry
- new dual-ring formatted report provides familiar NEMA 8-phase output
- upgraded Lane Configuration screen provides full intersection views
- upgraded Saturation Flow Rate screen shows all 12 movements at once
- saturation flow rate adjustment factors are now saved into the TIN file
- copy-and-paste intersections in the Map View, to avoid re-coding lane configuration, volumes, and timings
- combined the record type editor and GUI into one program
- agency information on formatted report and summary reports
- new route summary and satflow summary text reports
- improved optimization of grouped nodes
- import files from HCS-Interchanges
- the following CORSIM data can now be imported: pre-timed, semi-actuated, fully-actuated, multiple time periods, two-way stop
- direct CORSIM optimization can now optimize the following: pre-timed, semi-actuated, fully-actuated, multiple time periods
- import any CORSIM file, timed and optimized by any other method, and instantly improve on that design

- Russian language version
- import files from Artplan
- HTML-based Help system
- import files from HCS-Streets

## TRANSYT-7F Graphical User Interface (GUI)

The TRANSYT-7F graphical user interface (GUI), TRANSYT-7F.exe, is the primary user-interface program. TRANSYT-7F is typically launched by going to Start > Programs > McTrans > TRANSYT-7F within Microsoft Windows. Numerous sample images of this GUI are provided throughout the documentation.

The TRANSYT-7F macroscopic simulation engine (T7FWin2.exe) can be launched as a stand-alone program (the IOSPEC text file specifies the input and output file names) outside of TRANSYT-7F, but this is not recommended. Running the engine directly from the GUI is now necessary for proper functioning of several program features including genetic algorithm optimization, hill-climb cycle length optimization, graphical output screens, establishment of the current link numbering scheme, HCS file import, CORSIM processing, and so on. In certain parts of the software, the macroscopic simulation model will sometimes be referred to as "TRANSYT", whereas "TRANSYT-7F" will instead refer to the overall software package.

Opening data files from the GUI will provoke different behavior and responses, depending on the file type that was opened. File > Open allows the user to open existing TRANSYT-7F or CORSIM input files. If a TRANSYT-7F (\*.TIN) input file is opened, then the Edit menu and TRANSYT-7F processing features will be activated. If an HCS-Signals (\*.XHS) or HCS-Interchanges (\*.XHI) file is opened, a TRANSYT-7F input file will be generated to reflect the HCS data. If a CORSIM input (\*.TRF) file or an Artplan input (\*.XML) file is opened, then a corresponding TRANSYT-7F input (\*.TIN) file is automatically generated. If a CORSIM file is opened, but the corresponding TIN file already exists, then the Edit menu and TRANSYT-7F processing features will be enabled, to facilitate CORSIM processing or optimization.

File > New allows the user to create a template TRANSYT-7F input file having several default parameter values. Subsequently, it is necessary to use the various edit screens in order to create a realistic modeling scenario.

### Output Quick Jump and Time Period Quick Jump

The Output Quick Jump combo box is located at the top of the GUI. The Output Quick Jump can be used to jump between major sections of the legacy output report, such as the link performance and system summary output tables. Certain parts of the output file, such as the signal timing tables, route statistics, and text-based diagrams, will only be available on the Output Quick Jump list if they were requested by the input file. Certain other "optimization" output files, created by genetic algorithm optimization, hill-climb cycle length optimization, and actuated estimation, do not support (and usually do not require) Output Quick Jump capabilities.

The Output Quick Jump is therefore used to jump between major portions of the legacy output report. To search more slowly within a particular section of the legacy report, it is best to place the mouse cursor directly within the output file window, and navigate the report using the keyboard. Note that it is possible to quickly switch focus from the Output Quick Jump to the report text by using the <ENTER> key.



Movement/ Node No.	Flow vph	Sat Flow vphg	Degree of Sat %	Total Travel veh-mi	Travel Time Total veh-h	Avg sec/v	Delay			
							Uniform	Random	Total	Average
							veh-h	veh-h		sec/veh
NB THRU	450	1900P	69	112.41	7.09	56.7	3.48	0.38	3.86	30.9
LEFT	50	1805	28	12.49	0.95	68.4	0.58	0.01	0.59	42.5
SB THRU	450	1900P	69	112.41	8.16	65.2	4.66	0.27	4.93	39.4
LEFT	50	1805	28	12.49	0.99	71.5	0.63	0.01	0.63	45.7
EB THRU	450	1900P	96*	112.41	12.06	96.5	4.63	4.20	8.83	70.6
LEFT	50	1805	28	12.49	0.95	68.4	0.58	0.01	0.59	42.5
WB THRU	449	1900P	96*	112.41	13.33	106.8	7.65	2.44	10.09	80.9
LEFT	50	1805	28	12.49	1.01	72.4	0.65	0.00	0.65	46.6
1 :	1999	MAX =	96*	499.58	44.53		22.85	7.32	30.17	54.3 1

Movement/ Node No.	Uniform	Max Back of Queue Random	Total	Queuing Capacity	Time Full	Critical Link	Fuel Cons.	Eff Gr
		veh/link(ft/lane)			%		gal	s

The Output Quick Jump can also be used to select the summary text reports (Node Output, Node Input, Node Timing, Global Output, Global Input, Global Timing, Route Summary, and Node Satflow). When the node-specific (or route-specific) summary reports are selected, node numbers (or route numbers) are added to the Output Quick Jump, so that the user can choose the node for which a report will be obtained. Note that the appearance and content of these summary reports are sometimes affected by the chosen format (narrow or wide) of the legacy report.

The Time Period Quick Jump, directly to the right of the Output Quick Jump, is only activated when the input data file contains multiple time periods. The Time Period Quick Jump controls the input data and output data that will be displayed by the GUI. The Time Period Quick Jump is capable of dynamically updating the data that is displayed in several areas, including the Time-Space Diagram screen, the Map View, the legacy output file report, the new summary output reports, and the TRANSYT-7F input data edit screens. Note that switching between time periods does not automatically save recent changes to the input data. Changing time periods via the Options > Time Period > Select menu item also has the same effect as the Time Period Quick Jump does.

### Batch Processing

If a TRANSYT-7F input or output file is opened directly from Windows Explorer, the GUI should automatically be launched, to allow editing or viewing of this data. The GUI also offers batch processing functionality, such that third party programs can automatically launch TRANSYT-7F, and then execute the TRANSYT model if desired. TRANSYT-7F may be executed from a command prompt as follows:

<program path>\TRANSYT-7F.EXE <file path>\<file name>

...loads the input file into the interface without performing any runs.

<program path>\TRANSYT-7F.EXE <file path>\<file name> /r

...loads the input file into the interface and runs TRANSYT.

<program path>\TRANSYT-7F.EXE <file path>\<file name> /c

...loads the input file into the interface, runs TRANSYT, and then closes the interface.

For example:

"C:\Program Files\HCS+T7F\T7F11\TRANSYT-7F.exe" C:\Program Files\HCS+T7F\T7F11\Example\FourNode.TIN /r

...when typed from Start > Run in Microsoft Windows, should successfully run the software and process the data file FourNode.TIN.

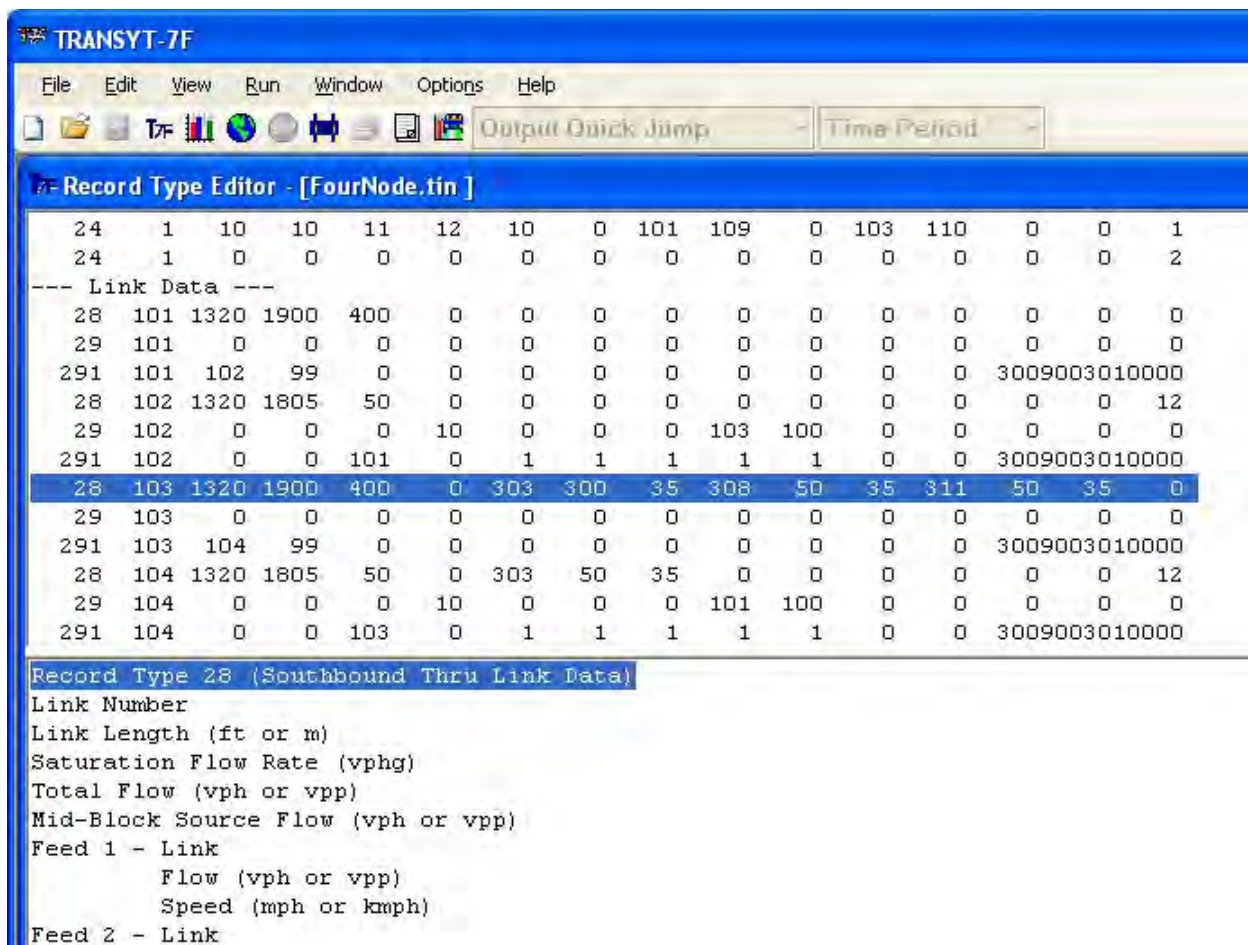
### Hotkeys

The following "hotkeys" are available within the GUI:

- PageUp, PageDown: switch node numbers on the currently active screen
- F1: view context-sensitive help for the currently active screen
- F2: launch record type editor to view TRANSYT-7F raw data for the currently active input file
- F3: launch Map View for the currently active input file
- F4: launch Lane Configuration screen for the currently active input file
- F5: launch Traffic screen for the currently active input file
- F6: launch Timing View screen for the currently active input file
- F7: launch Dual-Ring Timing screen for the currently active input file
- F8: launch Satflow View screen for the currently active input file
- F9: launch Feeders screen for the currently active input file
- F10: launch Analysis screen for the currently active input file
- F12: exit the TRANSYT-7F program
- Ctrl-O: open a new data file
- Ctrl-S: save the currently active TRANSYT-7F input (\*.TIN) file
- Ctrl-R: run simulation or optimization
- Ctrl-N: create a brand-new TRANSYT-7F input (\*.TIN) file
- Ctrl-P: print output report or the currently active GUI screen
- Ctrl-F: find words or numbers within the currently active text file

## Record Type Editor

The "record type editor", illustrated in the screen image below, can be used to view or modify standard TRANSYT-7F (\*.TIN) input files. This editor directly modifies the raw data recognized by the TRANSYT-7F software. In most cases, it will be more efficient and intuitive to use the TRANSYT-7F edit menu screens for input data entry. To launch the record type editor, the user can select View > Record Types, or simply press the "F2" function key.

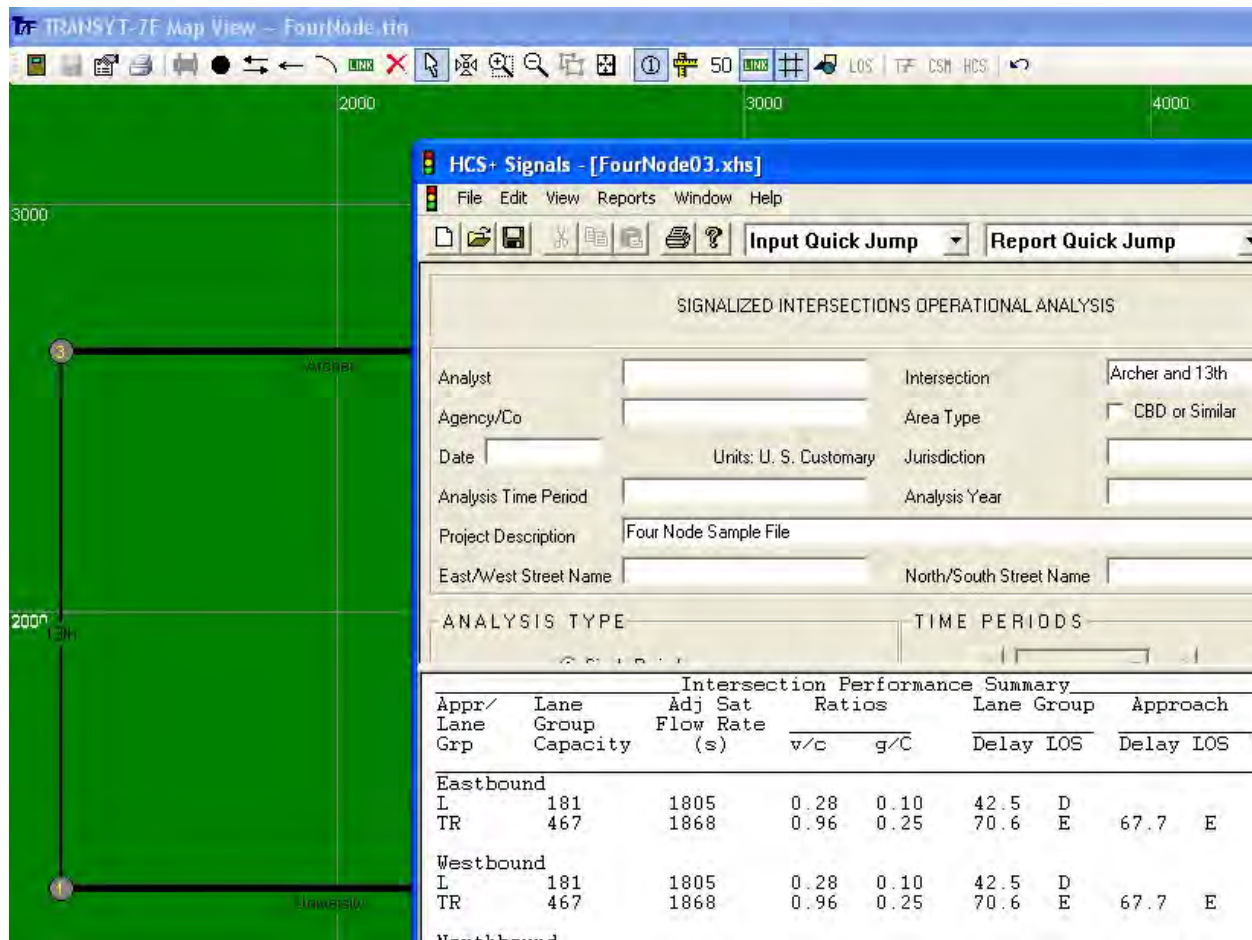


## HCS Import/Export

Highway Capacity Software (HCS) data files can now be loaded directly into TRANSYT-7F, provided that they are available in \*.XUS (HCS-Streets), \*.XHS (HCS-Signals), or \*.XHI (HCS-Interchanges) format. There are two possible methods for loading HCS files into TRANSYT-7F. The first method involves simply loading an HCS file by selecting File > Open from TRANSYT-7F. This will automatically generate a corresponding TRANSYT-7F data file for the HCS intersection(s).

The second method involves appending HCS intersection data to an existing TRANSYT-7F (\*.TIN) input file. First, it is necessary to open the existing TIN file with TRANSYT-7F. Next, it is necessary to select File > HCS > Append an HCS-Streets intersection to this network, and open an existing file of type \*.XUS. This will enable the "New Node Number" dialog box. After selecting the new node number, the new data will automatically be added to the TIN file.

It is also possible to right-click on an intersection in the Map View and view HCS results for that intersection. If the HCS file already exists, it will be opened automatically by HCS. If not, an HCS file will be created instantly by a data conversion program, and this brand-new file will be opened automatically by HCS. Timing plans can also be exported from TRANSYT-7F to HCS within the Timing View screen. For more technical details, refer to the section of documentation entitled "HCS Import Methodology". For a tutorial example, refer to the section of documentation entitled "Creating a New Input Data File by Importing HCS Files".



## CORSIM Import/Export/Processing/Optimization

CORSIM is a program that performs microscopic traffic simulation of surface streets and/or freeways. CORSIM is sold separately within the TSIS package, which also includes TRAFED (graphical input editor) and TRAFVU (animation and static graphics viewer). CORSIM cannot optimize signal timing on its own, but recent versions of TRANSYT-7F have been designed to work closely with CORSIM. Readers who do not own CORSIM, or who do not have an interest in CORSIM, can skip over this section.

### CORSIM Import

TRANSYT-7F can now import any CORSIM file that contains a signalized intersection. The following CORSIM data can now be imported: pre-timed intersections, semi-actuated intersections, fully-actuated intersections, multiple time period data files, and two-way stop (TWSC) intersections. The user can choose whether or not to import fully-actuated intersections and TWSC intersections, because these may sometimes interfere with an optimization project. However, for a straight CORSIM post-processing project, it is probably best to simply import everything. After importing a CORSIM file, TRANSYT-7F provides numerous helpful features for further analysis including:

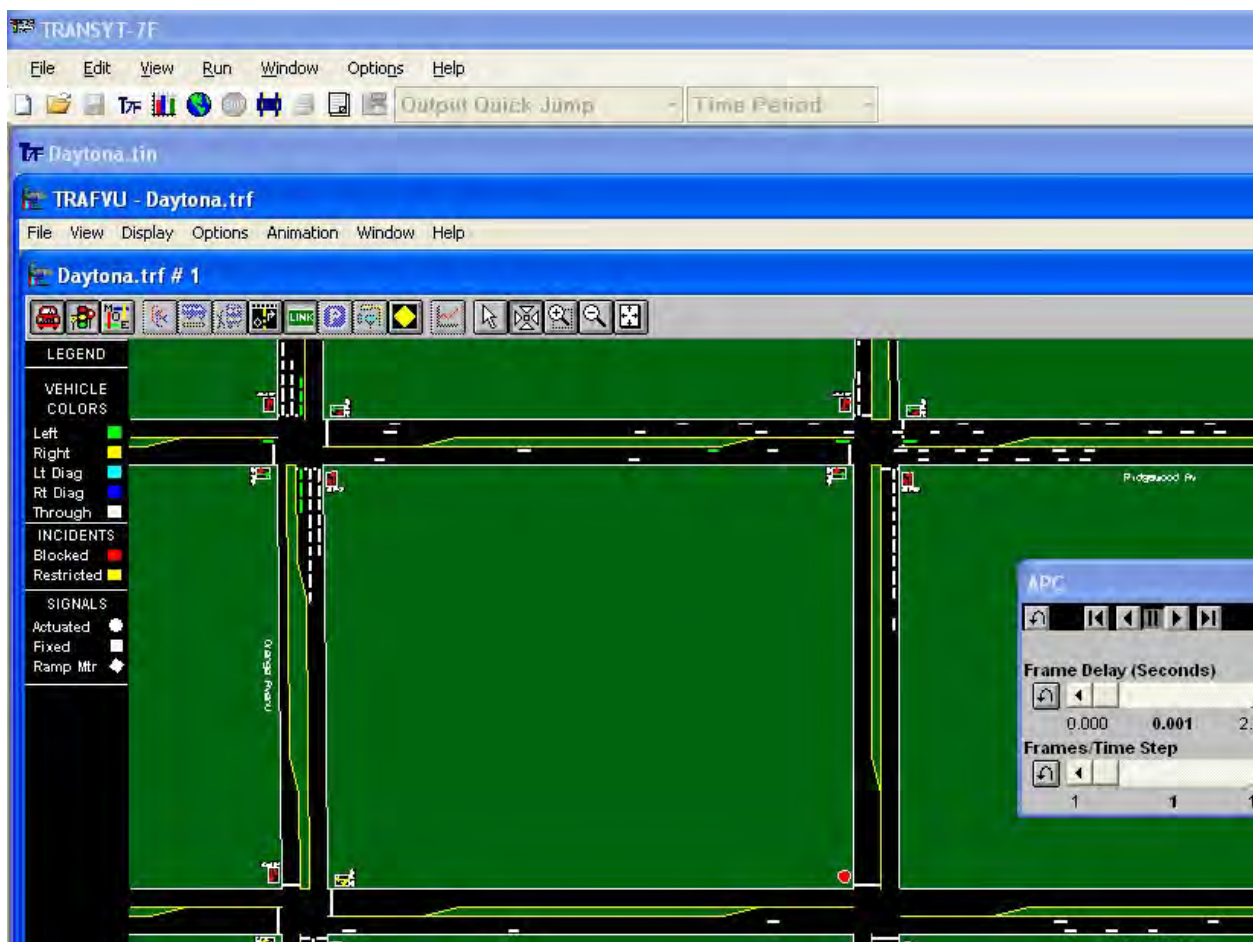
- direct CORSIM optimization
- conversion to macroscopic TRANSYT analysis
- post-processing of CORSIM's own simulation results



- time-space diagrams that reflect CORSIM's own signal timing
- formatted reports that reflect CORSIM's own simulation results
- superimpose CORSIM's own intersection-wide control delay and level of service on the Map View

## CORSIM Export

If no CORSIM file exists yet, TRANSYT-7F also offers "one-touch" CORSIM animation, or CORSIM file export. This pre-processing feature creates CORSIM networks from scratch, and is available for either the entire TRANSYT-7F network, or for individual intersections. To create a CORSIM input file that reflects an entire TRANSYT-7F network, click on the "CORSIM Animation" icon on the TRANSYT-7F toolbar, or select "File > CORSIM > View animation for this network". To create a CORSIM input file that reflects one intersection only, simply right-click on an intersection in the Map View, and select "CORSIM > View animation for this node". If the user doesn't wish to view animation, CORSIM input and/or output files can optionally be generated without launching animation.



Currently, CORSIM requires input data to be specified in English units. However, the CORSIM pre-processing feature is capable of automatically converting metric-unit TRANSYT-7F files to English-unit CORSIM files.

TRANSYT-7F can export the following data to CORSIM:

- pre-timed intersections, semi-actuated intersections, fully-actuated intersections, two-way stop

intersections

- multiple time period data files
- node coordinates from the Map View
- link curvature from the Map View
- bitmap backgrounds from the Map View

### CORSIM Processing

TRANSYT-7F can also be used as a CORSIM output post-processor for the following applications:

- perform multiple runs with multiple random number seeds
- calculate network-wide averages for vehicle delay and vehicle trips
- calculate intersection-wide control delay and level of service (LOS)
- perform sensitivity analysis using a network-wide traffic volume multiplier
- calculate link-specific vehicle delays, trips, queue lengths, phase failures, lane changes, storage, and occupancy

Link-specific results are computed for the surface street (NETSIM) level only. Despite this, it is still possible to process a CORSIM network containing both NETSIM and freeway (FRESIM) links. Network-wide results reflect both NETSIM and FRESIM links.

After a CORSIM file has been opened by TRANSYT-7F, CORSIM processing can be performed by simply selecting "Run TRANSYT-7F". Following the CORSIM processing run, all results are immediately presented in a summary text file (e.g., "Filename Summary.txt"). All results calculated by CORSIM processing are average results that can account for multiple simulation runs, and multiple random number seed variations. Following CORSIM processing, intersection control delay and level of service can be viewed in the TRANSYT-7F Map View by clicking on the "LOS" icon.

For more technical details, and a sample output report, refer to the CORSIM Processing Methodology section of the documentation. For tutorial examples, refer to the CORSIM Processing and Optimization Examples section of the documentation.

### CORSIM Optimization

CORSIM can optimize cycle lengths, force-offs, pre-timed green splits, offsets, and maximum green times, when used together with TRANSYT-7F. CORSIM simulates the candidate timing plans itself, and TRANSYT-7F manages the genetic algorithm. After a CORSIM file has been opened by TRANSYT-7F, the so-called "direct CORSIM optimization" can be performed by simply selecting "Run TRANSYT-7F", provided that optimization settings have been selected on the Edit > Analysis screen. Following the CORSIM optimization run, all results are immediately presented in a summary text file (i.e., "Genetic.txt"). The summary text report for direct CORSIM optimization contains the initial and optimized timing plans. In addition, a copy of the CORSIM input file (called "Genetic.trf") is automatically created, containing the optimized timing plan.

Direct CORSIM optimization is powerful due to the availability of (TRAFVU) animation for the optimized solution. However, the primary drawback of microscopic optimization (relative to traditional macroscopic optimization) is the program running time. The genetic optimization process requires dozens or hundreds of individual simulation runs, and the time required for this number of CORSIM runs may not be practical for large networks. This is because large networks finish simulation much more slowly than small networks. It is possible to request multiple random number seed variations per timing plan candidate, for improved reliability of the final solution, but this would drastically increase the program running time. For example, if two or three random number variations are required during the optimization process, this may double or triple the required running time all by itself.

If macroscopic hill-climb or genetic optimization (using the TRANSYT simulation engine) is

preferred, then the optimized timing could be optionally exported to a CORSIM input file. If a CORSIM input file was originally imported into TRANSYT-7F and converted to macroscopic analysis, then converting back to microscopic analysis would allow exporting timing back from TRANSYT-7F to CORSIM. If no CORSIM input file exists, then one-touch CORSIM animation could be used to generate the CORSIM input file from scratch.

It is possible to optimize a CORSIM network containing both NETSIM and FRESIM links, if desired. Phasing sequence optimization is not currently available via direct CORSIM optimization. Instead, phasing sequence optimization can only be accomplished using macroscopic (TRANSYT-based) simulation.

For tutorial examples, refer to the CORSIM Processing and Optimization Examples section of the documentation.

## Formatted Report

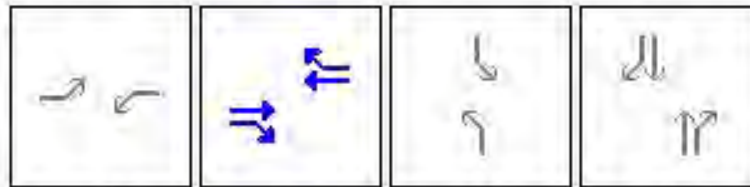
The "formatted report" within TRANSYT-7F provides a one-page summary report of the most important input and output data, including a phasing diagram. The formatted report is now available in both single-ring timing and dual-ring timing formats. The formatted report can be used to display and print data for both the TRANSYT and CORSIM models. The formatted report is launched by either selecting View > Report, or by clicking on the "View Report" icon on the TRANSYT-7F toolbar. Output data fields can only be filled out if a simulation-only run has been performed recently, such that the output data are readily available. Agency information that appears on the report can be specified via Edit > Analysis > Run Info. A sample formatted report for single-ring timing is illustrated below.

## Report View (press Ctrl-P to print)

Node Number 1 *University and 13th*

System Cycle 7

Grn <span style="border: 1px solid black; padding: 2px;">10</span>	Grn <span style="border: 1px solid black; padding: 2px;">25</span>	Grn <span style="border: 1px solid black; padding: 2px;">10</span>	Grn <span style="border: 1px solid black; padding: 2px;">35</span>
Yel <span style="border: 1px solid black; padding: 2px;">4</span>	Yel <span style="border: 1px solid black; padding: 2px;">4</span>	Yel <span style="border: 1px solid black; padding: 2px;">4</span>	Yel <span style="border: 1px solid black; padding: 2px;">4</span>
Red <span style="border: 1px solid black; padding: 2px;">1</span>	Red <span style="border: 1px solid black; padding: 2px;">1</span>	Red <span style="border: 1px solid black; padding: 2px;">1</span>	Red <span style="border: 1px solid black; padding: 2px;">1</span>

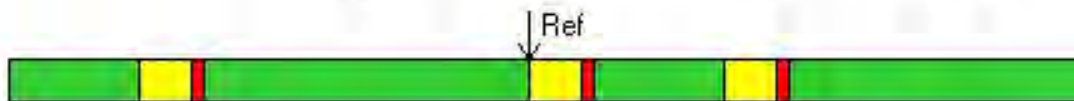


15

30

15

40



Eastbound			Westbound			Northbound			So
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left
Peak Adjusted Volume (vph)									
<span style="border: 1px solid black; padding: 2px;">50</span>	<span style="border: 1px solid black; padding: 2px;">400</span>	<span style="border: 1px solid black; padding: 2px;">50</span>	<span style="border: 1px solid black; padding: 2px;">50</span>	<span style="border: 1px solid black; padding: 2px;">400</span>	<span style="border: 1px solid black; padding: 2px;">50</span>	<span style="border: 1px solid black; padding: 2px;">50</span>	<span style="border: 1px solid black; padding: 2px;">400</span>	<span style="border: 1px solid black; padding: 2px;">50</span>	<span style="border: 1px solid black; padding: 2px;">50</span>
Saturation Flow Rate (vphg)									
<span style="border: 1px solid black; padding: 2px;">1805</span>	<span style="border: 1px solid black; padding: 2px;">1900</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">1805</span>	<span style="border: 1px solid black; padding: 2px;">1900</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">1805</span>	<span style="border: 1px solid black; padding: 2px;">1900</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">1805</span>
Queue Length (veh)									
<span style="border: 1px solid black; padding: 2px;">1.3</span>	<span style="border: 1px solid black; padding: 2px;">15.3</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">1.4</span>	<span style="border: 1px solid black; padding: 2px;">14.5</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">1.3</span>	<span style="border: 1px solid black; padding: 2px;">10.1</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">1.4</span>
Degree of Saturation (%)									
<span style="border: 1px solid black; padding: 2px;">28</span>	<span style="border: 1px solid black; padding: 2px;">96</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">28</span>	<span style="border: 1px solid black; padding: 2px;">96</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">28</span>	<span style="border: 1px solid black; padding: 2px;">69</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">28</span>
Control Delay (sec/veh)									
<span style="border: 1px solid black; padding: 2px;">42</span>	<span style="border: 1px solid black; padding: 2px;">70</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">46</span>	<span style="border: 1px solid black; padding: 2px;">80</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">42</span>	<span style="border: 1px solid black; padding: 2px;">30</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">45</span>
Level of Service									
<span style="border: 1px solid black; padding: 2px;">D</span>	<span style="border: 1px solid black; padding: 2px;">E</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">D</span>	<span style="border: 1px solid black; padding: 2px;">F</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">D</span>	<span style="border: 1px solid black; padding: 2px;">C</span>	<span style="border: 1px solid black; padding: 2px;"></span>	<span style="border: 1px solid black; padding: 2px;">D</span>

Refer to the Timing View documentation for a description of the phasing diagram, and the other timing plan data. Refer to the Traffic screen documentation for a description of the volume and saturation flow rate data. Refer to Chapter 4 for a description of the output data (Queue Length, Degree of Saturation, Control Delay, and Level of Service).



## Sample Data Files

Sample data files are a key component of the overall software package. They allow the user to instantly observe virtually any program feature, without the need to first create a brand-new input file from scratch. When using the installation CD, these sample files are installed on to the hard drive, along with the executable program files. In addition, these sample files (and others) are also available for downloading from the *McTrans* web site. Sample output file listings are shown throughout the documentation, particularly in chapters 5 ("TRANSYT-7F Optimization Process") and 6 ("Tutorial Examples"). Although most of the sample files are associated with the TRANSYT model, a few CORSIM files are also included, to facilitate demonstration of the CORSIM features within TRANSYT-7F.

By default, sample data files are installed to the following subdirectory:

C:\Program Files\HCS+T7F\T7F11\Example

When using File > Open from the TRANSYT-7F graphical user interface (GUI), the user can browse to the above location to access the sample files. The user can optionally make this the default data file folder by using the Options > Data Path menu selection. During the installation process, the user is allowed to install program files to non-default subdirectory locations, but the Example folder will always be automatically appended to this location. If the user does not have "write access" to the Example folder, the sample files should be copied to a more suitable location (e.g., the "My Documents" folder).

### Sample Data File Listing

Sample.bmp: This is a sample bitmap background file, which can be loaded into the Map View if desired.

\*.tof: All files with the "tof" extension, which stands for "TRANSYT-7F Output File", were generated by the TRANSYT-7F traffic simulation engine. These reports contain the detailed measures of effectiveness for the entire network, as well as for individual links and nodes. Throughout the documentation, these output files are also referred to as "legacy" reports. The summary text reports, which are more compact than the legacy report, have a "txt" extension.

\*.trf: These sample CORSIM input files can be used for CORSIM processing and optimization.

\*.xhs: These Extensible Markup Language (xml) data files were generated by HCS-Signals, and can be imported into TRANSYT-7F for building traffic networks.

Daytona.tin: This is the original Daytona Beach sample network from release 7. This sample network contains 12 intersections. It demonstrates several TRANSYT-7F simulation features including arterial routes, bottleneck links, and bus links.

Five Approach.tin: This is a variant of FourNode.tin (see below) having an intersection with five approaches.

FourNode.tin: This is a simple "box" network having four signalized intersections in a square configuration.

FourNode Lefty.tin: This is a left-hand drive version of FourNode.tin, and also uses metric units.

FourNode TP.tin: This is a multi-period version of FourNode.tin.

HCM Sample.tin: This is a simple two-phase intersection, and corresponds with the sample calculation of traffic-actuated phase times from HCM 2000 Chapter 16, Appendix B.

Level2Clean.tin: This is an arterial street having ten intersections, plus a variety of geometrics and phasing sequences.

Level2Perm.tin: This data set is related to Level2Clean.tin, but also contains numerous permitted movements.

Overlap.tin: This sample file illustrates the coding technique to achieve overlap phasing as short as 1 second. The coding technique requires adjustment of the start-up lost time and extension of effective green time at multiple movements.

Sample #3.tin: This input file is created during the second part of a chapter 6 tutorial exercise, entitled "Creating a Brand-New Input Data File".

Sign.tin: This sample file illustrates the coding of a two-way stop controlled (TWSC) intersection, as demonstrated within the Sign Control Examples tutorial section of the documentation.

## Technical Support and Problem Reporting

If you need assistance, or have trouble using TRANSYT-7F, you can obtain technical support in the following ways:

1. Visit the *McTrans* web site (<http://mctrans.ce.ufl.edu/>) for access to frequently asked questions (FAQs), Did You Know? articles, sample data files, known issues, and software patches.
2. Call *McTrans* technical support at **(+01) 352.392.0378**, or toll-free at **1.800.226.1013**.
3. E-mail your request for help (together with any data files that apply) to **mctrans@ce.ufl.edu**.
4. Fax your request for help to **(+01) 352.392.6629**.

If you have any suggestions, or believe you have discovered a problem in the software or documentation, you may also report them to *McTrans* using the contact techniques listed above.

# Chapter 2: TRANSYT-7F Applications

## Macroscopic Simulation of Traffic Flow

One of the two major functions of TRANSYT is to simulate the flow of traffic in a signalized network. Simulation is an analytical process that attempts to represent real world events, in this case traffic flowing through the network, being stopped at intersections by the red signals and soon thereafter moving on the green signals.

The traffic simulation model in TRANSYT-7F is among the most realistic of those available in the family of computerized macroscopic traffic models. A macroscopic model is one that considers platoons of vehicles rather than individual vehicles. TRANSYT simulates traffic flow in small time increments, so its representation of traffic is more detailed than other macroscopic models that assume uniform distributions within the traffic platoons.

The traffic model further utilizes a platoon dispersion algorithm that simulates the normal dispersion (i.e., the "spreading out") of platoons as they travel downstream. It also considers traffic delay, stops, fuel consumption, travel time and other system measures.

TRANSYT simulates traffic flow macroscopically, but in a step-wise manner. The cycle length is divided into small, equal time increments, called steps. A step is typically from one to three seconds, although the relationship between seconds and steps need not be an integer conversion. Step-wise simulation was primarily designed on the assumption of one step equal to one second. The duration of a step will, however, be the finest resolution to which the signal timing can be represented in simulation model. The smaller the step size, the finer the resolution, and the more "accurate" the results. The disadvantage to small step sizes is increased program running times on the computer.

Each signal phase is identified by its start and end times, which are then modified by the start-up lost time (*SLT*) and extension of effective green time (*EEG*) to calculate "effective green" (*g*) times. Since it is known which links may move during each phase, it is a simple matter (at least for unopposed movements) to construct the traffic flow patterns and flow profiles. Refer to the Simulation Methodology section of the documentation for a description of these calculations.

### Simulation Process

TRANSYT gives the user the option to choose between two simulation types: link-wise and step-wise simulation. Before discussing these types of simulation, it is important to understand the meanings of the terms "link" and "movement".

In TRANSYT, a movement is identified by its turning direction regardless of the number of lanes, and regardless of whether the lanes are shared or exclusive. For example, a left-turn movement may include all of the left-turning vehicles from an exclusive left-turn lane, plus any left-turning vehicles from an adjacent shared left-turn lane. This documentation will often refer to the three "turning movements" (left, thru, right) at an intersection approach. The abbreviated term "thru" is often used to designate a through movement, whose basic characteristics are in many ways similar to those of left-turn and right-turn movements.

Internally, within its simulation process, TRANSYT is primarily concerned with links. A link is a flexible entity that can be defined in different ways by the user, to allow for better modeling efficiency and accuracy. A single link could, for example, be defined to represent all three of the turning movements on a single intersection approach. Alternatively, three links could be defined, resulting in a link to represent each turning movement. Finally, in the case of a multi-lane turning movement, it is even possible to define multiple links in order to carefully represent each individual lane within the model.

This begs the question of how to intelligently define links for a real-world project. Although this issue is discussed in many places throughout the documentation, some preliminary remarks will be made here. In general, the software runs more quickly and efficiently with the smallest-possible total number of links defined. In addition, when the number of defined links is kept to a minimum, this also minimizes coding time for the user, and minimizes the possibility of human input errors. But before you proceed to define a single link to represent each entire intersection approach, there must be some mention of modeling accuracy.

When the operations of adjacent turning movements significantly differ from one another, they should be represented by different links, to facilitate more realistic modeling. For example, if a shared left-turn ever yields to an opposing thru movement, it is necessary to define a separate link to represent only the left-turns. Similarly, when the operations of adjacent lanes significantly differ from one another, they should be represented by different links. For example, if the queue length in thru lane #1 is typically double the queue length in thru lane #2, it is helpful to define a separate link to represent only lane #2. This application is known as a "lane-by-lane" analysis. In the current version of the software, link definitions can be handled fairly easily on the Lane Configuration screen described in Chapter 3.

### Link-Wise Simulation

This was the only simulation process available in older versions of TRANSYT-7F, prior to 1998. Under this type of simulation, each individual link is simulated through to completion before moving on to simulate the next link. Links are processed in a specific order such that an upstream-to-downstream propagation is ensured. The "link tree" also takes loops around blocks into account.

Although link-wise simulation is computationally efficient, and is considered to be acceptable for undersaturated conditions, it is not suitable for congested conditions. This is because the effects of queue spillback and queue spillover cannot be accounted for in this simulation process.

In addition, within link-wise simulation, vehicles can only be assumed to arrive at the stop line rather than at the back of queue, as in the real world. This assumption ignores the travel times required between the queued vehicles and the stop line. Also, the queue is assumed to be "stacked" vertically at the stop line. This means vehicles are queued on a link without consideration of the link length. Thus, the simulation can produce a back of queue length that actually exceeds the link's queue storage capacity. Lastly, although TRANSYT reports the maximum back of queue, the calculation is less realistic under link-wise simulation. In spite of these deficiencies, this simulation approach is reasonably realistic for undersaturated conditions. Its main advantage, as noted before, is that it is very efficient in terms of computer running time.

Under link-wise simulation, TRANSYT simulates links (and nodes) by passing through all time steps on each link in turn as follows:

## LOOP THROUGH ALL LINKS

### LOOP THROUGH ALL TIME STEPS

The order the links are processed is determined internally to ensure an upstream-to-downstream propagation. The "link tree" also takes loops around blocks into account.

#### Step-Wise Simulation

This simulation process was initially introduced within release 8 of TRANSYT-7F. Under this type of simulation, all links are simulated for the first time step (typically one second) before moving on to simulate the next time step. Links are processed in a specific order such that a downstream-to-upstream propagation (opposite the sequence of link-wise simulation) is ensured. This simulation process allows the following:

- explicit modeling of saturated conditions including spillback effects
- more accurate calculation of the back of queue
- placing vehicle arrivals at the back of queue
- explicit modeling of multiple (a sequence of) cycles and multiple periods
- allowing different cycle lengths among intersections (for uncoordinated operations)

Step-wise simulation requires longer program running times on the computer than link-wise simulation, but given the speed of today's computers, the difference is negligible for certain projects. From a modeling standpoint, step-wise simulation is superior to link-wise simulation, provided that the input data are specified correctly. There are certain input data pitfalls that are specific to step-wise simulation. For example, incorrectly specified link lengths could create severe queue spillback problems and gridlock, even though the same inputs under link-wise simulation would merely cause incorrect simulation of platoon dispersion. Incorrectly specified queuing capacities could also lead to overly pessimistic results.

Many of the recent modeling features of TRANSYT-7F either require, or automatically specify, step-wise simulation. These include:

- traffic-actuated signal timing estimation
- Profile View static graphics
- Spyglass internal data viewer
- multi-cycle simulation
- multi-period simulation
- multi-period optimization
- uncoordinated operations (different cycle lengths at each intersection)
- optimization under congested conditions (using certain objective functions)

Under step-wise simulation, TRANSYT passes through all links for each time step in turn as follows:

### LOOP THROUGH ALL TIME STEPS

#### LOOP THROUGH ALL LINKS

This is somewhat analogous to treating all links as being parallel (as in a bowling alley) and marching down them simultaneously in time. The links are, however, simulated in a downstream-to-upstream sequence.

#### Traffic Flow Model

Consider a single lane of traffic (or link) with a standing queue at an intersection, waiting for the green display. Following the phase change to green for this link, there is a slight delay before the driver of the first vehicle reacts and crosses the stopline. This is the start-up lost time.

After several vehicles have crossed the stopline (usually about three vehicles), the queue begins to discharge at essentially a constant rate. This rate is called the saturation flow rate. The units of saturation flow rate are either vehicles per lane per hour of green (vplphg), or the total vehicles per hour of green (vphg) across all lanes. Units are expressed in "hours of green", because the saturation flow rate represents the number of vehicles that theoretically would be served if a full hour of green time were available. For example, a saturation flow rate of 1895 vplphg would indicate a 1.9 second front bumper-to-front bumper headway between vehicles, throughout the hour.

The capacity of a link tends to be much lower than its saturation flow rate, since links rarely receive a green light throughout the period of analysis. Capacity is the maximum possible traffic volume in vehicles per hour (vph), considering red time and lost time. Capacity is the product of the saturation flow rate ( $s$ ) and the effective green time ( $g$ ).

Following the phase change from green to yellow for a given link, there is a slight delay before the drivers decide to stop at the stopline. This is the extension of effective green time.

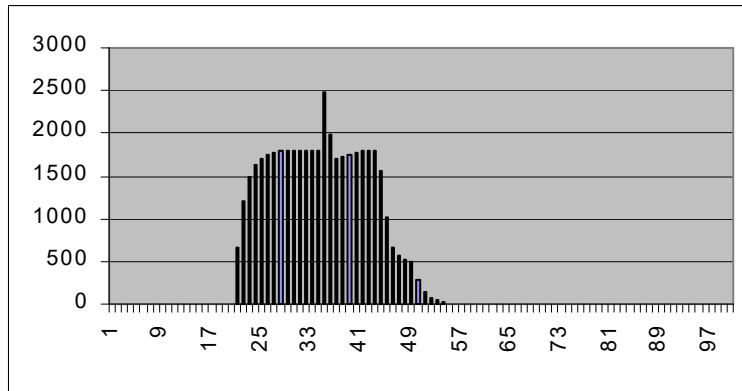
In summary, the macroscopic traffic flow simulation process involves:

1. Start-up lost time ( $SLT$ ) -- time lost due to the initial start up of traffic. Note that TRANSYT assumes that all  $SLT$  accrues to the "first" vehicle, which is a simplifying assumption.
2. Extension of effective green ( $EEG$ ) -- usage of the yellow (and perhaps all-red) change period by vehicles entering during this time.
3. Effective green ( $g$ ) -- actual green minus  $SLT$ , plus  $EEG$ , which is the "go" window that TRANSYT allows traffic on a link to move.
4. Saturation headway ( $h$ ) and saturation flow rate ( $s$ ) -- the minimum headway as the queue departs, which translates to the maximum possible flow rate by,

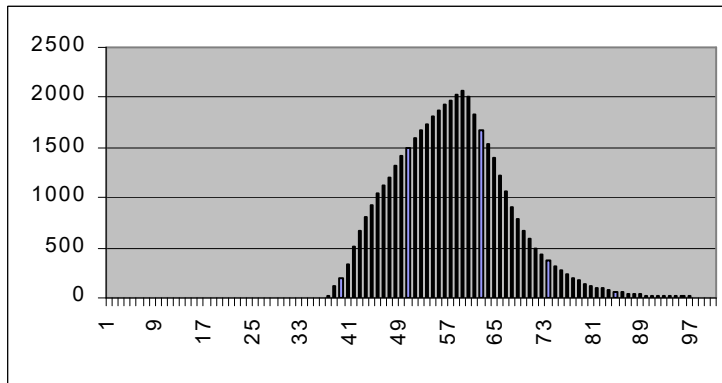
$$s = \frac{3600}{h}$$

If the effective green time is divided into short, equal intervals of time, referred to as steps, the number of vehicles crossing the stopline in each step is the flow rate in vehicles per step (veh/step). This is the vehicular departure profile. TRANSYT always assumes that the input flow distribution from outside the system (e.g., on "external" links) is uniform, or constant. This is how "random" arrivals would tend to look when averaged over a long period of time, say, one hour.

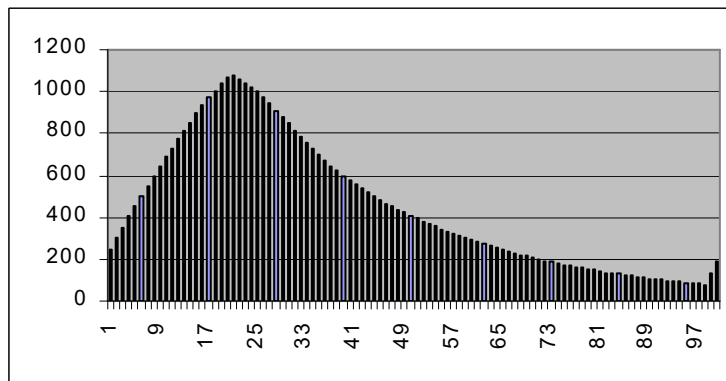
This type of representation, namely the histogram in the figure(s) below, is referred to in TRANSYT as a "flow profile," in this case the arrival profile. This is a "snapshot" of the traffic flow at the downstream end of the link and is one of the most important features of TRANSYT.



**300 ft**



**1320 ft**



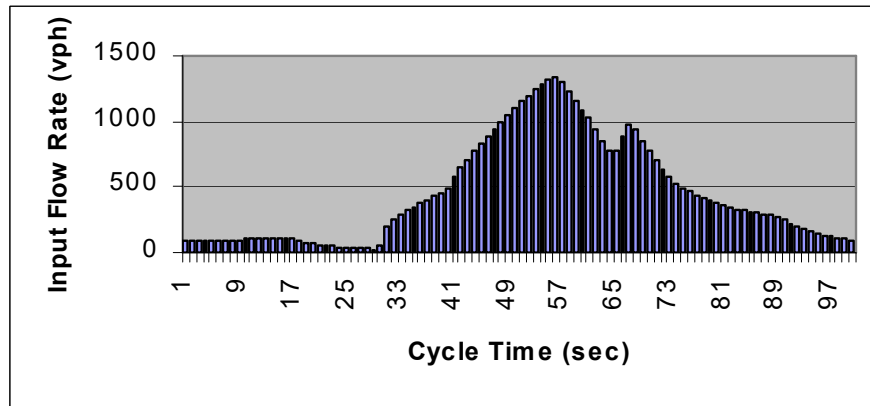
**5300 ft**

As traffic moves downstream, the initially tight platoon formed from the departing queue tends to disperse the farther downstream it travels. Because drivers tend to maintain safe headways, or spacing, between vehicles and often travel at different speeds, the platoon tends to spread out -- a few moving ahead and some dropping back. The flow rate decreases with time as the platoon reaches each point of

observation. This is why signal coordination loses much of its effectiveness if the distance between signals becomes too high (e.g., 1 mile or more).

Extending this simplified platoon dispersion example, consider the example in the figure below. In this case there are two movements upstream, and the resulting flow profile contains multiple peaks.

## Merging Platoons



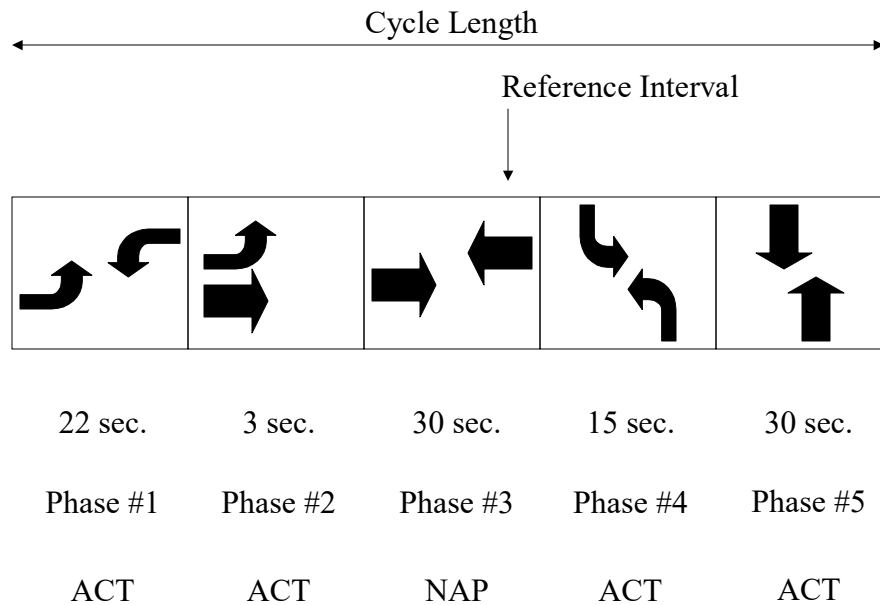
Using the node/link formulation, coupled with flow profiles and link-to-link propagation, TRANSYT can very realistically model the flows of traffic on an arterial, or in a two-dimensional network. It is important, therefore, to know not only how many vehicles arrive at the reference point of a link (that is the stopline), but it is equally important to know from whence they came upstream. As illustrated in the previous figures, TRANSYT is concerned with the origin of upstream traffic, and this is an input to the model.

This section demonstrated the nature of the traffic modeling concept of TRANSYT. All upstream inputs to all downstream links are explicitly considered in the arrival patterns downstream; thus queues are expressly considered. The platoon dispersion concept was documented by Robertson [3] and was based on work in the U.K. by Hillier and Rothery [8]. Flow profiles provide the fundamental representation of traffic flow in TRANSYT. For more technical details, refer to the Simulation Methodology section of the documentation.

## Traffic Signal Timing Elements and Characteristics

After simulation of traffic flow, the second major function of TRANSYT-7F is to develop optimized traffic signal timing plans. In link-wise simulation, only one signal cycle is simulated. This cycle and its resulting traffic operations are assumed to repeat as "average" conditions throughout the period of analysis. In step-wise simulation, the number of simulated signal cycles depends on the analysis period length, and whether or not multi-cycle simulation has been selected. Some of the more important signal timing elements are illustrated in the figure below and described thereafter.





### Cycle Length

The cycle length is the amount of time during which all movements at a signalized intersection are accommodated. In a coordinated signal system the cycle length is constant for all controllers during any given control period. This is true whether the intersection controllers are pre-timed or actuated. A semi-actuated controller under system control will always return to the green of the non-actuated phase (e.g., "NAP" Phase #3 in the figure above) at regular intervals. TRANSYT only deals explicitly with fixed-length phase operations; thus, average phase times are required when the controllers are actuated. In this case, an equivalent fixed-length plan must be coded, including a fixed phasing sequence. TRANSYT-7F is capable of estimating the average phase lengths that are likely to occur, or likely to be optimal, under actuated control.

TRANSYT also allows for double cycled signals, provided that the maximum number of phases is not exceeded. In this case certain controllers operate on a local cycle that is one-half the system cycle length. Thus, the timing at a double cycled intersection will be repeated twice within one system cycle. The program will actually recommend when this treatment may be beneficial, based on a user-supplied threshold.

When using the step-wise simulation process, uncoordinated operations (in which different cycle lengths are used at different intersections) may be simulated and/or optimized.

### Phasing Sequence

A typical signal cycle may be composed of from two to six "single-ring" phases, depending upon the traffic movements requiring protection during their respective green periods. The previous figure illustrates five phases, with overlapping left-turns in the East/West direction. In TRANSYT, the maximum number of single-ring phases is seven. There are usually a maximum of six vehicular traffic phases, plus an optional exclusive pedestrian phase for a typical four-legged intersection. This exclusive pedestrian phase should not be confused with the all-red clearance interval. The all-red traffic clearance intervals are not exclusive pedestrian phases.

Phasing sequences may consist of numerous combinations of protected and permitted

movements. While any phase sequence may be specified in TRANSYT, the traffic engineer must exercise professional judgment as to which sequences are most practical (or safe) for the intersections under consideration. The flexibility available in TRANSYT allows the user to model any practical phasing sequence, including standard NEMA dual-ring phasing, left-turn conditional service, true and quasi double cycling (the latter being alternating phases of unequal lengths), permitted lead-lag left-turns (the so-called "Dallas phasing"), and so forth.

Engineering judgment is particularly needed in the case of permitted left-turn operations (assuming right-hand drive). Multiple phase sequences may be either protected-only or protected-plus-permitted. Some of the latter combinations, notably lead-lag sequences, are sometimes not good practice. Specifically, allowing a permitted left-turn in the middle (overlap) phase of a lead-lag phase sequence may leave the left-turner trapped in the middle of the intersection, if no gap was available during the permitted phase.

### Interval and Phase Lengths

As discussed above, the cycle is normally composed of two to six phases. The amount of time devoted to each phase is referred to as the phase length (split). A phase is composed of two or more signal intervals. The distinction between these terms is explained as follows:

1. An interval is a segment of the cycle during which all signal displays, both traffic and pedestrian, are unchanged.
2. A phase is the combination of intervals during which the traffic movements given the right-of-way during the green and change intervals are unchanged. If several intervals provide right-of-way for pedestrians only, this is also a phase.

Thus, a phase consists of two or more intervals that are generally expressed in seconds, or percent of cycle length. In the case of actuated control, phase lengths may be either coded by the user, or may be calculated by TRANSYT-7F.

Interval definitions within TRANSYT-7F may indicate green, yellow, or all-red intervals. When pedestrians are present, the green interval may be further subdivided into the walk interval, and the flashing don't walk interval. The green interval may also be subdivided into fixed green intervals (that cannot be optimized), and a variable green interval (that can be optimized).

### Offsets and Yield Points

The offset is normally the time from some system reference time point to the beginning point (reference interval) of the cycle at each signal controller in the system. Offsets are generally determined so that, to the extent possible, traffic can flow through a number of signals without stopping.

In TRANSYT, an offset (or yield point for controllers coded as actuated) may be referenced to the start of any interval in the cycle. In the case of uncoordinated operations, in which different cycle lengths may be used at different intersections, offsets may not apply, unless specific clusters within the network share the same cycle length. For isolated intersections and uncoordinated intersections, offsets are not applicable.

## **Traffic Signal Timing Optimization**

Optimization consists of a series of trial simulation runs, using the TRANSYT simulation engine. Each simulation run is assigned a unique signal timing plan by the optimization "umbrella", or processor. The optimizer typically applies the hill-climb and/or genetic algorithm searching strategies. The trial simulation run resulting in the best performance is reported as optimal. Prior the optimization project,

field studies related to travel times and platoon dispersion, as well as advance considerations for a signal timing review process [24], are recommended.

### Optimization Strategy

Prior to a signal timing project, an optimization strategy should be carefully considered. This optimization strategy should reflect the engineer's goals for the project.

Elimination of oversaturated conditions is often a logical goal, because this is usually the best way to minimize vehicle delay and fuel consumption. However, what if one or more intersections are guaranteed to be oversaturated due to insufficient laneage, regardless of signal timing? If the sum of flow ratios for critical lane groups,  $Y_c = \sum (v/s)$ , is greater than 1.0 at a given intersection, then one or more queue lengths at this intersection will constantly be increasing throughout the analysis period. An optimized signal timing plan could minimize the amount of oversaturation, but would never be able to eliminate the oversaturation. (The exception to this rule is phasing sequence optimization, because the  $Y_c$  calculation is affected by the phasing sequence.) If an intersection is destined for oversaturation, the engineer might elect to allow side streets to fail, giving extra green to the major streets so they can flow smoothly. One-touch HCS analysis would be a good way to **identify any intersections with inadequate laneage**. The "Capacity and LOS Worksheet" in HCS provides the  $Y_c$  computation.

The **analysis period duration** and **number of time periods** are also key aspects of the optimization strategy. Conventional wisdom indicates that if the traffic network is undersaturated throughout the peak hour, a signal timing plan should be developed to accommodate the peak 15-minute period. Indeed, if a signal timing plan can effectively handle the peak 15-minute period, it can probably also handle the less congested 15-minute periods as well. On the other hand, if the traffic network is severely oversaturated, it is known that vehicle delay and other performance measures cannot be computed accurately unless a multi-period simulation is performed. Moreover, any multi-period simulation must be undersaturated at the beginning of the first time period, and undersaturated at the end of the final time period, in order for delay and other performance measures to be computed accurately. This situation often presents a dilemma for the engineer, because multi-period models require more input data, more time, and more effort.

After the analysis period duration and number of time periods have been chosen, it is always advisable to perform a **simulation of existing conditions**, prior to any optimization. The reason for this is to ensure that the traffic network data have been coded correctly, and that the software is capable of reasonably replicating field conditions. If the traffic network data have been incorrectly coded, or if the software is incapable of reasonably replicating field conditions, then any optimization efforts will be futile (i.e., "garbage in, garbage out"). The engineer should be familiar with vehicle delays and queue lengths in the study area, in order to determine whether existing conditions are being modeled properly. It would then be possible to increase volumes or change laneage to optimize for future predicted conditions, if desired. Of course, for one or more brand-new traffic signals, simulation of existing conditions might not be feasible.

The optimization **objective function** is the next key aspect of the optimization strategy. One could choose to minimize delay, or choose to minimize fuel consumption, or choose to maximize progression, or choose to maximize vehicle trips, etc. Ironically, although perfect progression between intersections often appears as effective signal timing in the street, it is known that perfect progression does not minimize delay or fuel consumption. In addition, choosing an objective function is sometimes related to the analysis period duration and number of time periods that were chosen. For example, if the engineer chose to optimize a severely oversaturated network with only one time period, they would know that progression between intersections might not be possible, and that accurate computation of delay or fuel consumption might not be possible. In this case, maximizing the number of vehicle trips might be the best strategy.

The **underlying analysis engine or simulation engine** is the next key aspect of the optimization strategy. There are numerous advantages and disadvantages when considering macroscopic TRANSYT

optimization versus microscopic CORSIM optimization. TRANSYT optimization runs much faster on the computer, but direct CORSIM optimization increases the likelihood of observing a smooth-running traffic network in TRAFVU animation. In addition, direct CORSIM optimization does not offer phasing sequence optimization. In certain cases it may be advisable to optimize a network using both TRANSYT and CORSIM. For example, TRANSYT could quickly perform hill-climb cycle length optimization, and then that cycle length could be locked in prior to a more-detailed CORSIM optimization of splits and offsets.

The issue of **sequential or simultaneous optimization of signal settings** is another important consideration. With regard to the cycle length, phasing sequence, splits, and offsets, sometimes it best to optimize all of these simultaneously. Other times it is best to optimize them sequentially, or to hold certain settings constant. Also with regard to holding certain settings constant, it is often very helpful to omit well-performing intersections from the optimization node list, for better efficiency. Simultaneous optimization of all signal settings tends to work best for small traffic networks, or on fast computers. When performing sequential optimization, it is best to optimize cycle length and/or phasing sequence first, and then optimize the splits and offsets last. The logic behind this design sequence is that the cycle length and phasing sequence are "major" signal settings that control what the splits and offsets need to be. Splits and offsets are "minor" signal settings that can usually be fine-tuned without disrupting the optimized cycle length and phasing sequence. But anytime the cycle length and phasing sequence are changed, the original splits and offsets are no longer applicable, so they must be re-designed.

The **initial timing flag** can greatly assist the optimization process in some cases, but can interfere with the optimization process in other cases. Naturally, the optimization process tends to be quicker and/or more effective when the initial timing plan is as close as possible to the global optimum solution. The initial timing flag activates the initial timing "model", a quick-and-dirty optimization technique for pre-assigning green time according to the volume-to-saturation flow rate ratio of each movement. The initial timing flag tends to help the optimization process in cases where actuated controllers are present, or when the user-coded timing plan is ineffective. Time permitting, it is sometimes helpful to try optimization runs with and without the initial timing flag, to see which approach is more effective. The initial timing flag can only be used in conjunction with the TRANSYT engine, although the HCS-Signals "quick estimation procedure" contains a similar initial timing model of its own.

Finally, the optimization **search technique** is another aspect of the optimization strategy. There are numerous advantages and disadvantages when genetic algorithm optimization versus hill-climb optimization. Hill-climb optimization runs much faster on the computer, but genetic algorithm optimization increases the likelihood of locating the global optimum solution. In addition, hill-climb optimization does not offer phasing sequence optimization, or uncoordinated optimization. In certain cases it may be advisable to optimize a network using both hill-climb and genetic algorithms. For example, one could quickly perform hill-climb cycle length optimization, and then that cycle length could be locked in prior to a more-detailed genetic algorithm optimization of splits and offsets.

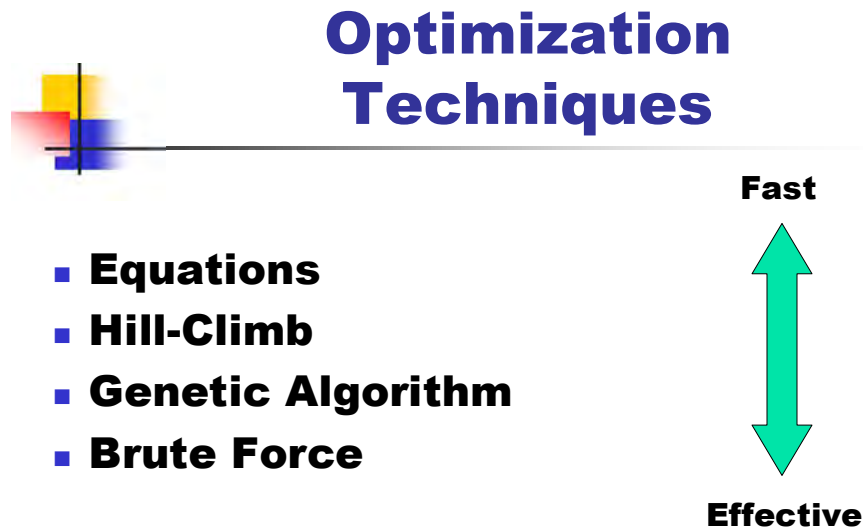
Thus, an intelligent traffic signal timing project could involve the following basic considerations:

1. identify any intersections with inadequate laneage
2. choose the analysis period duration and number of time periods
3. perform a simulation of existing conditions
4. choose an optimization objective function
5. choose an underlying analysis engine or simulation engine
6. choose between sequential or simultaneous optimization of signal settings
7. decide whether or not to use an initial timing model
8. decide which intersections to omit from the optimization node list
9. choose an optimization search technique

### Optimization Search Techniques

The genetic algorithm is mathematically better suited for determining the absolute or global optimal solution, relative to hill-climb optimization. On the other hand, the genetic algorithm generally

requires longer program running times on the computer, relative to hill-climb optimization. Purely analytical equation-based optimization and "brute force" optimization are available in certain software packages but not TRANSYT-7F. Given the speeds of today's computers, brute force optimization (which evaluates every mathematically feasible timing plan) is generally not yet practical for anything beyond isolated intersections. The figure below illustrates the relative computational speed and efficiency of these four known optimization search techniques.



Overview of Optimization

### Performance Index

When optimizing, TRANSYT-7F maximizes (or minimizes, depending on the selection) an objective function called the performance index (PI). The objective function is selected by the user. The best objective function to use depends on the desired operational characteristics of the system under consideration.

### Standard TRANSYT-7F Disutility Index (DI)

The "standard" delay and stops DI is a linear combination of these measures, qualitatively defined as:

$$DI = [Delay (veh - hr) on a link \bullet a link - specific delay weighting factor] + [a system - wide "stop penalty" \bullet stops (veh - hr) \bullet a link - specific stops weighting factor]$$

The "stop penalty" establishes the relative importance of delay and stops, and is user-selectable. This formulation can also be specified so that the weighting factors are the same as those in the fuel consumption model. This results in a DI that is explicitly excess fuel consumption. Optionally, the user can add a penalty to the DI to minimize the effect of queue spillback.

### Excess Operating Cost

TRANSYT-7F can explicitly optimize excess operating cost, although total operating cost is also reported as a measure of effectiveness. The formulation is somewhat similar to the DI above, but more factors enter into the estimate, and the coefficients are naturally different.

### Progression Opportunities (PROS)

The concept of "forward link opportunities" [9,10] was first introduced in release 7. A forward progression opportunity is defined as the ability presented at a given point in time to enter one intersection on green (including the change period) and have the expectation to travel at the desired progression speed through the next downstream intersection without stopping, independent of other traffic.

Each such opportunity available during a given period of time (e.g., a "step" in TRANSYT) is tabulated as one progression opportunity. Multiple opportunities, both in time and space, are accumulated as "PROS". The number of PROS in a given direction, for a given time period (or step), are the number of successive green signals that will be encountered at the design speed without stopping. The aggregate PROS is found by summing the PROS over all time periods in both directions. A subset of the PROS concept is to consider only the through PROS in each direction. Optimization of this subset would have an effect similar to optimizing for maximal bandwidth.

### Objective Functions for Congested Conditions

When conditions are severely oversaturated, multi-period optimization of the disutility index (DI) is usually preferable. This is because progression (PROS) is probably not feasible with vehicles stopping at each intersection, and because multi-period simulation becomes the only way to accurately compute delay and stops. Unfortunately, multi-period modeling is sometimes impractical, given the time constraints and field data constraints that engineers must sometimes deal with. For this reason, several queuing and throughput-based objective functions have been made available. These objective functions are specially designed for mitigating severe congestion problems (with or without spillback), in the context of a single-period optimization.

Congested links should be identified for further investigation. A number of links may have congestion problems (with or without spillback), but it is recommended that the focus be on solving congestion problems of critical links. A critical link is the most downstream link of a continued series of links that have congestion problems (in other words, the first to experience congestion problems).

The objective functions for single-period optimization of severely oversaturated conditions include:

1. Minimizing the traditional disutility index (the DI, which is a combination of stops and delay) multiplied by the sum (over all links) of the ratio of the average back of queue on the link (in vehicles) during the simulation period, to the maximum number of vehicles that can be accommodated on the link, i.e., queuing capacity.
2. Maximizing the ratio of the throughput (the sum of the vehicles leaving the stopline) on selected links to the DI.
3. Initially maximizing the throughput on selected links, then minimizing the DI without reducing the throughput.
4. Maximizing the throughput on links with a penalty imposed if a maximum v/c ratio threshold is exceeded on any link.

In addition, the DI optionally includes a penalty that multiplies the number of steps with spillback for each internal link. This DI can be used either alone or as part of any of the objective functions listed

above.

The objective functions above can only be used in conjunction with step-wise simulation. Link-wise simulation is not suitable because it is not capable of simulating congested conditions.

### Optimization Objective Functions

The objective function in TRANSYT-7F may be defined by the user in a number of ways as stated earlier. One of the most important parts of the objective function is the disutility index (DI), as generally defined in the equation below. The DI may be specified in any of several ways. Simply stated, the options are:

$$DI = \left[ \begin{array}{l} \text{Standard delay and stops} \\ \text{Excess fuel consumption} \\ \text{Excess operating cost} \end{array} \right] + [\text{Double count progressive links}]$$

where DI is one of the disutility indices defined previously. One of the three in the first bracket is required, but the other bracket is optional.

The performance index (PI) may thus be defined as follows:

$$PI = \left[ \begin{array}{l} \text{DI only} \\ \text{PROS only} \\ \text{PROS \& DI} \\ \text{PROS / DI} \\ \text{DI} \cdot \sum_i \frac{\text{Average back of queue on link } i}{\text{Queuing capacity on link } i} \\ \text{Throughput only} \\ \text{Throughput \& DI} \\ \text{Throughput / DI} \end{array} \right]$$

For PROS formulations, the PROS may be weighted with respect to any of the following, as applicable:

- PROS on individual arteries and/or direction, and
- PROS versus DI.

As a result of these options, the user has a great deal of power in defining the TRANSYT-7F PI. The case study from MOST Volume 4 chapter 9 [1] has an example of using different objective functions to optimize a given system, and selecting the solution that produces the best results.

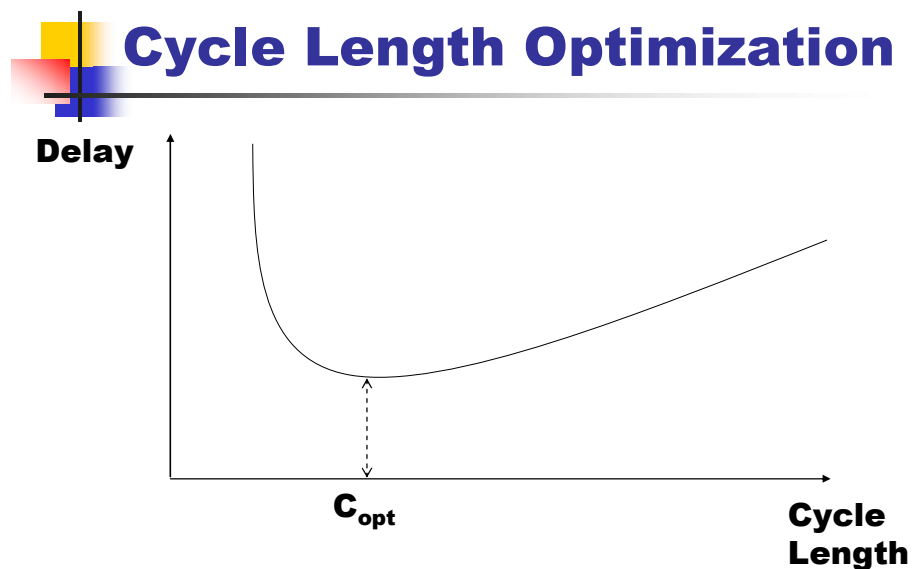
As a rule, the higher the PI value, the better the system performs. This applies to both positive and negative PI values (e.g., the PI value based on the "DI \* Queuing Ratio" objective function is always negative). The exception to this is "DI Only", for which the lowest values are considered optimal.

### Cycle Length

Cycle lengths should be long enough to provide sufficient minimum times for all phases, considering both vehicle and pedestrian requirements. Ideally, cycle lengths should be long enough to

ensure that no traffic movement is saturated. That is, the degree of saturation should be less than 100% for all approaches at all intersections. This constraint generally requires a higher cycle length than the vehicle and pedestrian requirements alone. However, the cycle length should not be so long as to cause unacceptably high delays and/or queues that would cause spillback conditions. In a progressive system, the cycle length should be long enough to facilitate traffic progression. This constraint often requires a background cycle length that is somewhat higher than the cycle lengths that would minimize delay at the individual intersections. In a progressive system, cycle lengths should also be identical amongst all intersections so the timing plans can remain synchronized, although low-volume intersections can be "double-cycled".

If an isolated signalized intersection has enough lanes to service the incoming traffic volumes, it can be mathematically demonstrated that one and only one optimum cycle length exists for the intersection. The figure below illustrates this minimum delay cycle length. If there are nearby intersections that might be affected by platoon progression or queue spillback, the mathematics of cycle length optimization become too complicated, and simulation-based optimization becomes necessary. If a signalized intersection does not have enough lanes to service incoming traffic volumes, or if it becomes completely blocked by queue spillback, the exact value of the cycle length becomes somewhat irrelevant.



Overview of Optimization

High cycle lengths are helpful if they are needed to eliminate oversaturation (i.e., all queues served during one cycle), or if they are needed to serve oncoming platoons of vehicles in a progressive system. When volumes are low, low cycle lengths are better at minimizing delay because not only do they eliminate oversaturation, but the red times shown to each movement are shorter.

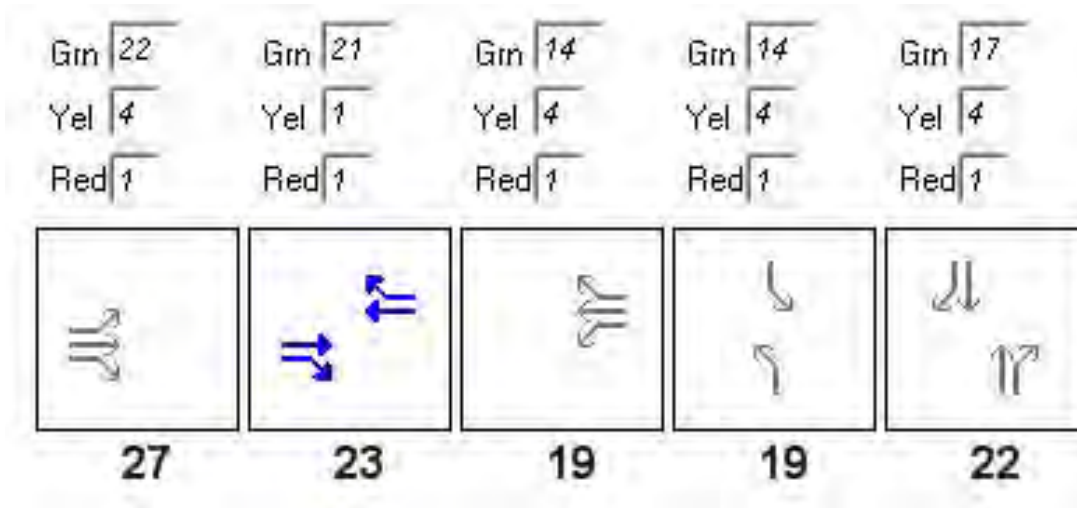
When volumes are high enough that signal timing optimization is unable to eliminate oversaturation, the percentage of green time allocated to each movement is often more important than the cycle length. Changing the cycle length would have very little effect on vehicle delay, but changing the percentage of green time allocated to each movement would have a big effect on vehicle delay. In this type of a situation, the "optimum" cycle length reported at the end of an optimization run may actually be a somewhat random or arbitrary value that lies somewhere within the user's specified range of allowable candidate cycle lengths.



Hill-climb cycle length optimization simply performs a series of split/offset optimization runs constrained to a single cycle length within the range. For example, for a cycle length range of 60-100 with a 10-second search increment, five separate hill-climb optimization runs would be performed. Genetic algorithm cycle length optimization performs only one optimization run, and is constantly experimenting with new cycle lengths during the run. Refer to the Cycle Length Optimization section of the documentation for more details.

### Phasing Sequence

Phasing sequence optimization is primarily effective at improving progression for coordinated intersections. If conditions are severely oversaturated such that progression is not possible, and vehicles must stop at every intersection, the exact phasing sequence becomes somewhat irrelevant (in comparison to the amount of green time received by each movement). Lead-lag phasing (shown below) can be useful in allocating extra green time for a particular thru movement, e.g., the eastbound thru gets more green time than the westbound thru. Lead-lag phasing can also be useful in attempting to achieve two-way progression, e.g., progression in both the eastbound and westbound directions. Phasing sequences that allow for overlapping left-turns (assuming right-hand drive) are able to give extra green time to the left-turn that has heavy traffic volumes.



Some believe that new and optimized phasing sequences should perhaps not be implemented in the field unless evidence suggests network performance will be significantly improved. This is because drivers will not be accustomed to the new phasing sequence, thus possibly creating a safety hazard.

TRANSYT-7F performs phasing sequence optimization by using the genetic algorithm. Refer to the Phasing Sequence Optimization section of the documentation for more details.

### Splits and Offsets

TRANSYT-7F can explicitly optimize splits (phase lengths) and offsets within any range of cycle lengths. Optimization of splits and offsets can be accomplished using either the hill-climb method or the genetic algorithm. The user can indicate that splits, or offsets, or both are to be optimized in a particular run. For isolated intersections and uncoordinated intersections, offsets are not applicable and cannot be optimized. For closely spaced intersections, the user may define "grouped nodes", such that the difference of offset between these intersections is held constant during the optimization process. Refer to the Grouped Nodes section of the documentation for more details.

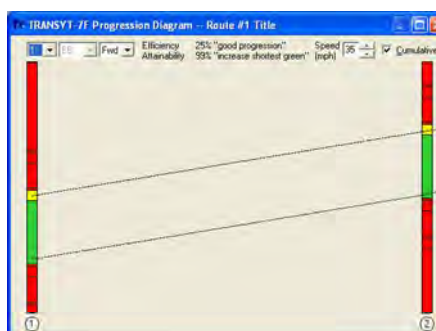
Optimized splits tend to produce degrees of saturation in the 80%-90% range for most traffic movements. This degree of saturation range tends to be optimal because:

1. When a movement's own degree of saturation rises above 90%, its own delay increases exponentially, because some signal cycles will not have enough green time to serve the queue. This forces some vehicles to wait through multiple signal cycles before being served.
2. When a movement's own degree of saturation falls below 80%, all other movements' delays increase, because they are not getting enough green time.

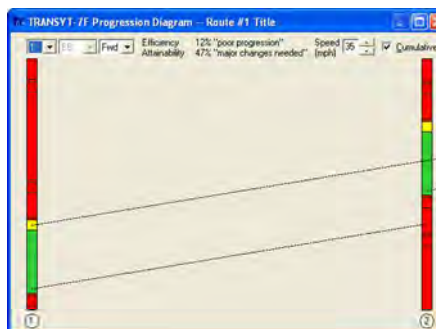
Optimized offsets generally allow traffic to flow through a number of signalized intersections without stopping. If conditions are severely oversaturated such that progression is not possible, and vehicles must stop at every intersection, the offsets become somewhat irrelevant (in comparison to the amount of green time received by each movement). In this type of a situation, the "optimum" offset reported at the end of an optimization run may actually be a somewhat random or arbitrary value.

The time-space diagrams below illustrate distance (in feet or meters) between signals along the X-axis, and cycle length time (in seconds) along the Y-axis. Offsets in the upper time-space diagram below illustrate "perfect bandwidth". However, because some vehicles travel more slowly than the average speed, this design would force some vehicles to arrive at the beginning of red. Therefore, the optimum offset design for minimizing delay and fuel consumption is actually shown in the lower time-space diagram below. In the upper diagram, the offset at node #2 (on the right) is 26 seconds higher than the offset at node #1 (on the left), because the travel time between these intersections is 26 seconds. In the lower diagram, the offset at node #2 is 40 seconds higher than the offset at node #1 (on the left), allowing all vehicles to either arrive on green or just before the beginning of green.

**Perfect bandwidth,  
Medium delay**



**Medium bandwidth,  
Minimum delay**



### Design Sequence

TRANSYT-7F supports simultaneous optimization of all signal settings (cycle length, phasing

sequence, splits, offsets) in a single run, using the genetic algorithm, but this can take a very long time on the computer. The optimization process is often more efficient when one or more signal settings can be held constant (not optimized). If hill-climb optimization is used, all signal settings except the phasing sequence can be optimized simultaneously.

When signal settings are optimized separately or sequentially, the following design sequence is recommended for best results:

1. Cycle length
2. Phasing sequence and offsets simultaneously
3. Splits and offsets simultaneously

In other words, optimize cycle length first, and eventually optimize the splits and offsets last. Time permitting, it may be helpful to repeat the same steps using the original optimal solution as the starting point. If the program is unable to locate a better solution than the original optimal solution, the design process may be complete. The logic behind this design sequence is that the cycle length and phasing sequence are "major" signal settings that control what the splits and offsets need to be. Splits and offsets are "minor" signal settings that can usually be fine-tuned without disrupting the optimized cycle length and phasing sequence. But anytime the cycle length and phasing sequence are changed, the original splits and offsets are no longer applicable, so they must be re-designed.

### Punch Files

Whenever an optimization run is performed, TRANSYT-7F has the ability to produce "punch" data files. Punch files (\*.PUN) are mostly duplicate copies of the original input (\*.TIN) file. They contain the optimal signal settings from the previous run, and may be used for subsequent analysis or additional optimization. If they are used for additional optimization, the user should select File > Save As to change the extension from \*.PUN to \*.TIN, so that the punch file doesn't overwrite the input file. Currently, punch files can only be generated for "simulation-only" runs if a) the initial timing flag is on, and b) multi-period simulation is not used.

Punch files from macroscopic, TRANSYT-based optimization are named <user filename>.PUN. Traffic-actuated signal timing estimation punch files are somewhat unique because they contain the "estimated" actuated phase times, instead of the "optimized" timings contained by most punch files. CORSIM-based optimization punch files are always called "Genetic.trf", because the software assumes that all \*.PUN files adhere to the traditional TRANSYT input file format.

## **Traffic-Actuated Control**

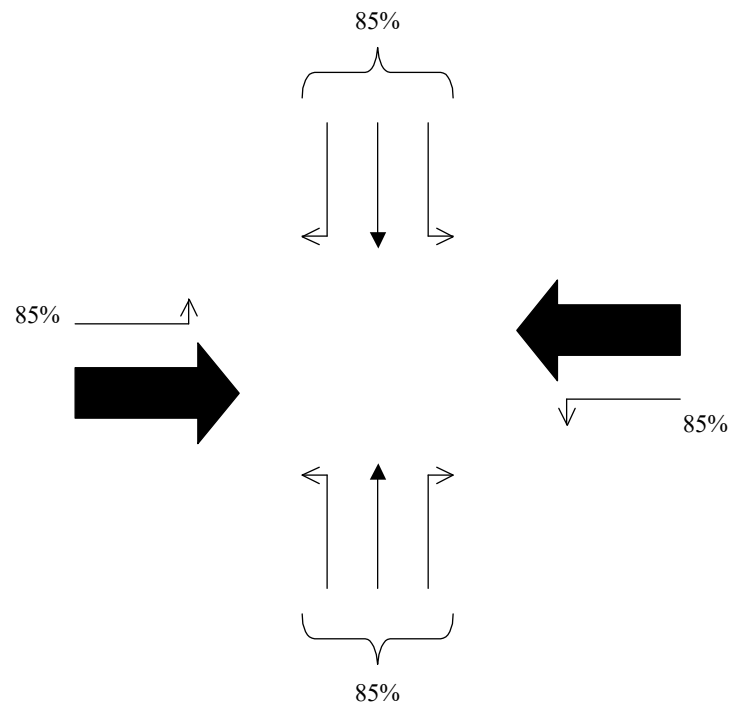
TRANSYT-7F contains two models for actuated control. The first model (target degree of saturation model) is a design model, intended to locate the actuated phase times that would optimize the traffic network. The second model (actuated estimation model) is an evaluation model, intended to evaluate existing conditions. These two models are designed to accommodate the TRANSYT macroscopic simulation model. When the CORSIM microscopic simulation model is being used to analyze actuated control, these two models are not applicable.

### Target Degree of Saturation Model

The primary application of the target degree of saturation model is optimization of coordinated traffic-actuated control plans. The target degree of saturation model is contained within the initial timing model, which is discussed in Chapter 5. Both optimization processes (genetic algorithm and hill-climb) benefit from the target degree of saturation model, because both optimization processes are significantly affected by their "starting point". That is, the optimization processes are much more efficient and effective when their starting point is as close as possible to the global optimum solution. The sample input files

"FourNode.tin" and "FourNode Lefty.tin" (installed in the TRANSYT-7F Example folder) can demonstrate the target degree of saturation model, but only if the user turns on the initial timing flag in the Edit > Analysis screen. Signal timing tables in the legacy report show that average phase times are affected by the initial timing flag.

The default target degree of saturation within the TRANSYT-7F software is 85%. Typically, the optimum degree of saturation for actuated phases is in the 80%-90% range. This target degree of saturation range tends to be optimal because 1) When the degree of saturation rises above 90%, this causes random delay (d2) to increase exponentially, and 2) When the degree of saturation falls below 80%, this either takes too much green time away from the major street through movements, or produces long cycle lengths that inflate uniform delay (d1). Refer to the Initial Timing Model section of the documentation for procedural and mathematical details. The figure below illustrates minor actuated movements operating at 85% degree of saturation, with all remaining green time in the cycle being donated to the major street through movements.



When using direct CORSIM optimization to optimize actuated control, force-offs and maximum green times are optimized explicitly, and the degrees of saturation are unknown. The advantage of this method is that the optimized force-offs and maximum green times can be easily and directly implemented in a field actuated controller. With the macroscopic TRANSYT engine, some additional engineering knowledge is required for knowing how to translate the optimized single-ring phase times into the force-offs and maximum green times that the controller requires. The disadvantage of using direct CORSIM optimization to optimize actuated control is that the program running time on the computer is much higher, and phasing sequence optimization is not available.

#### Traffic-Actuated Signal Timing Estimation

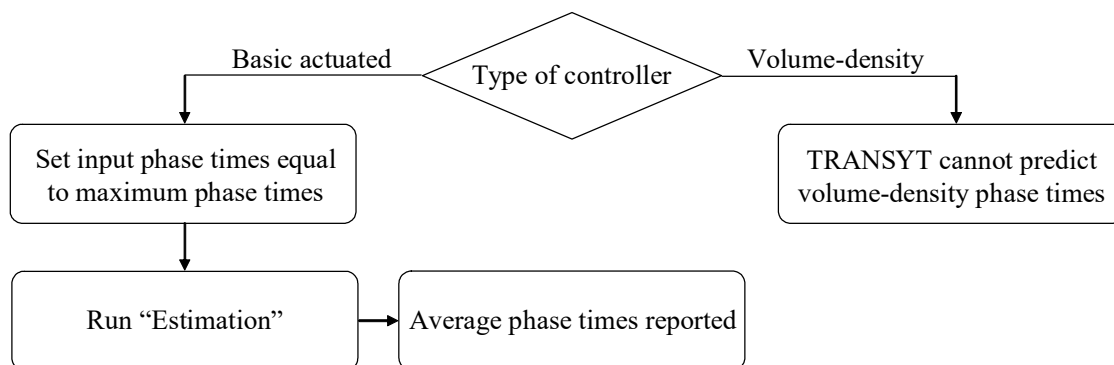
Traffic-actuated signal timing estimation is typically used to compute traffic-actuated phase times that are likely to materialize in the field, based on the user-specified maximum green and force-off settings. If the average phase times have already been measured in the field with a stopwatch, the user

can bypass this estimation model by simply coding those phase times into the software.

If average phase times have not been measured in the field, the application flowchart for evaluating existing conditions is shown below. The actuated estimation model was not designed to predict phase times for "volume-density" type control, which is sometimes used at isolated intersections. To model volume-density control the analyst would need to either measure average phase times in the field with a stopwatch, or simply guess the average phase times, before coding these phase times into the software. When coding the input phase times for basic actuated control with stopline detection, note that these phase times must be set equal to the summation of the maximum green times (or force-off locations, if coordinated) plus the yellow and all-red clearance times.

## Application Flowcharts

- **Evaluating existing conditions**
- **Given a set of maximum greens and/or force-offs, what average phase times will occur in the field?**

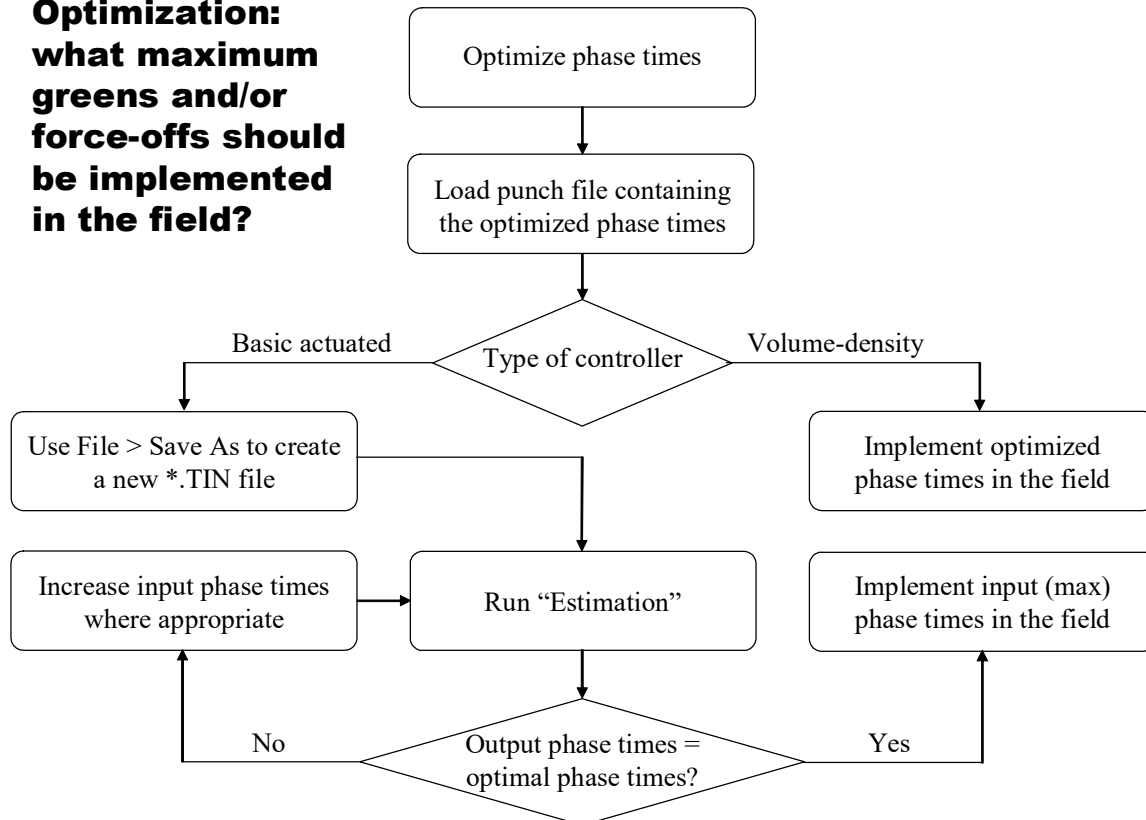


The estimation model could be used to analyze the results of an optimization run, if desired, although this amount of analysis is usually not necessary. It is usually best for maximum phase times in the field to be set equal to the optimized phase times. Following a typical TRANSYT-7F optimization run, the estimation model could be used to for two purposes. First, it could be used to confirm that the optimized actuated phase times could indeed materialize in the field. Second, it could be used to determine maximum green and force-off settings for implementation in the field, to achieve the program-recommended optimum actuated phase times.

The application flowchart for actuated control optimization is shown below. If a high level of precision is desired for phase times in the field matching the optimized phase times from the software, a trial-and-error process may be needed to determine the exact maximum greens and/or force-offs that would accomplish this. Otherwise, it is a safe bet for maximum phase times in the field to be essentially set equal to the optimized phase times. The estimation model often demonstrates that the maximum

phase times are equal to the optimized phase times, because this allows the associated movements to operate in the efficient 80%-90% degree of saturation range. When the estimation model indicates that the maximum phase times must be slightly higher than the optimized average phase times, this sometimes indicates a high-volume actuated movement that would gap out after 35 seconds or more, or that the global optimum solution was not achieved by the optimization process.

■ **Optimization:  
what maximum  
greens and/or  
force-offs should  
be implemented  
in the field?**



When using CORSIM to analyze actuated control, average phase times can be obtained by observing TRAFVU animation, such that no phase time estimation model is needed. Measuring average phase times in TRAFVU would be somewhat similar to measuring average phase times in the field with a stopwatch.

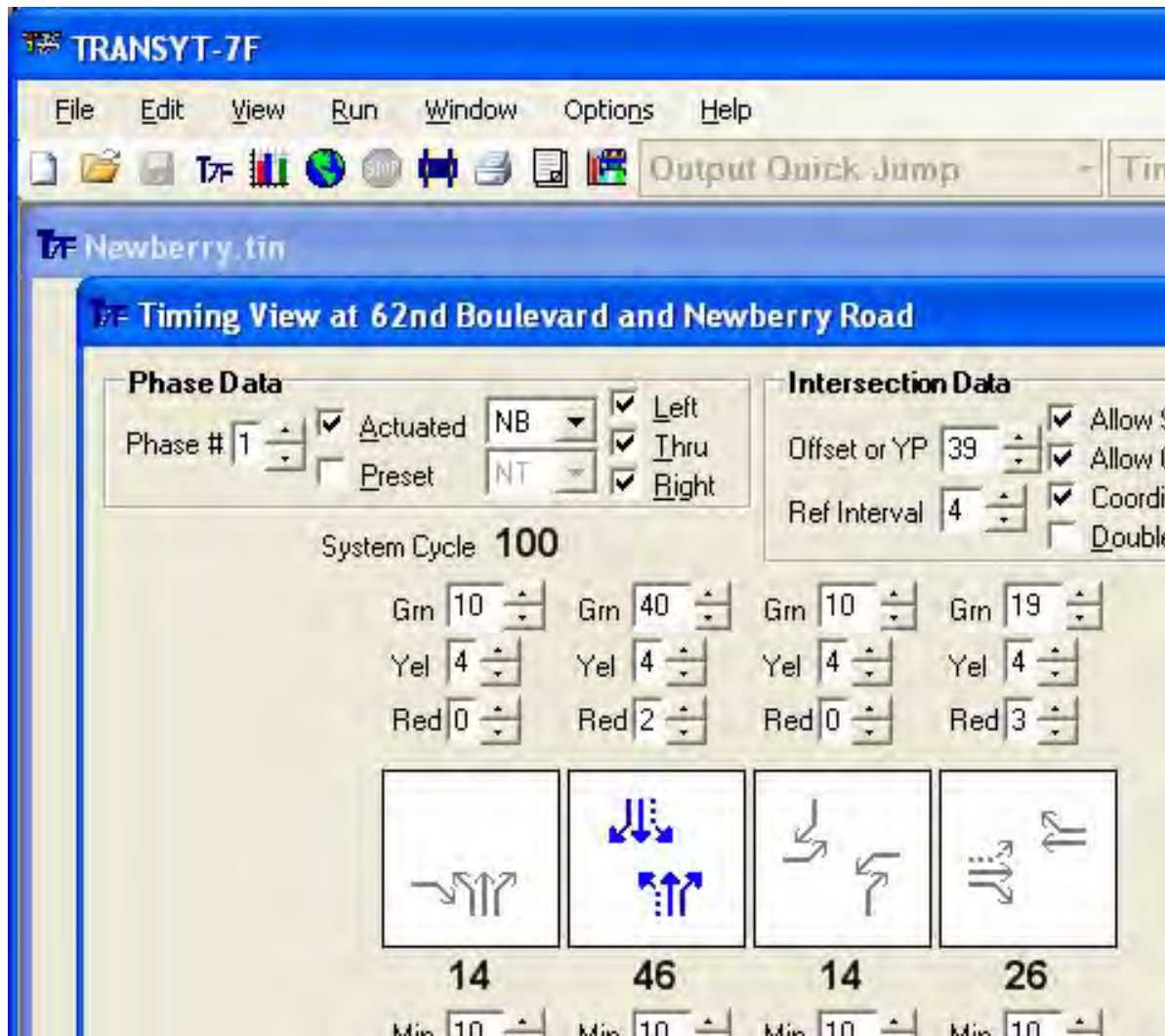
Refer to the Actuated Estimation Methodology section of the documentation for more details about the model, and refer to the Actuated Estimation Examples section for some tutorial examples.

## Permitted Movements

A permitted (also known as "give way") movement is defined as a vehicular traffic movement that must yield to opposing or conflicting vehicular or pedestrian flow. Opposing flow is typically considered to arrive from the opposite side of the intersection, whereas conflicting flow is the term used to describe vehicles arriving from a 90 degree angle.

The Timing View screen below illustrates some examples of permitted movements, for a right-

hand drive intersection. In phase #1 of the timing plan, the westbound right-turn is permitted, and yields to the conflicting northbound through movement. In phase #2 of the timing plan, the northbound left-turn is permitted, and yields to the opposing southbound through movement. The sample input file "Level2Perm.tin", which is installed within the TRANSYT-7F Example folder, demonstrates permitted movements at several intersections. TRANSYT-7F also supports left-hand driving, which sometimes has permitted right-turns along the major street.



The issue of turn "protection" is often important. Assuming right-hand drive, the most common types of left-turn protection are:

- 1) Permitted only (green ball)
- 2) Protected only (green arrow, for example, the westbound left-turn in phase 3 above)
- 3) Protected-permitted (compound phasing, for example, the northbound left-turn in phases 1 and 2 above)

It is also possible to have permitted-protected phasing, if the "green arrow" follows the "green ball". Generally, compound phasing allows for the lowest delay among the permitted turners, since they have two phases available in which to move. However, some jurisdictions opt for protected-only phasing

on major roadways, because permitted movements across multiple through lanes at high speeds can lead to serious accidents. Permitted-only phasing is the optimum turn protection when there are low permitted movement volumes, because introducing the green arrow would take green time away from the through movements.

TRANSYT provides access to a number of permitted movement models, including a unique model for simulating permitted movements on red. In the U.S. (right-hand drive), this is typically called "RTOR" (right-turns on red). Permitted right-turn movements are modeled similarly to permitted left-turns in TRANSYT. The only practical differences are in the coding. It is important to select the proper permitted movement model, and to specify the proper opposing movement(s). When coding the phasing sequence in the Timing View screen, the software draws permitted movements as dashed arrows, to distinguish between them and protected movements. Refer to the Permitted Movement Methodology section of the documentation for procedural and mathematical details.

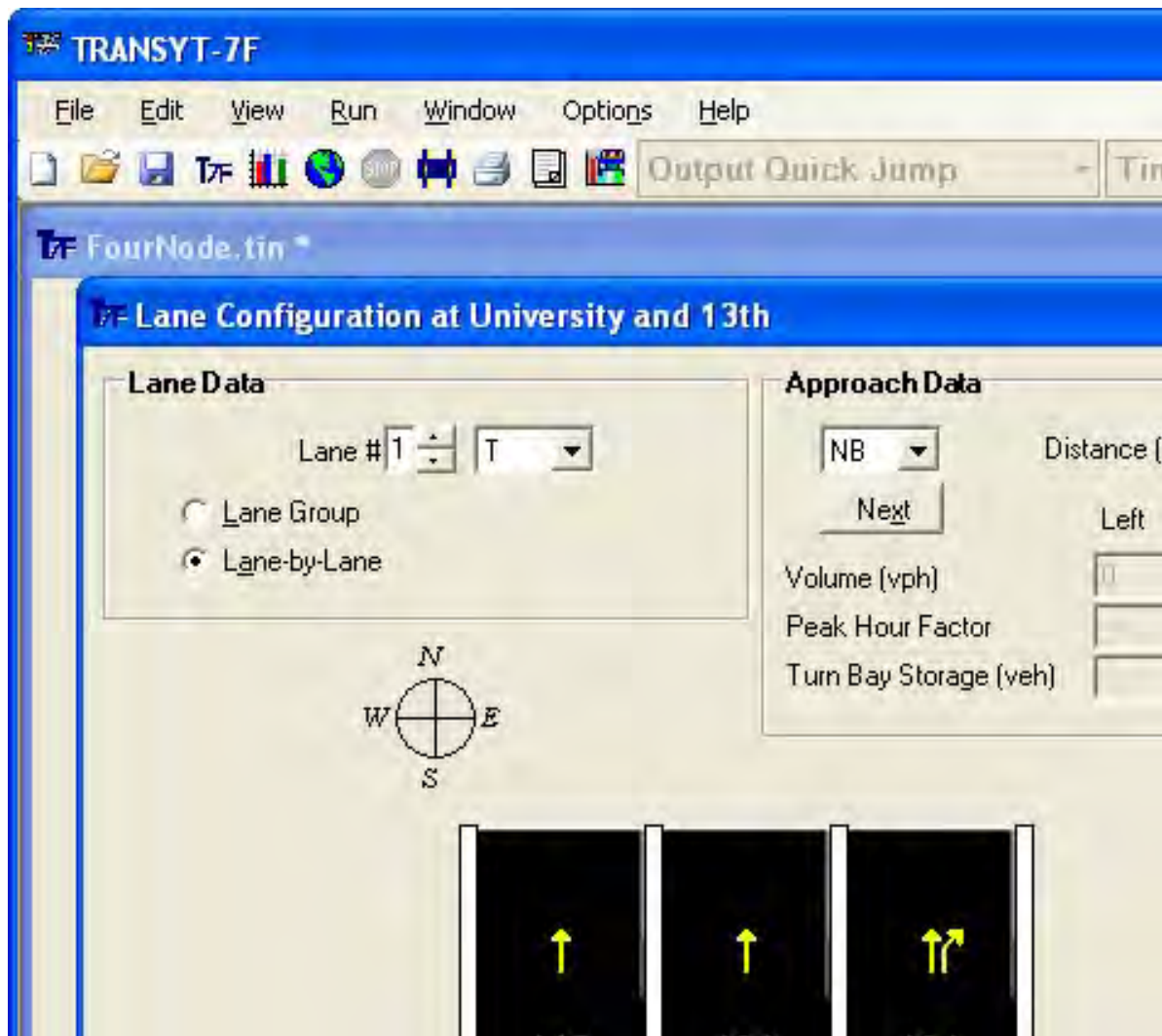
## Shared Lanes and Lane-by-Lane Analysis

In the context of traffic network study, a "shared lane" usually refers to a single lane (approaching an intersection) from which multiple vehicular turning movements (left, through, or right) can be made. A "shared lane group" usually refers to a shared lane that is modeled together with adjacent through lanes. A shared lane group is associated with one overall volume, one saturation flow rate, one queue length, one delay, and so on. In TRANSYT, turning movements from a shared lane are called "secondary" shared stopline links. Typically, the through movement is defined as a "primary" shared stopline link. In order to apply the software properly (i.e., code the input data properly) vis-a-vis shared lanes, it is helpful to understand how the model handles them.

When lane utilization is fairly even within each lane of a shared lane group, the entire lane group may be defined and modeled fairly accurately as a single link. On the other hand, highly unequal lane utilizations are modeled more accurately when a link is defined to represent each individual lane. In general, the software runs more efficiently when a single link number can be used to represent multiple lanes. This allows for lane group modeling, which is consistent with Highway Capacity Manual [12] procedures, and many other software packages. However, for special network locations in which traffic utilization of a given lane is significantly different from other lanes in the same lane group, "lane-by-lane" modeling becomes necessary for achieving realistic results. Note that secondary shared turning movements are considered to be part of the through-movement lane group, whereas exclusive turn lanes are always considered to be a separate lane group.

For example, in the Lane Configuration screen shown below, the first lane (on the far left) has been assigned a unique link number (113) by the user, because this lane has a significantly different queue length than the other two lanes (represented by link number 101). Significantly different lane utilization necessitates the ability to obtain separate model results for the leftmost lane. The right-turning movement is represented by link number 109, but this could only be confirmed by viewing the Traffic screen, because the Lane Configuration screen only displays primary link numbers. The lane configuration below actually demonstrates an effective combination of lane-by-lane modeling (link 113), and lane group modeling (the two lanes represented by link 101). Later in the data entry process, when coding the traffic volumes and saturation flow rates, it becomes necessary to remember and understand these lane and link assignments. Refer to the documentation section entitled Lane-by-Lane Analysis Example for a tutorial example.





Permitted shared lanes, in which the secondary turning movement must yield to an opposing movement, are modeled differently than protected shared lanes. Refer to the Shared Lanes Methodology section of the documentation for procedural and mathematical details.

## Multi-Cycle and Multi-Period Simulation

Prior to release 8, results from TRANSYT were based on simulation of one cycle length (referred to as "single-cycle" simulation). This did not permit modeling the effects of traffic conditions from one cycle to another. TRANSYT has always been capable of using only one cycle of traffic patterns to estimate performance for the entire analysis period. However, with single-cycle link-wise simulation, residual queues cannot be modeled. With single-cycle step-wise simulation, queue spillback can be simulated for one cycle, but this might not be sufficient if multiple cycles are required before spillback occurs. Moreover, even if spillback would never occur, multiple cycles of simulation would be needed to accurately estimate queue lengths, delays, and other measures of effectiveness for oversaturated conditions.

The multi-cycle and multi-period simulation options were implemented within TRANSYT to

facilitate accurate modeling of oversaturated conditions. Before utilizing these procedures, the user should be aware of some practical disadvantages. First, simulation of additional cycles will always require longer program running times on the computer. Program running time is sometimes an issue for optimization, which may require hundreds or thousands of iterative simulation runs. Second, the usage of multi-period simulation requires additional input data from the user, above and beyond what a "traditional" (single-period) analysis would require.

### Multi-Cycle Simulation

TRANSYT provides the "multi-cycle" simulation option to allow simulation of the complete time period through a sequence of cycle lengths. This takes into account cycle-to-cycle effects, such as the accumulation of residual queues from one cycle to the next. When conditions are steady-state and completely undersaturated, results may be nearly identical under single-cycle and multi-cycle simulation, because the traffic flow patterns do not change between cycles.

Although multi-cycle simulation reflects oversaturated conditions more accurately than single-cycle simulation, it contains a modeling limitation of its own. Specifically, in order to accurately measure delay and other measures, it is necessary to know when the residual queues were finally dissipated. If residual queues remain on the network following multi-cycle simulation, it is not possible to predict the delay experienced by these vehicles without simulating the next time period. This is where multi-period simulation comes into play.

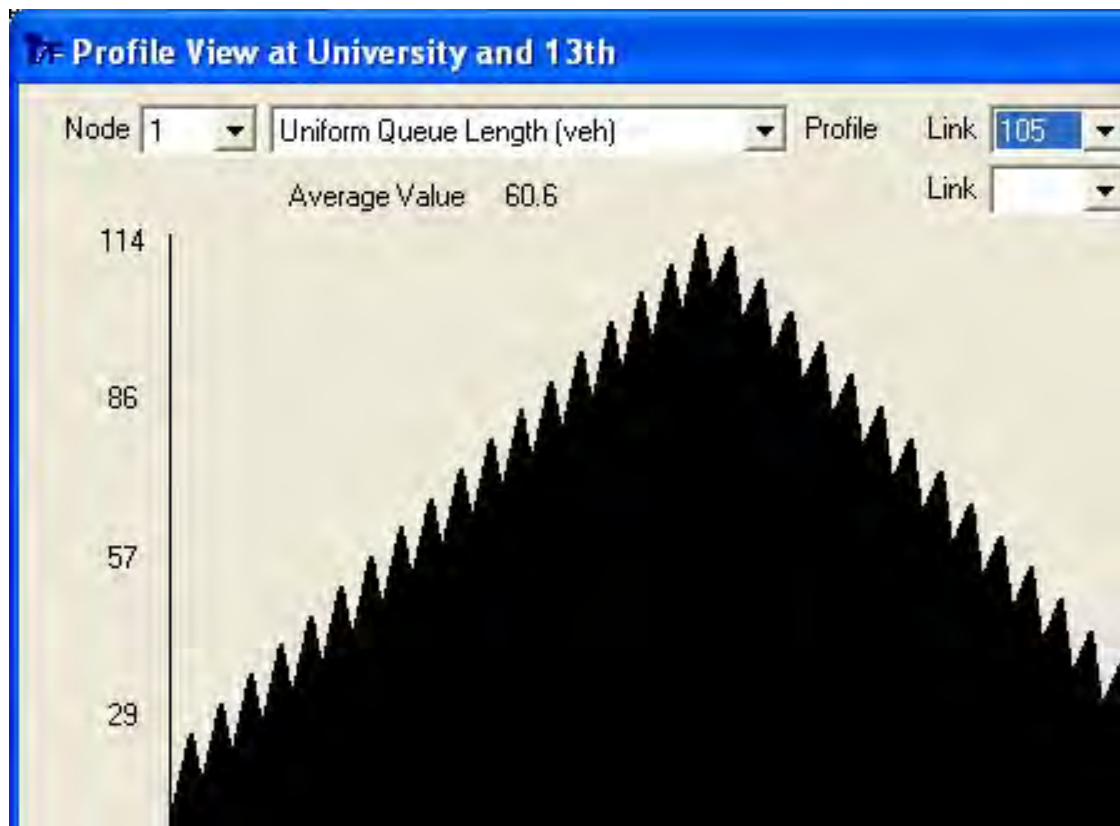
### Multi-Period Simulation

The user can request multi-period simulation to analyze systems with time-varying conditions, including traffic volume and signal timing plan variations. When conditions are steady-state and completely undersaturated, results may be nearly identical under single-period and multi-period simulation, because traffic flow patterns from the first time period do not affect the second time period.

Time periods can be defined in at least a couple of different ways. For example, each day typically contains three time periods of peak traffic volume (AM peak, PM peak, midday peak), with off-peak periods of low volume in between. Thus it would be possible to simulate an entire day by defining the time periods in this manner. Another way to define time periods would be to select an individual time frame such as the PM peak, and to subdivide it into multiple time periods of varying traffic volume demands.

The typical multi-period analysis for a signal timing project divides the peak hour(s) into 15-minute time periods of varying traffic volume demand. Under this scenario, it would be possible to develop an optimized timing plan to mitigate the residual queuing that occurs across multiple time periods. For an accurate model, the first time period must be chosen such that there are no residual queues at the very beginning, and the last time period must be chosen such that there are no residual queues at the very end. This allows delay (and other measures of effectiveness) to be measured accurately, even though the time period boundaries during the middle of the analysis experience residual queues. If it is also necessary to analyze another time frame within the day, such as the AM peak, a second input data file is recommended for this analysis.

For example, the Profile View screen image below plots four 15-minute time periods. There is no residual queue at the beginning of time period #1 or at the end of time period #4. For conditions such as this, a single-cycle or single-period style analysis would be unlikely to accurately predict delay (for this movement) without knowing the conditions of the upcoming time periods. At the end of the first cycle or first time period, there are residual vehicles remaining on the network, and the traffic model could not accurately predict how much longer they will be delayed. However, when multi-period simulation is performed, delay is easily computed for this movement by determining the area under the "queue accumulation polygon", which in this case stretches across all four time periods.



Thus, the most common (and recommended) application of multi-period simulation is to examine the effects of oversaturated traffic volumes and residual queues on subsequent time periods, where only the user-coded volumes change between time periods. Multi-period simulation is not recommended for analyzing undersaturated conditions. When conditions are undersaturated, not only would single-period simulation produce the same results as multi-period simulation, but it would run more quickly on the computer, and the analyst would not need to code input data for the extra time periods.

Another (less common) application of multi-period simulation is to analyze signal timing plan transition between time periods, usually when conditions are oversaturated. Although TRANSYT can simulate timing plan variation (cycle length, splits, and offsets) among time periods, the phasing sequence must remain constant during all time periods. Timing plan transition analysis is also not needed when conditions are undersaturated. If residual queues are not passed on from one time period to the next, it is more effective to perform separate single-period analyses using separate input data files.

Multi-period simulation can be used to analyze basic traffic-actuated control, where the green times change from cycle to cycle in response to traffic volume variations. However, the user should not define separate time periods for each signal cycle, because the model is designed to simulate (and optimize) the average actuated phase times and volumes. For example, suppose the volume is higher in time period #2 than time period #1, for a given phase. The average actuated phase time for time period #1 might be 16 seconds, whereas the actuated phase time during time period #2 could be 21 seconds. In time period #1, the average phase time of 16 seconds would be simulated during each cycle, but the stochastic models (random delay, random queue, etc.) within TRANSYT would account for phase time variations occurring in the field.

The actuated estimation model cannot directly process multi-period data files. Since the purpose of actuated estimation is to calculate the (undersaturated) actuated phase times, this purpose could be accomplished by simply making one or more single-period copies of the multi-period data file, before

running actuated estimation.

The input data screens in TRANSYT-7F support the creation and editing of multi-period data files. When coding link connections in the Map View, this invokes automatic flow balancing for all time periods of a multi-period input file. A sample input data file for multi-period simulation called ("FourNode TP.tin") is installed within the TRANSYT-7F Example folder. TRANSYT-7F optimizes multi-period operations by using the genetic algorithm. TRANSYT-7F can also import and optimize multi-period CORSIM files. For more technical details, and tutorial examples, refer to the section of the documentation entitled "Multi-Period Analysis Examples".

## Uncoordinated Operations

TRANSYT can simulate uncoordinated operations in which different cycle lengths are used at different intersections. Intersections explicitly marked as "uncoordinated" are always simulated with uniform flow profiles, because there is no reason to expect vehicle platoons to arrive during any particular point within the signal cycle. If an intersection has a different cycle length than the network's background cycle length, but is still marked as "coordinated", it could still experience platooned vehicle arrivals from nearby signals.

Uncoordinated operations can be optimized as well as simulated. Uncoordinated operations can only be optimized using the genetic algorithm. At the uncoordinated intersections, splits and cycle lengths can be optimized independently of the network background cycle length. For isolated intersections and uncoordinated intersections, offsets are not applicable and cannot be optimized. Phasing sequence optimization, although allowed at isolated intersections, is primarily effective at improving progression at coordinated intersections.

Although multiple cycle lengths may be simulated within one network, and although multiple cycle lengths may be optimized if some intersections operate in isolation, the software's optimization process expects only one background cycle length (or one coordinated signal system) within any one data file. If the user wishes to optimize multiple coordinated signal systems, it may be necessary to do this using multiple input data files.

For more technical details, refer to the sections of the documentation entitled "Uncoordinated Operations Methodology" and "Uncoordinated Optimization".

## Sign Control

TRANSYT can explicitly model two-way stop-controlled (TWSC) intersections and yield-controlled intersections. They are coded with only one phase in the timing plan. The user can imply the difference between TWSC and yield control by calibrating the maximum flow rates on permitted links. On the major street uncontrolled links, if field observations indicate interference from traffic on the sign-controlled links, saturation flow rates could be adjusted accordingly (average cruise speeds may also be affected). The sign-controlled links would be assigned saturation flow rates determined from field measurements, or from other capacity estimates.

For signal timing and signal analysis, TRANSYT can implicitly model the effects of roundabouts and/or all-way stop controlled (AWSC) intersections on nearby signals, by eliminating platoons between signals. Roundabouts and AWSC intersections typically eliminate any platooning occurring along the major street, resulting in a uniform flow profile. One way to model roundabouts and AWSC intersections is to delete the link that connects two signals. Another way to accomplish this without deleting links is to mark the adjacent nodes as "uncoordinated signals". Output measures of effectiveness (MOEs) for roundabouts and AWSC intersections are not estimated or provided.

Refer to the Sign Control Methodology section of the documentation for procedural and mathematical modeling details, and refer to the Sign Control Examples section for tutorial examples.

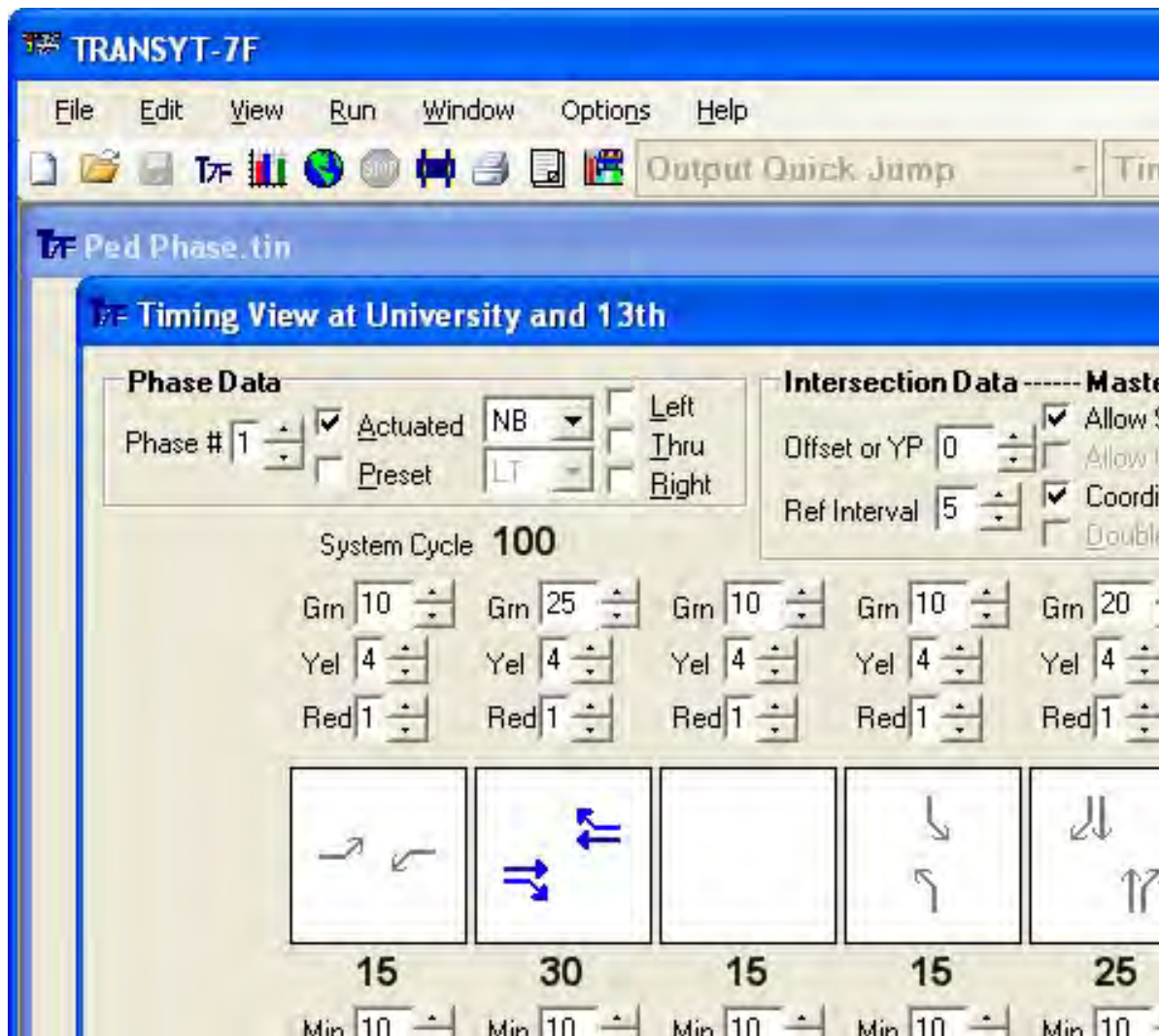
## Mid-Block Sources

Parking lots, shopping centers, freeway off-ramps, and other facilities that serve as mid-block sources of traffic demand are modeled as uniform input flows to the link. The value coded for the uniform flow should be the total output volume from the source. TRANSYT automatically accounts for flow losses into such facilities, so it is not necessary to code any mid-block "sink" volumes that might exist in the field. As a general guide, a mid-block source should be modeled if its output constitutes 10% or more of the total volume on a link, or if it adds more than two vehicles per cycle. Refer to the following sections for more details on the impact of mid-block source volumes: Traffic Screen, Automatic Flow Balancing, and Platoon Dispersion.

## Pedestrians, Bicycles, and Exclusive Pedestrian Phases

When pedestrians and bicycles move together with vehicles in the same phase, saturation flow rate adjustments are used to reflect the impact on traffic operations. For more details, refer to the Satflow View and Saturation Flow Rate Calculator sections of Chapter 3.

There are two ways to model an exclusive pedestrian phase. The first method is to code an empty phase with no moving links. The second method is to code a separate phase with a low-volume (10-vph) traffic movement, using an extra link (e.g., link 113). The optimization process will likely force such an exclusive pedestrian phase length to its minimum specified duration, especially if no pedestrian links with pedestrian traffic are modeled. Because of this, the minimum phase length specified should be sufficiently long to ensure that the variable green interval is not reduced to an unsafe length. This screen image below illustrates an exclusive pedestrian phase, which is the third phase in the timing plan.



## Bus and Carpool Operations

Buses can be explicitly or implicitly modeled by TRANSYT. Explicit modeling is accomplished by assigning separate links to the buses. These "bus links" may be entirely separate to simulate bus-only lanes, or they may "share" a lane or lanes with the auto traffic link to simulate buses traveling in mixed mode with other traffic. A provision is included to take the dwell time at bus stops into account. Priority treatment can be given to buses by applying high individual weights to the delay and/or stops (that is at traffic signals, not at bus stops) for the bus links. Carpools that are assigned separate lanes can be modeled similarly to buses, except no dwell times are coded.

For most practical purposes, and especially for shared bus lanes, implicit modeling is recommended instead of explicit modeling. In this case, an adjusted saturation flow rate is coded to account for bus blockage, in accordance with Highway Capacity Manual [12] guidelines. Performance estimates from the model, as well as optimized timing plans, can then appropriately reflect the impact of bus operations. If the number of buses on the link is relatively high, the user could optionally increase "average vehicle spacing" within the model, to account for the effect of vehicle length on queuing. Whenever there is a possibility of queue spillover or queue spillback, model results are potentially

sensitive to the average vehicle spacing.

## **Bottlenecks**

Often there are mid-block restrictions such as narrow bridges, lane drops, curbside parking, etc., that reduce mid-block capacity below the saturation flow rate of the downstream links. TRANSYT has the capability of modeling these bottlenecks.

A bottleneck is modeled as a dummy node (i.e., no node number is assigned), and the affected links effectively have 100% "green" time. Saturation flow rate on the affected link is then specified at the appropriate value, usually the capacity of the bottleneck. Refer to the Traffic Screen documentation for more details.





# Chapter 3: TRANSYT-7F Input Data

---

## TRANSYT-7F Input Screens

### Input Data Screen Overview

The TRANSYT-7F input screens allow data to be coded without directly editing "record types" (refer to the Record Type Format section of the documentation for details). The input data screens generate these record types in the background, but provide a more intuitive user interface.

It is possible to generate a fully functional TRANSYT data set simply by creating a new file (using File > New), and then using the first seven edit screens (Map, Lanes, Traffic, Timing, Satflow, Feeders, Analysis). The remaining edit screens (Global, Route, Timing, Queuing, Actuated, Permitted, Weighting, Capacities, Intersection Names) provide additional input data that is considered optional. In some cases the Traffic screen could be considered optional, because the Lanes screen can automatically generate traffic volume data. In some cases the Feeders screen could be considered optional, because the Map View can automatically generate feeding link data. In some cases the Satflow screen could be considered optional, because adjusted saturation flow rates could be coded directly into the Traffic screen.

The Map View is available for displaying network geometry, input data, and output data. The Map View is activated by selecting Edit > Map, or by pressing the F3 key when a TRANSYT-7F input file is loaded. The Map View is different from other input screens in that most of the data is drawn graphically instead of being coded numerically. Beyond input coding, the Map View also serves the parallel purpose of displaying certain output data graphically. Strictly speaking, use of the Map View is also optional, because the program can perform an error-free run without any node coordinate information. However, use of the Map View tends to facilitate quicker input coding and better understanding of the overall analysis. For more details, refer to the Map View section of the documentation.

When coding a new TRANSYT input file from scratch (after using File > New), it is logical to use the seven primary edit screens in the order that they appear on the Edit menu, with one important exception. After dragging and dropping nodes to their appropriate positions on the Map, link connections should not be drawn until after volumes have been coded on the Lanes and/or Traffic screens. Drawing link connections on the Map after volumes have been coded allows the program to perform "automatic flow balancing". Later on in the data entry process, it becomes possible to bypass the Feeders screen, because feeding flow information has already been coded by automatic flow balancing. Moreover, if all volumes are coded on the Lanes screen, it is possible to skip volume coding on the Traffic screen.

Finally, if all saturation flow rates are known, these can be coded directly into the Traffic screen, and the Satflow screen can be skipped entirely. For a tutorial example, refer to the section entitled "Creating a Brand-New Input Data File".

There are a few general rules that apply to the input data screens:

- 1) When input data are modified, note that the file name at the top of the graphical user interface (GUI) now has an asterisk next to it, to the right. This is called the "dirty file flag", which indicates that input data have been modified since the file was last saved. Input data are generally modified by pressing Tab after a numeric input parameter has been changed, or by using the mouse to add and delete intersections, lanes, and phases. To save an input file for future analyses it is necessary to select File > Save, press Ctrl-S, or click on the floppy disk icon at the top of the GUI.
- 2) Another flag exists called the "dirty run flag". Although invisible to the user, the dirty run flag gets activated whenever input data are modified, and gets deactivated whenever a run is performed. The dirty run flag is used to alert the user of scenarios where an output file might become inconsistent with its associated input file. If a file is about to get closed while the dirty run flag is still on, the GUI will warn the user that a run has not been performed recently, and give the user an opportunity to cancel out of closing the file.
- 3) By default, pressing the Enter key will close the screen, while saving all changes. The alternative behavior for the Enter key is to simply move the cursor to the next data input field. This behavior can be specified in the Options > Preferences screen.
- 4) Pressing the Escape key, clicking on the red "X" in the upper right hand corner of the window, or clicking on the Close button, will close an edit screen. If the user wishes to close an edit screen immediately after coding a numeric value, it is preferable to first press Tab before closing the screen, to ensure that the numeric value is saved.
- 5) Pressing Page Down, pressing Page Up, or using the Node Jump combo box automatically loads data for the next or previous intersection in the network. This only applies to intersection-specific screens (e.g., Traffic) and is not applicable to screens that do not relate to individual intersections (e.g., Analysis).
- 6) The Next Links/Prior Links pushbuttons automatically load data for the next/prior 12 links at the intersection. TRANSYT allows simulation of 50 links per intersection. For simple analyses involving 12 or fewer links per intersection, this feature may not be important; but for advanced lane-by-lane analyses, or analyses of intersections having numerous approaches, this feature will be necessary.
- 7) The Next Data/Prior Data pushbuttons load new data that are applicable to the same screen. For example, the Queuing screen contains 12 items of input data, but only displays six at a time.
- 8) The Duplicate RT pushbutton automatically generates a duplicate record type in the data set. This function is only applicable to certain screens (i.e., Global, Route, Weighting).
- 9) The following input screens allow some or all of their data to be edited in a tabular format, with rows and columns of numeric input fields: Lanes, Traffic, Timing, Satflow, Feeders, Permitted, Queuing, Actuated. In these screens, the Tab key switches fields horizontally (left-to-right), whereas the up and down arrow keys switch fields vertically (top-to-bottom). If any input values are modified, the up and down arrow keys will activate the dirty file flag, just like the Tab key.

The Add/Delete Intersection selection, available from the Edit > Optional > Intersection menu, allows the user to select an intersection number for which all associated record types will be automatically added or deleted. Intersections can also be added and deleted graphically on the Map View. The Intersection Names screen, available by selecting Edit > Optional > Intersection > Names, allows the user to specify street names (or alternate node numbers) to be displayed on the interface and in the printed reports. Node numbers can be changed by selecting Edit > Optional > Intersection > Node Numbers, and

then specifying the old/new node numbers.

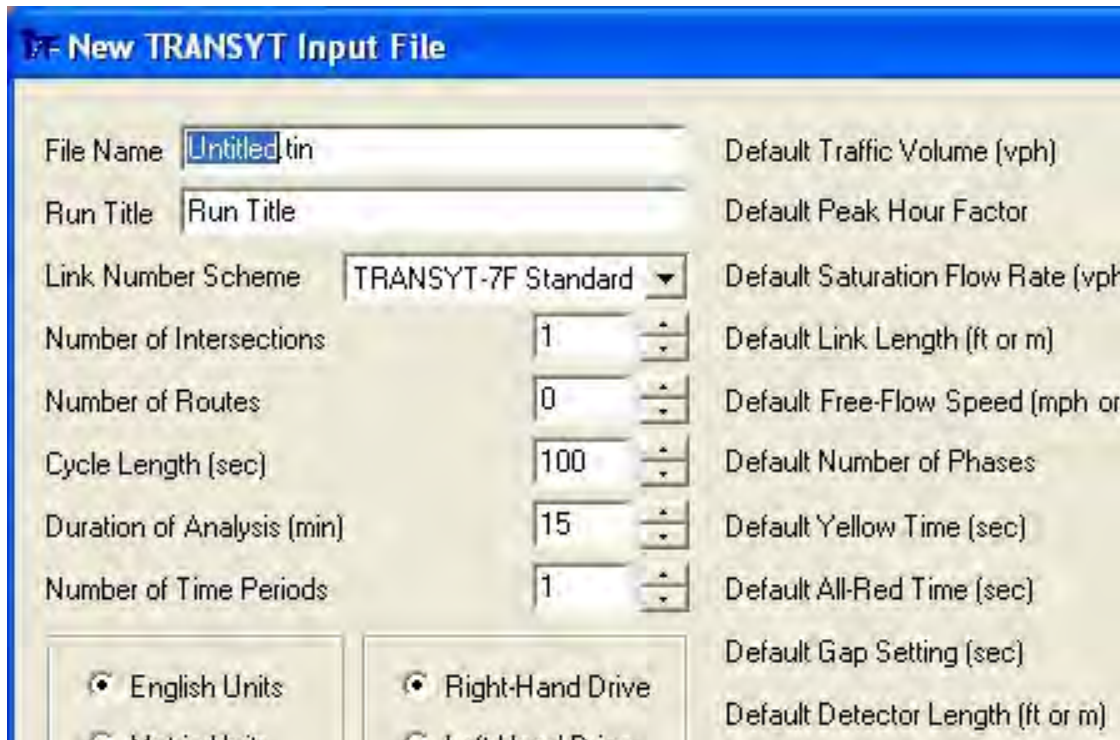
The user can further customize interface behavior (behavior of the Enter key, and automatic link lengths) in order to speed up the data entry process.

When performing a multi-period analysis, optional input data (under the Edit > Optional menu) are generally not expected to vary between time periods. This is why the Optional menu is disabled when editing data for time periods #2 and higher.

Each of the input data screens provides context-sensitive help, through use of the F1 key.

## New TRANSYT Input File

The image below illustrates a sample New File dialog screen, as rendered by the TRANSYT-7F graphical user interface (GUI). For a tutorial example, see the section of the documentation entitled "Creating a Brand-New Input Data File".



### File Name

The file name typed here will contain the input data specified on the remainder of the New TRANSYT Input File dialog box. Most of the software package is designed to expect the "\*.TIN" extension. The user should change the "Untitled" part of the file name, but the "TIN" extension should be left alone.

### Run Title

The run title appears at the top of the record type input file, and also at the top of output reports.

### Link Number Scheme

The scheme specified here will be applied to the new data set. For more information, see the Link Numbering Scheme section of the documentation.

### Number of Intersections

The number of intersections specified here will be created within the new data set. Later on, the user may add or delete intersections in the Map View, or by using the Edit > Optional > Intersection > Add/Delete menu selection. If the number of intersections specified here is greater than one, the master controller will be node #1 by default. The master node number can be edited at any time on the Edit > Optional > Global > Network Parameters screen (or record type 10).

### Number of Routes

The number of progression routes specified here will be created within the new data set. However the links contained within each route would need to be coded by the user on the Edit > Optional > Route screen (or using the Map View) later on. It is not mandatory to define any routes within the data set, but routes are sometimes useful when using the PROS objective function, or when time-space diagrams are desired.

### Cycle Length

The overall system cycle length for the new data set is specified here. Later on, the user may either change the overall system cycle length, or instead change the cycle length at individual intersections (uncoordinated operations). For more information, see the Traffic Signal Timing Elements section of the documentation.

### Duration of Analysis

The duration of simulation is specified here. According to conventional wisdom, 15 minutes is an appropriate default value, because this is the amount of time for which traffic conditions can be assumed to remain constant. Later on, the user may change the duration of analysis on the Edit > Analysis screen.

### Number of Time Periods

The user can request "multi-period" simulation to analyze systems with oversaturated conditions and residual queues. The software currently allows a maximum of 9 time periods per data file. Later on, time periods may be added or deleted by selecting Options > Time Period > Add/Delete.

### English Units

The units of output for the printed reports are specified here. If the check box is turned off, results will be reported in metric units. If necessary, the entry here can be changed later on the Edit > Analysis screen.

### Right-Hand Drive

This check box indicates that motorists are assumed to be driving on the right-hand side of the road. If the check box is turned off, the software will assume that motorists drive on the left-hand side of the road. This piece of information affects several aspects of the software. For example, if right-hand driving is selected, the new data set will contain (by default) exclusive left-turn lanes, along with shared through and right-turn lanes. If left-hand driving is selected, the new data set will contain (by default) exclusive right-turn lanes, along with shared through and left-turn lanes. Later on, the user can modify the lane geometry (e.g., which lanes are shared and exclusive) on the Edit > Lanes screen. The

information also affects appearance and behavior of the Timing View screen, and whether leading and lagging left-turns or right-turns will be optimized in the context of phasing sequence optimization. If necessary, the entry here can be changed later on the Edit > Optional > Global > Run Controls screen.

### Default Traffic Volume

The default traffic volume for each turning movement is specified here. Later on the user will need to code the correct hourly traffic volumes on the Edit > Lanes screen, or the Edit > Traffic screen. Hourly volumes (vph) are typically specified, even if the duration of analysis is only 15 minutes. An option is also available (through Options > Preferences) to specify volumes in units of vehicles per time period (vpp). Refer to the Preferences Screen section of the documentation for more details.

### Default Peak Hour Factor

The default peak hour factor (PHF) for each turning movement is specified here. Later on the user will need to code the correct peak hour factors on the Edit > Lanes screen. When the peak hour factor is lower than 1.00, it implies that the user wishes to model the peak period within the hour, which experiences a somewhat higher flow rate than the hourly volume would indicate. It is usually calculated using the following formula:

$$PHF = \frac{\text{hourly volume (vph)}}{4 \times (\text{peak 15 minute volume})}$$

The peak hour factor should be coded as 1.00 under the following conditions:

- the duration of analysis is 60 minutes or more
- a multi-period analysis
- volumes are being specified in units of vehicles per time period (vpp)

### Default Saturation Flow Rate

The default saturation flow rate to be used in the new data set is specified here. This value should be the "adjusted" saturation flow rate instead of the "ideal" saturation flow rate. Later on, the user may modify saturation flow rates for individual movements on the Edit > Traffic screen, or by using one of the saturation flow rate edit screens (Edit > Satflow, or Edit > Optional > Capacities).

### Default Link Length

The default distance between intersections is specified here. Later on, the user may change individual link lengths on the Edit > Lanes screen, or the Edit > Traffic screen. Link lengths potentially have a significant impact on the modeling of platoon dispersion and queue spillback, throughout the network.

### Default Free-Flow Speed

The default free-flow speed for the new data set is specified here. This value affects fuel consumption, and has a significant impact on the modeling of traffic progression. Later on, the user may modify free-flow speeds for certain links (on the Edit > Feeders screen), or the default external link free-flow speed (on the Edit > Optional > Global > Network Parameters screen).

### Default Number of Phases

The default number of "single-ring" phases for the new data set is specified here. The default number of phases should be within the range of 2 through 7. Later on, the user may add or delete phases from the timing plan at each individual intersection by using the Edit > Timing screen. For more

information, see the Traffic Signal Timing Elements section of the documentation.

#### Default Yellow Time

The default yellow interval time to be used in the new data set is specified here. Later on, the user may modify yellow times at each individual intersection by using the Edit > Timing (View) or Edit > Optional > Timing screens.

#### Default All-Red Time

The default all-red clearance time to be used in the new data set is specified here. All-red times are optional, such that a coded value of zero seconds is acceptable. Later on, the user may modify all-red times at each individual intersection by using the Edit > Timing (View) or Edit > Optional > Timing screens. In addition, the user can optionally combine the yellow and all-red times into one interval, using the Timing View or Timing screens. Note that combining these intervals and interval durations into one should not have an effect on the modeling or the results. For more information on signal intervals, see the Traffic Signal Timing Elements section of the documentation.

#### Default Gap Setting

The default gap setting (a.k.a. "unit extension") to be used in the new data set is specified here. Later on, the user may optionally modify the gap settings for certain links on the Edit > Optional > Actuated screen. The gap setting affects the calculation of random vehicle delay, and can potentially affect the computation of actuated phase times, if the actuated estimation model is used. If the new network to be created is fully pre-timed, the gap setting value is irrelevant to the model, and will be ignored.

#### Default Detector Length

The default presence detector length to be used in the new data set is specified here. Later on, the user may optionally modify the detector lengths for certain links on the Edit > Optional > Actuated screen. The detector length can potentially affect the computation of actuated phase times, if the actuated estimation model is used. If the new network to be created is fully pre-timed, or if the actuated estimation model is not used, the detector length value is irrelevant to the model, and will be ignored.

## **Map View (associated with record types 70 and 71)**

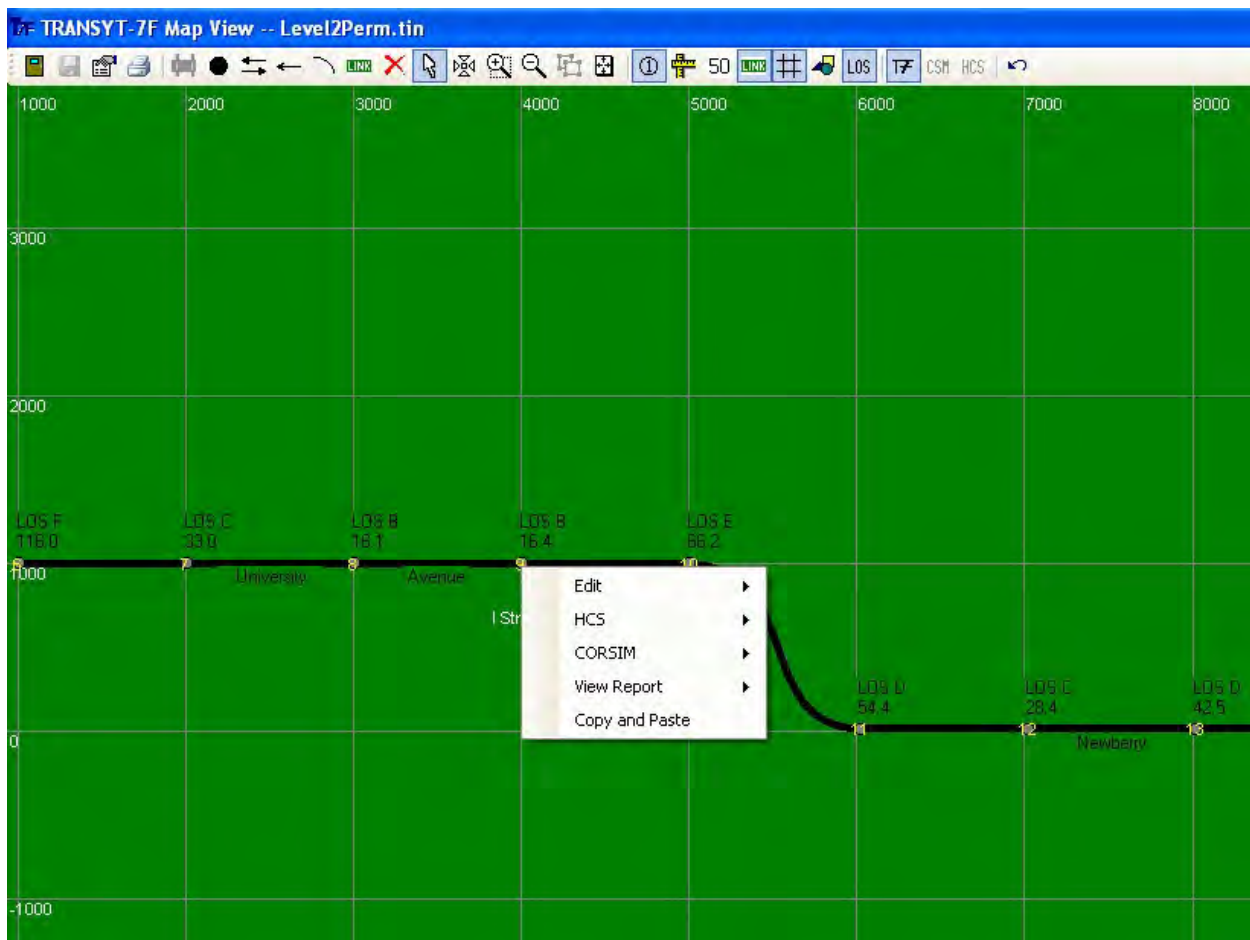
A map rendering screen ("Map View") is available for displaying network geometry, and editing network input data. Use of the Map View is optional because input data can be coded on the other edit screens, and because node coordinates do not affect results from the model. However the Map View may allow the user to better understand the data entry process, which could indirectly improve results from the model, and may assist in explaining the model to unfamiliar parties. In addition, the ability to display input and output data on the map can be useful for verification and validation of the model.

The Map View is activated by selecting Edit > Map, or by pressing the F3 key when a TRANSYT-7F input file is loaded. In the background, node coordinate and link curvature data are saved into (and read from) the \*.TIN file on record types 70 and 71. The Map View currently offers the following features:

- Link curvature
- Printing support
- Dynamic scrolling of the map
- Drag and drop nodes on the map
- Zoom in, zoom out, pan, show all
- Graphical add/delete of links/nodes

- Select and move entire groups of nodes
- User-friendly bitmap background scaling
- Right-click on nodes to edit their properties
- Nine available map background colors to choose from
- Progression routes drawn with bold lines for emphasis
- Right-click on a node and "View Report" for this node
- "Undo" feature for canceling recent changes to the map
- Right-click on a node and "View HCS results for this node"
- View intersection names by moving the mouse pointer over a node
- Right-click on a node and "View CORSIM animation for this node"
- Define progression routes automatically by selecting a group of nodes
- Ability to underlay user-supplied bitmap backgrounds and aerial photos
- Display level of service on the map for TRANSYT, CORSIM, and HCS
- Graphical coding of progression routes, arterial weighting factors, and route titles
- Copy-and-paste intersections, to avoid re-coding lane configuration, volumes, and timings
- Link lengths, free-flow speeds, link names, gridlines, intersection delay and LOS on the map

A sample Map View is illustrated below.



### General Information and Basic Map Behavior

In a brand-new data file, since network geometry has not been specified yet, all nodes will appear in the middle of the map. Subsequently the user can drag and drop the nodes into their appropriate positions. Link connections between nodes are drawn based on connecting link data from the Feeders screen. If intersection names have been coded on the Edit > Optional > Intersection > Names screen, they can be displayed when moving the mouse cursor over a node.

In general, appearance of the map is controlled by dragging and dropping nodes, or by using the toolbar (near top of screen) containing several icons. Individual nodes or groups of nodes may be dragged and dropped anywhere on the map. Any previously coded link lengths will not be modified by dragging and dropping nodes, so the exact location of nodes is not critical. Sometimes, nodes can be dragged and dropped more precisely by deactivating any superimposed output data on the map, because the software will not allow node placement directly above the superimposed output data. After finishing with the node placement, the superimposed output data can be re-activated at any time.

The Map View coordinate system is specified in units of "twips", which are a unit of measurement on the computer screen. If English units are in effect, a ratio of one foot per twip is recommended, simply for ease of understanding. If metric units are in effect, a ratio of one meter per twip is recommended. For example, in the sample file FourNode.tin that contains quarter-mile signal spacing, the distance between nodes in the Map View is 1320 twips. In the corresponding metric unit file FourNode Lefty.tin, the distance between nodes in the Map View is 402 twips, which represents 402 meters. However, the user could apply any desired scaling, such as 10 feet per twip, 0.5 meters per twip, and so on. Since the node coordinate system has no effect on model results, precise placement of nodes is not required.

Node coordinates may range between a minimum value of -9999 twips, and a maximum value of 99999 twips. This applies to both the horizontal and vertical (X and Y) node coordinates. Coordinates of the lower left-hand corner of the map, as viewed on the computer screen ("Screen Origin"), must also fall within this range of -9999 to 99999. If a bitmap background image is being used, its lower left-hand location ("Bitmap Origin") must also fall within this range. When navigating the map via Zoom Out or Pan, if a warning message pops up that says "You have reached the map boundaries", this implies that the specified navigation would result in an out-of-range Screen Origin or Bitmap Origin.

Whenever the Map View is initially loaded, the "Pointer" icon is automatically activated on the toolbar toward the top of the screen. The Pointer icon is located directly to the right of the red X ("Delete" icon), and directly to the left of the white cross ("Pan" icon). When the Pointer icon is activated, it allows the user to select individual nodes (or groups of nodes), drag-and-drop nodes, and right-click on nodes to edit their properties. Other icons on the toolbar may be used to superimpose information on the map (e.g., link lengths, level of service), code new data (e.g., add or delete links and nodes, define progression routes), or change the view (e.g., zoom in, zoom out, show all). Sometimes when using the toolbar icons, the mouse cursor will change shape accordingly. For example, if the "Two-Way Link" icon is selected, the mouse cursor changes to that of a two-way link for subsequent drawing. To return to the default Pointer icon at any point during the session, the user can either right-click on the general map area, or click back on the Pointer icon. Right-clicking on the general map area is often more convenient, because then it is not necessary to move the mouse back-and-forth between the toolbar and the middle of the map.

The Pointer icon can be used to draw a "rubber band" around groups of nodes. To do this, left-click on the general map area and then hold the mouse button down without releasing. Then, drag the rubber band around a group of nodes, and then finally release the mouse button. Any nodes within the rubber band area should become highlighted. This group of nodes can then be dragged-and-dropped across the map simultaneously, lined up to have the same X or Y coordinate, or defined as a new progression route (as described later). After finishing manipulation of any group of nodes, make sure to "unselect" them by clicking anywhere on the general map area.

Available "hotkeys" include F1 (help), F12 (quit without save), Escape (exit without save), U (scroll up), D (scroll down), L (scroll left), R (scroll right), Z (map properties dialog or progression route



dialog), Ctrl-S (save), and Ctrl-Z (undo). The Z key will load the Map Properties dialog. If a progression route is currently selected, the X key will load the Progression Route dialog. These two pop-up screens will be described in more detail later. In order to scroll the map more quickly, simply press the <Alt> key in addition to the first letter of the four directions (<Alt>-U, <Alt>-D, <Alt>-L, and <Alt>-R). However, the map can be scrolled more precisely by using the horizontal and vertical scrollbars.

### Map Scrolling and Dynamic Scrolling

The Map View offers "dynamic scrolling", which means that the entire network can be dragged in any one of four directions (up, down, left, and right). In order to scroll the map, simply use the horizontal scrollbar at the bottom of the screen, or the vertical scrollbar at the right edge of the screen. Dynamic scrolling simply changes the X and Y coordinates of the Screen Origin, without changing the location of any links and nodes, and without changing the origin of the bitmap background (if one has been loaded). Because of this, dynamic scrolling makes it appear that the observer is traveling across the map, as if they were flying above in an airplane.

Dynamic scrolling can sometimes occur very slowly on a relatively slow computer, or with a relatively large network, or a relatively large bitmap background. Because of this, dynamic scrolling can be deactivated on the Map Properties dialog screen. The Map Properties dialog can be loaded by clicking on the Properties icon, which is the third icon from the left on the toolbar. Deactivating dynamic scrolling will result in "static scrolling", in which the map position is not altered until the user releases the scrollbar button.

By dragging the scrollbar button, the Screen Origin X and Y coordinates can be changed very quickly between their minimum and maximum values of -9999 and 99999. "Analog scrolling" can be achieved by clicking in the white space that surrounds the scrollbar button, or by clicking on the arrows at the ends of both scrollbars. Clicking in the white space increments the Screen Origin by 1000 twips, whereas clicking on the arrows increments the Screen Origin by 100 twips. Analog scrolling can also be accomplished by clicking on the "Pan" icon, which is described later.

### One-Touch HCS Analysis

If *HCS+* is installed on the computer, right-clicking on a node will allow the user to "View HCS results for this node". If *HCS+* is installed on the computer, and a pre-existing HCS file (\*.XHS) resides in the same folder as the TIN file, right-clicking on a node will allow the user to "Link to HCS". If "View HCS results for this node" is selected, a brand-new HCS file is automatically generated and loaded by HCS. If "Link to HCS" is selected, the user can select an HCS file to be associated with the current node number. This causes the HCS filename to be automatically coded as an Intersection Name. Once an HCS filename has been coded as an Intersection Name, it becomes possible to right-click on that node in the future and "Open with HCS", which will automatically launch HCS-Signals to contain the associated data file. In addition, once an HCS filename has been coded as an Intersection Name, this makes it possible to display HCS level of service on the map at a later stage. Finally, even if *HCS+* is not installed on the computer, the user can always right-click on a node and select "Generate HCS file for this node".

### One-Touch CORSIM Animation

If TSIS-CORSIM is installed on the computer, right-clicking on a node will allow the user to "View animation for this node" or "Generate input and output files for this node". If "View animation for this node" is selected, a brand-new CORSIM file is automatically generated, simulated by CORSIM, and animated by TRAFVU. To request one-touch CORSIM animation for the entire network instead of just a single intersection, use the TRANSYT-7F GUI instead of the Map View. Even if TSIS-CORSIM is not installed on the computer, the user can always right-click on a node and select "Generate TRF file for this node".

### **Status Bar Items**

## Mouse

Coordinates of the mouse cursor are displayed here. The mouse coordinate values are updated whenever the mouse cursor is moved across the general map area. The mouse coordinate values may not be updated if the mouse cursor is moved across a map label, or if the mouse cursor is moved outside the general map area.

## Zoom Factor

The Zoom Factor value is updated whenever the user zooms in or out, or possibly when the Undo and Show All icons are clicked. When the Zoom Factor is equal to 1.0, there is a one-to-one correspondence between X and Y coordinates in the TIN file (on record type 70) and the twips on the computer screen. If the user zooms in, the Zoom Factor increases. For example, a Zoom Factor of 2.5 implies 250% magnification of the map. Likewise, zooming out decreases the Zoom Factor. When English units are in effect, the minimum Zoom Factor is 0.25, and the maximum Zoom Factor is 8.00. When metric units are in effect, the minimum Zoom Factor is 0.75, and the maximum Zoom Factor is 25.00. These zoom ranges are intended to support a one-to-one correspondence between twips/feet or twips/meters. Whenever the Map View is initially loaded, the resulting Zoom Factor value is a function of the Show All methodology, which is invoked automatically upon loading. The Show All methodology is designed to display a reasonable view of the overall network link-node diagram, and can be invoked manually by clicking on the Show All toolbar icon.

## Screen Origin

The Screen Origin represents the X and Y coordinates of the lower left-hand corner of the map, as viewed on the computer screen. Therefore, the Screen Origin value is updated whenever the map position is scrolled, panned, or zoomed.

## Bitmap Origin

The Bitmap Origin represents the X and Y coordinates of the lower left-hand corner of any bitmap background file that the user has loaded. The Bitmap Origin value defaults to 0,0 and is only updated if the user codes specific Bitmap Origin values on the Map Properties dialog. When performing one-touch CORSIM animation for the entire network, TRANSYT-7F will attempt to export bitmap background information to CORSIM, although the two programs use different input data to scale the bitmaps.

## Time Period

The "current" or "active" time period number is displayed here. If a multi-period analysis file is loaded, overall intersection average delay and level of service are displayed for the current time period number, displayed by the Time Period Quick Jump. If the current time period number is changed, the Map View will dynamically refresh any delay and level of service results shown. For a single-period file, it will always say "Time Period #1". If the "LOS" icon is deactivated, such that no delay or level of service results are superimposed on the map, the Time Period number is irrelevant and will be hidden.

## **Toolbar Icons**

### Exit

The first icon on the far left-hand side of the toolbar depicts a door through which the user can "Exit". Exiting in this manner will not save any node coordinates or map settings, but if these settings were saved earlier by using the "Save" icon, the changes will not be reversed. The Map View can also be exited by pressing the Escape key, or by pressing the F12 key, or by clicking on the upper right-hand "X" to close the map window, or by exiting the entire TRANSYT-7F GUI by pressing File > Exit or F12.

### Save

The Save icon is enabled as soon as the user changes any node position or map setting. The "dirty file flag" is also invoked whenever the user changes any node position or map setting, which means that the data file name at the top of the screen will have an asterisk to the right. Clicking on the Save icon writes the new node coordinates into the TIN file, stores new map settings on the computer, and removes the dirty file flag. After clicking on the Save icon, the original node coordinates could still be retrieved by closing the TIN file without saving, and reopening the TIN file. However, closing the TIN file (without saving) and reopening would not restore saved map settings such as "Show Grid", "Link Lengths", etc. Whenever the dirty file flag is on, the map can always be returned to its previous state by clicking on the "Undo" icon. The Save function can also be activated by pressing Ctrl-S.

### Properties

The Properties icon loads the Map Properties dialog screen. The Map Properties dialog screen is described in more detail later. The Map Properties dialog can also be activated by pressing the Z key.

### Progression Route

The following seven icons are considered "input coding" icons, because they assist with coding TRANSYT input data, and/or cosmetic data associated with the Map View. Only one input coding icon can be activated at a time.

The Progression Route icon, which is only enabled if a progression route is currently selected, loads the Progression Route dialog screen. The Progression Route dialog screen is described in more detail later. Progression routes are selected by drawing a rubber band around two or more nodes, or by using <Shift>-click to select multiple nodes. The Progression Route dialog can also be activated by pressing the X key, if a progression route is currently selected.

### Add Intersection

After clicking on the Add Intersection icon, the user can drop a new node anywhere on the map. After dropping the new node, the Add Intersection dialog (same as Edit > Optional > Intersection > Add) is displayed, which allows the user to code default settings for the new node. If the user wishes to add several new intersections in rapid succession, having the same default values, the "Don't show this again" checkbox can be used to skip the Add Intersection dialog. The "Don't show this again" checkbox will be restored the next time the GUI is launched.

### Two-Way Link

As stated in the Input Data Screen Overview section, link connections should not be drawn until after volumes have been coded on the Lanes and/or Traffic screens. Drawing those link connections on the Map after volumes have already been coded allows the program to perform "automatic flow balancing". The Map View issues a warning message if the user attempts to draw a new link before any volumes have been coded. After the Two-Way Link icon is selected, the user can click on the upstream node to be connected, and then click on the downstream node. When a new link is added in this manner, automatic flow balancing is performed for all time periods. If no link lengths have been coded yet (on the Lanes screen or the Traffic screen), link lengths are automatically coded the distance between node coordinates. New graphical link connections must be defined as northbound, southbound, eastbound, or westbound, so the "Orientation" combo box (from the Map Properties dialog) is used to control these directional assignments. For intersections with five or more approaches, link connections for the "extra" approaches should be coded manually on the Feeders screen.

### One-Way Link

The One-Way icon works just as the Two-Way Link icon does, but the second (downstream) node that is selected will not send traffic to the first (upstream) node that was selected.

## Link Curvature

After the Link Curvature icon is selected, link number labels are automatically placed in the middle of each link. Left-clicking on any link number label will then display two "control points" in the middle of that link. Dragging and dropping either one of these control points will then create link curvature. Doing this will automatically save the current map settings, although the original node coordinates could be retrieved by closing the TIN file without saving. Note that while the Link Curvature icon is selected, right-clicking on any link number label would automatically delete any previously coded curvature on that link.

When zooming and scrolling the map, link curvature is preserved. However, when dragging and dropping one or more nodes across the map, curvature control points are not automatically updated. Because of this, link curvature may need to be re-drawn after moving nodes across the map. It may be preferable to code link curvature after all nodes are in their appropriate positions.

Link curvature data is purely cosmetic, and does not affect TRANSYT model results. Instead, model results are sensitive to user-coded link lengths on the Lanes or Traffic screens. When performing one-touch CORSIM animation for the entire network, TRANSYT-7F will attempt to export link curvature information to CORSIM, although the two programs use different input data to specify curvature.

## Link Names

After the Link Names icon is selected, link number labels are automatically placed in the middle of each link. Left-clicking on any link number label will then display a text box for typing in the link name. The Escape key can be used to cancel the text box. Otherwise, pressing the Enter key (or clicking on the general map area) will save the link name. Doing this will automatically save all current map settings. While the Link Names icon is selected, right-clicking on any link number label would automatically delete any previously coded link name on that link. As discussed later, there is also a map adornment icon ("Show Link Names") that allows link names to be displayed on the map, or hidden, at any time. Link name data is purely cosmetic, and does not affect TRANSYT model results. When performing one-touch CORSIM animation, TRANSYT-7F will export link names to CORSIM.

## Delete Link or Node

After the Delete Link or Node icon is selected, the user can then click on a link or node to be deleted. Doing this will automatically save the current map settings, although the original node coordinates could be retrieved by closing the TIN file without saving. When the Delete Link or Node icon is selected, link number labels are automatically placed in the middle of each link. Clicking on any link number label will then delete the associated link by deleting all link connection information (record types 28 and 29) in the TIN file. This includes any link connection information for extra links (i.e., link numbers 13 and higher) or extra approaches (i.e., approach numbers 5 and higher). Clicking on any node will display the Delete Intersection dialog (same as Edit > Optional > Intersection > Delete), for a chance to confirm that all data for this intersection should be deleted. If the user wishes to delete several intersections in rapid succession, the "Don't show this again" checkbox can be used to skip the Delete Intersection dialog. The "Don't show this again" checkbox will be restored the next time the GUI is launched.

## Pointer

As discussed earlier, the Pointer icon allows the user to select individual nodes (or groups of nodes), drag-and-drop nodes, and right-click on nodes to edit their properties. When selecting nodes by simply dragging a rubber band around them, it is sometimes difficult to select (or exclude) all of the desired nodes. In this case, extra nodes may be selected and added to the group by using Ctrl-<Right Click>, Alt-<Right Click>, or Shift-<Right Click>. Of course, a simple <Right Click> by itself, instead of selecting that node, will prompt the user to edit the properties of that node. If only one node is selected at

any given time, after using the mouse rubber band or Ctrl-<Right Click>, Alt-<Right Click>, or Shift-<Right Click>, then the Map Properties dialog box can display the coordinates of that particular node.

### Pan

The following five icons are considered "view changing" icons, because they assist with altering the Map View without altering any input data. Only one view changing icon can be activated at a time.

The Pan icon can be used to scroll the map in the horizontal and vertical directions, simultaneously. After selecting the Pan icon, simply click down on the approximate map location that is desired to be moved (scrolled) to the center of the screen.

### Zoom In

After clicking on the Zoom In icon, the map can be magnified in one of two ways. First, by clicking and holding down the left mouse button, the user can drag a rubber band around a section of the map to be magnified. In fact, if the horizontal and vertical proportions of this rubber band area match the horizontal and vertical proportions of the screen, the selected area may become the exact viewing area on the entire screen. Manual (rubber band) zooming magnifies the screen by an amount consistent with the selected rubber band area. Secondly, analog zooming can be achieved by simply left-clicking once inside the general map area. Analog zooming magnifies the screen by 50%. Pressing the plus sign (+) key on the keyboard also activates analog zooming.

As stated earlier, when English units are in effect, the maximum Zoom Factor is 8.00. When metric units are in effect, the maximum Zoom Factor is 25.00. If the user zooms in to the maximum Zoom Factor, the Zoom In icon may become temporarily disabled. In addition, if the user zooms in to a location that would cause an out-of-range Screen Origin or Bitmap Origin, a warning message may be issued that says "You have reached the map boundaries".

### Zoom Out

After clicking on the Zoom Out icon, the map view can be compressed (analog zooming) by simply left-clicking once inside the general map area. This causes the Zoom Factor to be divided by 150%, and the location where the user clicked becomes the center of the new viewing area. Pressing the minus sign (-) key on the keyboard also activates analog zooming.

As stated earlier, when English units are in effect, the minimum Zoom Factor is 0.25. When metric units are in effect, the minimum Zoom Factor is 0.75. If the user zooms out to the minimum Zoom Factor, the Zoom Out icon may become temporarily disabled. In addition, if the user zooms out to a location that would cause an out-of-range Screen Origin or Bitmap Origin, a warning message may be issued that says "You have reached the map boundaries".

### Bitmap Scaling

The Bitmap Scaling icon, which is only enabled when a bitmap background icon is loaded, loads the Bitmap Scaling dialog screen. Bitmap backgrounds can be loaded by opening a bitmap file from the Map Properties dialog, and then pressing the Bitmap Background icon. The Bitmap Scaling dialog screen is described in more detail later.

### Show All

Anytime the Map View is initially loaded, the "Show All methodology" is automatically applied. Often times after zooming or scrolling the map for awhile, the user will want to return to that initial view where the link-node diagram was centered within the screen. Clicking on the Show All icon accomplishes this. The Show All methodology is also automatically applied by the Undo icon, after it has reversed any recent changes to the map.

If possible, the Show All methodology will place the link-node diagram at the center of the screen, at a Zoom Factor of 4.00 (if English units) or 12.50 (if metric units). However, if the link-node diagram is too large to fit within the screen, the Zoom Factor will be reduced to the largest possible value that makes the entire link-node diagram visible.

### Node Numbers

Beginning with the Node Numbers icon, this group of seven icons is considered a group of "map adornment" icons. These icons are either pressed (activated), or unpressed (deactivated). When a map adornment icon is pressed, the associated information will be superimposed on the map. Only unpressing the icon, or closing the map without saving, will remove the associated information from the map.

By default, the Node Numbers icon is pressed so that node numbers can be displayed on top of each node. The user can unpress this icon to hide the node numbers, if desired.

### Link Lengths

If the Link Lengths icon is pressed, link lengths will be superimposed on the middle of each link. The link lengths shown here are taken directly from the TIN file (record type 28), and can be viewed on the Feeders screen. Ideally the user would place nodes at a distance such that the distances (in twips) would be equal to the link lengths (in feet or meters). If not, this will still not affect the displayed link lengths, and will not affect any model results. The user can unpress this icon to hide the link lengths, if desired. Link lengths can never be displayed while the Delete Link or Node icon is activated, or while the Link Curvature icon is activated.

### Free Flow Speed

If the Free Flow Speed icon is pressed, free-flow speeds will be superimposed on the middle of each link. The free-flow speeds shown here are taken directly from the TIN file (record type 28), and can be viewed on the Feeders screen. The user can unpress this icon to hide the free-flow speeds, if desired. Since link lengths are also displayed in the middle of a link, free-flow speeds and link lengths cannot be viewed simultaneously. The same is true for link names. Free-flow speeds can never be displayed while the Delete Link or Node icon is activated, or while the Link Curvature icon is activated.

### Show Link Names

The Show Link Names icon is only enabled if link names were coded through the Link Names input coding icon, as discussed earlier. If the Show Link Names icon is pressed, link names will be superimposed on the middle of each link. The link names shown here are taken directly from the TIN file, and can be viewed in the record type editor. The user can unpress this icon to hide the link names, if desired. Since link lengths are also displayed in the middle of a link, link names and link lengths cannot be viewed simultaneously. The same is true for free-flow speeds. Link names can never be displayed while the Delete Link or Node icon is activated, or while the Link Curvature icon is activated.

### Show Grid

If the Show Grid icon is pressed, horizontal and vertical gridlines will be superimposed across the map. This can be helpful for understanding the amount of real-world distance represented by the map, but on some computers, the gridlines may slow down any dynamic scrolling (when dragging the scrollbar buttons). Dynamic scrolling can be deactivated in the Map Properties dialog if desired. The normal grid spacing is 1000 twips. For Zoom Factors lower than 0.50, the grid spacing is 2000 twips. Note that this 2000 twip spacing would actually occupy a space of 1000 twips, on screen, if the Zoom Factor were slightly below 0.50. For Zoom Factors higher than 4.00, the grid spacing is 100 twips. This 100 twip spacing would actually occupy a space of 400 twips, on screen, if the Zoom Factor were slightly above

4.00. The user can unpress this icon to hide the gridlines, if desired.

### Bitmap Background

The Bitmap Background icon is only enabled if the user loads a bitmap background file through the Map Properties dialog screen. To demonstrate this feature, a sample bitmap background file called "Sample.bmp" is installed along with the software, and can be loaded from the Example folder. If the Bitmap Background icon is pressed, the bitmap background will be displayed on the map. Bitmap backgrounds can be helpful for understanding the real-world signal system represented by the link-node diagram, and can be helpful when dragging and dropping nodes to their appropriate locations. On some computers, the bitmap background may slow down any dynamic scrolling (when dragging the scrollbar buttons). Dynamic scrolling can be deactivated in the Map Properties dialog if desired. The Bitmap Origin and Bitmap Scaling Factor can also be specified in the Map Properties dialog. To scale the bitmap background file for consistency with the link-node diagram, the Bitmap Scaling icon can be selected. When performing one-touch CORSIM animation for the entire network, TRANSYT-7F will attempt to export bitmap background information to CORSIM, although the two programs use different input data to scale the bitmaps.

### Level of Service

The Level of Service (LOS) icon is only enabled if an output file is present for TRANSYT, CORSIM, or HCS-Signals. For an output file to be present, it must be located in the same folder as the TIN file, which happens automatically following any TRANSYT simulation run or CORSIM processing run. For HCS, input and output are actually contained in the same file. TRANSYT output files, which contain LOS results for each intersection, are named <filename>.TOF. CORSIM processing output files, which also contain LOS results for each intersection, are named <filename> Summary.TXT. HCS-Signals input/output files, which contain LOS results for one intersection, must be named <inname>.XHS. In this case, <inname> must match an intersection name coded within the TIN file, in order to be recognized. When HCS files are imported into TRANSYT-7F, or when "Link to HCS" is used, the HCS filename is automatically coded into the TIN file as an intersection name.

If the LOS icon is pressed, overall intersection average delay and LOS can be superimposed above each node. The background color of this label is one of the 9 available map background colors. If a TOF file is present, the "T7F" icon is enabled for displaying the TRANSYT LOS. If a CORSIM processing output file is present, the "CSM" icon is enabled for displaying the CORSIM LOS. If an HCS file is present, the "HCS" icon is enabled for displaying the HCS LOS. If output files are present for multiple programs, the user can click back and forth between the T7F, CSM, and HCS icons, to see a direct graphical comparison between model results. Certain issues regarding CORSIM LOS are discussed in the "CORSIM Processing Methodology" section. The user can unpress the LOS icon to hide the intersection delay and LOS, if desired.

If a multi-period TOF file is present, overall intersection average delay and level of service are displayed for the "current" or "active" time period number, displayed by the Time Period Quick Jump. If the current time period number is changed, the Map View will dynamically refresh any delay and level of service results shown. If a multi-period HCS file is loaded, the Map View will only display results for the first time period. If a multi-period CORSIM is loaded, the Map View will display cumulative results that represent all time periods.

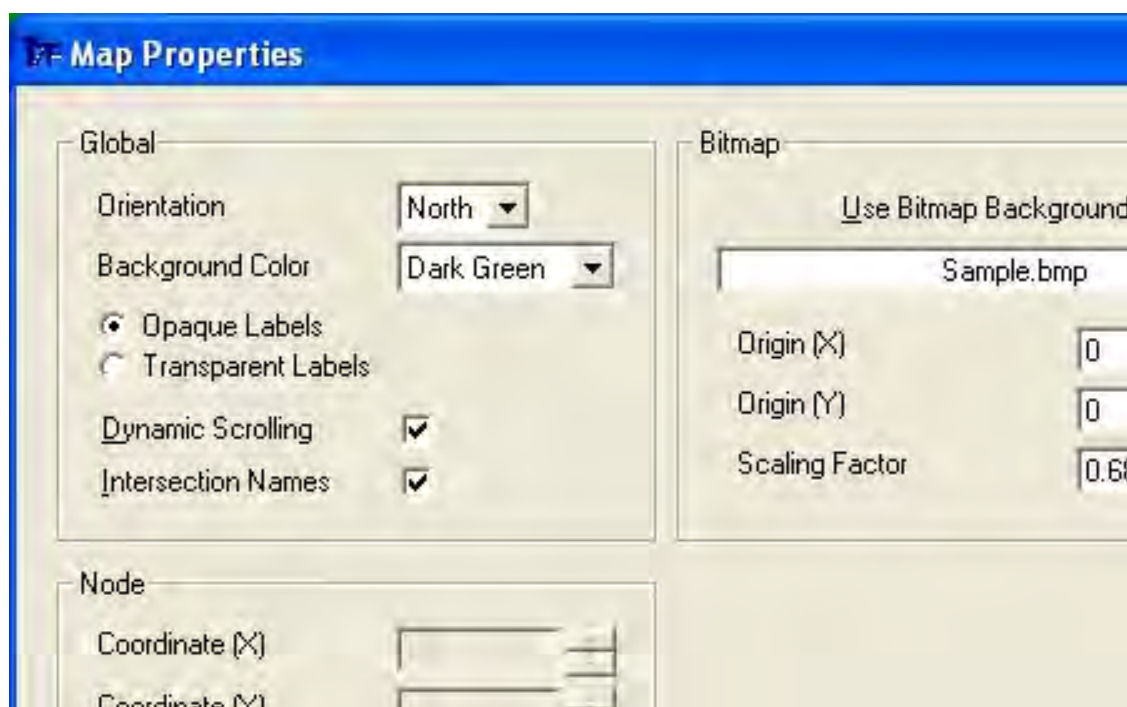
### Undo

If the Undo icon is clicked, the Map View is returned to its most recently saved state, and the Show All methodology is again executed. If the filename at the top of the screen has no asterisk to the right, this means that the dirty file flag is currently off, and there are no node coordinates or map settings available to be reversed. If the dirty file flag is on, clicking on the Undo icon returns the map to the state it was in the last time the dirty file flag was off. The Undo function can also be activated by pressing Ctrl-Z.

If a node is dragged and dropped, its node coordinates are changed, but the Undo icon would reinstate the original node coordinates. If any of the seven map adornment icons (node numbers, link lengths, free-flow speed, show link names, show grid, bitmap background, level of service) are changed between pressed and unpressed, the Undo icon would reinstate the original settings. Zooming or scrolling the map changes the Screen Origin, but does not change any node coordinates or map settings. Other input data coding (i.e., adding and deleting of progression routes, links, and nodes) automatically causes all node coordinates and map settings to be saved. The original node coordinates and input data could be retrieved by closing the TIN file without saving, and reopening the TIN file. However, closing the TIN file (without saving) and reopening would not restore saved map adornment settings such as Show Grid, Link Lengths, etc.

## Map Properties Dialog

The Map Properties dialog screen is shown below, with sample data in the input fields. The Map Properties dialog is loaded by clicking on the Properties icon, which is the third icon from the left on the toolbar, or by pressing the Z key. Values and settings within this screen are stored into memory on the computer every time the "Apply" or the "OK" button is pressed. Except for the node coordinates and the global map orientation, values and settings within this screen are not saved inside the TRANSYT-7F input data file. Therefore, if different data files (representing different networks or different cities) are loaded into the software, the same map properties (bitmap background, map orientation, etc.) will be applied to all data files unless the user makes a change.



## Orientation

By default, the Map View assumes a North orientation, where northbound points toward the top of the screen. If the orientation were changed to East, northbound would point towards the left side of the screen. Link connections between nodes are drawn based on connecting link data from the Feeders screen. When new links are drawn, the Feeders screen is called, and automatic flow balancing is performed. New graphical link connections must be defined as northbound, southbound, eastbound, or westbound, so the Orientation combo box is used to control these directional assignments. For



intersections with five or more approaches, link connections for the "extra" approaches should be coded manually on the Feeders screen. Apart from when links are being added and deleted in the Map View, the Orientation combo box has no other effect on the model, or other parts of the software.

### Background Color

There are nine different background colors to choose from. If the user loads a bitmap background file, this background color will not be visible, except as a background to the link length and delay/LOS labels.

### Opaque Labels/Transparent Labels

Labels for map adornments (e.g., delay/LOS, link lengths, etc.) are opaque by default. Opaque labels ensure that information on the labels will be visible above any link, gridline, or bitmap background. If transparent labels are selected, delay/LOS and link lengths can still be superimposed on the map, but they may be obscured by the other elements (link, gridline, or bitmap background).

### Dynamic Scrolling

Dynamic scrolling allows the entire network to be continuously visible as it is dragged in any one of four directions (up, down, left, and right). In order to dynamically scroll the map, the horizontal and vertical scrollbar buttons must be dragged and dropped. Dynamic scrolling simply changes the X and Y coordinates of the Screen Origin, without changing the location of any links and nodes, and without changing the origin of the bitmap background (if one has been loaded). Because of this, dynamic scrolling makes it appear that the observer is traveling across the map, as if they were flying above in an airplane.

Dynamic scrolling can sometimes occur very slowly on a relatively slow computer, or with a relatively large network, or a relatively large bitmap background. Because of this, the Dynamic Scrolling checkbox can be deactivated on the Map Properties dialog. Deactivating dynamic scrolling will result in "static scrolling", in which the map position is not altered until the user releases the scrollbar button.

### Intersection Names

If intersection names have been coded on the Edit > Optional > Intersection > Names screen, they can be displayed when moving the mouse cursor over a node. If the Intersection Names checkbox is deactivated on the Map Properties dialog, intersection names will not be displayed when moving the mouse cursor over a node, even if intersection names have been coded on the Edit > Optional > Intersection > Names screen.

### Node Coordinate (X and Y)

The Node Coordinate fields are only enabled if one or more nodes on the map are currently selected. If only one node is selected, its current node coordinates will be displayed in these two enabled fields. If multiple nodes are selected, these two fields will be enabled, but blank. If the user codes new node coordinates here, and then clicks on "OK" or "Apply", all selected nodes will be moved to the specified location. It is also possible to enter an X coordinate while leaving the Y coordinate blank, or vice-versa. This would allow a certain set of nodes to be lined up either vertically or horizontally. If desired, node coordinates can also be coded onto record type 70 by using the record type editor.

### Node Radius

When English units are in effect, the default Node Radius is 30 twips. When metric units are in effect, the default Node Radius is 9 twips. These default settings are intended to provide reasonable approximations of intersection width. Assuming a scaling of 1 twip per foot, or 1 twip per meter, this would result in 60-foot or 18-meter intersection widths. If the user changes the Node Radius, circular

nodes on the map view can be drawn either larger or smaller than this. As with the node coordinates and other map settings, the Node Radius has no effect on TRANSYT model results.

### Use Bitmap Background

This checkbox indicates whether or not a bitmap background file will be used. When this checkbox is turned on, the user will be able to browse the computer to select any available bitmap background (\*.BMP, \*.GIF, \*.JPG, \*.JPEG) file, and the bitmap background toolbar icon will be enabled. To demonstrate this feature, a sample bitmap background file called "Sample.bmp" is installed along with the software, and can be loaded from the Example folder.

### Bitmap Origin (X and Y)

By default, the lower left hand corner of the bitmap background is located at 0,0 twips. Redefining the Bitmap Origin allows the bitmap background to be moved without changing any of the node coordinates. Relative to the 0,0 starting point, coding a bitmap background origin such as 750,750 would cause the lower left-hand corner of the bitmap background to move closer to the center of the screen.

### Bitmap Scaling Factor

The default Scaling Factor is 1.00, which implies no extra compression or magnification. A Scaling Factor of 1.00 combined with a Zoom Factor of 1.00 eliminates any compression or magnification. If the Zoom In or Zoom Out toolbar icons are used, this magnifies the link-node diagram and bitmap background by changing the Zoom Factor, but doesn't change the Scaling Factor. When the bitmap background is scaled by using the Bitmap Scaling dialog, the Scaling Factor is automatically modified by the software. The Bitmap Scaling dialog is described later.

## **Progression Route Dialog**

The Progression Route dialog screen is shown below, with sample data in the input fields. The Progression Route dialog is loaded by clicking on the Progression Route icon, which is the fourth icon from the left on the toolbar. The Progression Route icon is only enabled if a progression route is currently selected. The Progression Route dialog can also be activated by pressing the X key, if a progression route is currently selected.

Progression routes are selected by drawing a rubber band around two or more nodes, or by using <Shift>-Click to select multiple nodes. When selecting nodes by simply dragging a rubber band around them, it is sometimes difficult to select (or exclude) all of the desired nodes. In this case, extra nodes may be selected and added to the group by using <Ctrl>-Click, <Alt>-Click, or <Shift>-Click.

Values and settings within this screen are stored into the TRANSYT-7F input data file when the "OK" button is pressed. For new routes, the link list (associated with the intersections selected on the Map View) will be automatically written onto record type 42. This link list will apply the through movement link numbers in accordance with the user's previously chosen link numbering scheme. If the "Delete" button is pressed, the currently selected progression route will be deleted from the TIN file.

This graphical coding technique is limited to a maximum of 14 intersections. If the user wishes to code an arterial route containing more than 14 intersections, it will be necessary to use the Edit > Optional > Route screen or the record type editor to code the additional links.

Refer to the Progression Opportunities (PROS) section of the documentation for figures and examples of proper and improper arteries for PROS.

#### Route Title

Route Titles are displayed in the legacy report (\*.TOF), and on the Time-Space Diagram screen, for each progression route. If a Route Title already exists within the TIN file, it is displayed here, and the user can modify it here if desired. Route Titles cannot exceed the typical record type width of 80 characters. A default Route Title such as "Route #1" or "Route #2" is displayed if a new group of nodes has been selected.

#### Route Number

Because Route Numbers must begin at '1' and not skip any numbers, the appropriate Route Number is automatically displayed in this field, and cannot be modified by the user. For example, if there are 4 progression routes in the network, they will be numbered 1, 2, 3, and 4. Unusual Route Numbers or Route Names can be documented by using the Route Title. If a progression route is deleted via the "Delete" button, Route Numbers might be automatically recoded within the TIN file. Inside the TIN file, Route Numbers are listed on record types 42, 43, and 61. The Master Intersection, which is listed on the Edit > Optional > Global > Network Parameters screen (or record type 10), can only be coded on Route Number 1.

#### Number of Nodes

This field automatically displays the currently selected number of nodes.

#### Forward Direction

The Forward Direction is the critical direction for progression in the peak hour. This combo box contains the four possible Forward Directions Eastbound, Westbound, Northbound, and Southbound. The Forward Direction should be consistent with the Orientation from the Map Properties dialog screen. If the Forward Direction is not consistent with the Orientation when trying to code a progression route, an error message will be issued. Inside the TIN file, the Forward Direction code is listed on field 16 of record type 60.

By default, the Orientation is northbound, such that the northbound direction travels straight up the computer screen, and the eastbound direction moves from left to right across the computer screen. Assuming a northbound Orientation, if intersection numbers are listed 1, 2, 3 ... 10 from left to right across the screen, selecting an eastbound Forward Direction will cause link numbers for nodes 1, 2, 3, etc. to be listed first in the progression route. Selecting westbound would cause the link numbers for nodes 10, 9, 8, etc. to be listed first in the progression route. However, selecting a northbound or southbound Forward

Direction under these conditions could cause the Orientation error message listed above.

### Two-Way Route/One-Way Route

If "Two-Way Route" is selected, the progression route is defined with both a forward direction and a reverse direction. "One-Way Route" could be selected for one-way streets, or in situations where no progression whatsoever was desired for the reverse direction.

### Use Directional Weighting

When using one of the PROS objective functions, individual routes can be given preference, and/or directions on the routes can be given preference, during the optimization process. Clearly, a 90:10 directional weighting would focus on maximizing progression in the forward direction, possibly at the expense of the reverse direction. Note that accepting a 50:50 directional weighting may have a substantially different effect on the optimization process, when compared to not using any directional weighting. Refer to the PROS section of the documentation for more details.

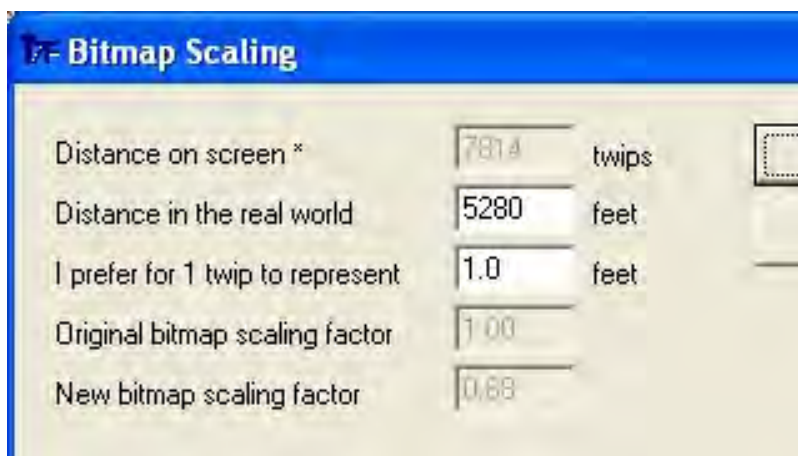
Inside the TIN file, Forward and Reverse Weightings for any given progression route are listed on record type 43, but only fields 1 through 4 are recognized by the Progression Route dialog. If directional weighting is not used, record type 43 will be omitted for the given route.

### Forward and Reverse Weighting

The weighting factors are proportional. For example, if '66' is coded for one route and '33' for another, the former will have twice the weight applied to PROS than the latter. Although this is not a specifically required approach, the weighting factors can be considered as percentages. For example, to achieve 60% of progression in the forward direction, code '60' and '40' as the forward and reverse factors, respectively. If a 50:50 split on a second route is desired, but double the priority as the first route, code '100' in both fields as both the forward and reverse factor.

### **Bitmap Scaling Dialog**

The Bitmap Scaling dialog screen is shown below, with sample data in the input fields. The purpose of this dialog is to assist the user in determining an appropriate Bitmap Scaling Factor. The Bitmap Scaling Factor can be coded manually into the Map Properties dialog, but it might be easier to determine an appropriate value by using the Bitmap Scaling dialog.



Field	Value	Unit
Distance on screen *	7874	twips
Distance in the real world	5280	feet
I prefer for 1 twip to represent	1.0	feet
Original bitmap scaling factor	1.00	
New bitmap scaling factor	0.68	

The Bitmap Scaling dialog is loaded by clicking on the Bitmap Scaling icon, which sits between

the Zoom Out and Show All toolbar icons. After clicking on the Bitmap Scaling icon, the mouse cursor changes its shape, to become a distance measuring tool. To cancel this operation, simply right-click the mouse cursor on the general map area. Otherwise, the user must now left-click on a pair of points along the bitmap background, between which the real world distance is known. After left-clicking on the second map location, the Bitmap Scaling dialog is displayed. Values and settings within this screen are stored into memory on the computer every time the "OK" button is pressed.

#### Distance on screen (twips)

This field displays the adjusted on-screen distance, in units of twips, between the pair of points. This on-screen distance is adjusted by the Zoom Factor, for consistency with the node coordinates. For example, an on-screen distance of 1000 at a Zoom Factor of 4.00 will actually appear to cover 4000 twips. This distance cannot be modified by the user, except by canceling this operation, and left-clicking on a new pair of points.

#### Distance in the real world (feet or meters)

The real-world distance must be coded here by the user. The default real-world distance displayed in this field is not likely to match the on-screen distance unless the bitmap background has already been scaled previously. Normally the user would select a pair of points along the bitmap background whose real-world distance is obvious, such as two major arterials that are separated by a mile.

#### I prefer for 1 twip to represent (feet or meters)

As stated earlier, the Map View coordinate system is specified in units of twips, which are a unit of measurement on the computer screen. If English units are in effect, a ratio of one foot per twip is recommended, simply for ease of understanding. If metric units are in effect, a ratio of one meter per twip is recommended. Thus the default value in this field is 1.0. However, the user could apply any desired scaling, such as 10 feet per twip, 0.5 meters per twip, and so on.

#### Original bitmap scaling factor

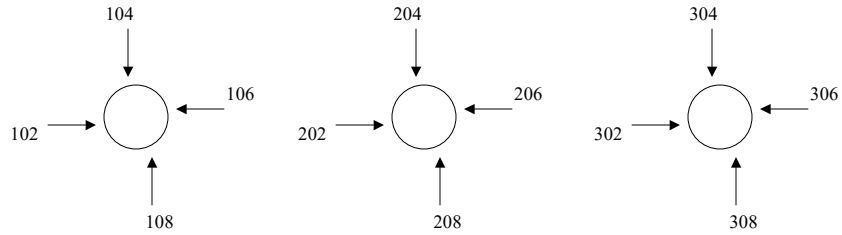
The original bitmap scaling factor represents the magnification or compression factor that is currently being applied to the bitmap background image.

#### New bitmap scaling factor

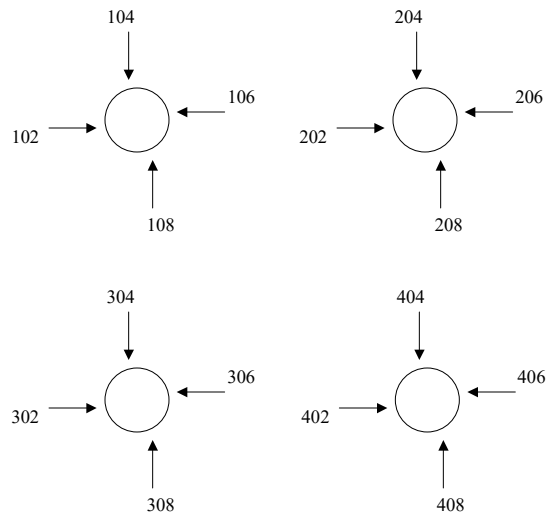
The new bitmap scaling factor represents the magnification or compression factor that will now be applied to the bitmap background. This new scaling factor is computed as a function of the four preceding parameter values.

## **Link Numbering Scheme**

The network of streets and intersections is represented in TRANSYT by a node/link identification scheme. Most of the TRANSYT-7F input data screens are designed to display the pertinent link and node numbers at all times. The node/link identification scheme is designed by the user, and once finalized, all other data are referenced to it. A node is an intersection and a link is a unidirectional traffic movement, or movements, between two nodes. In many situations, more than one link may connect two nodes. The figures below show a few of the typical link/node schemes.

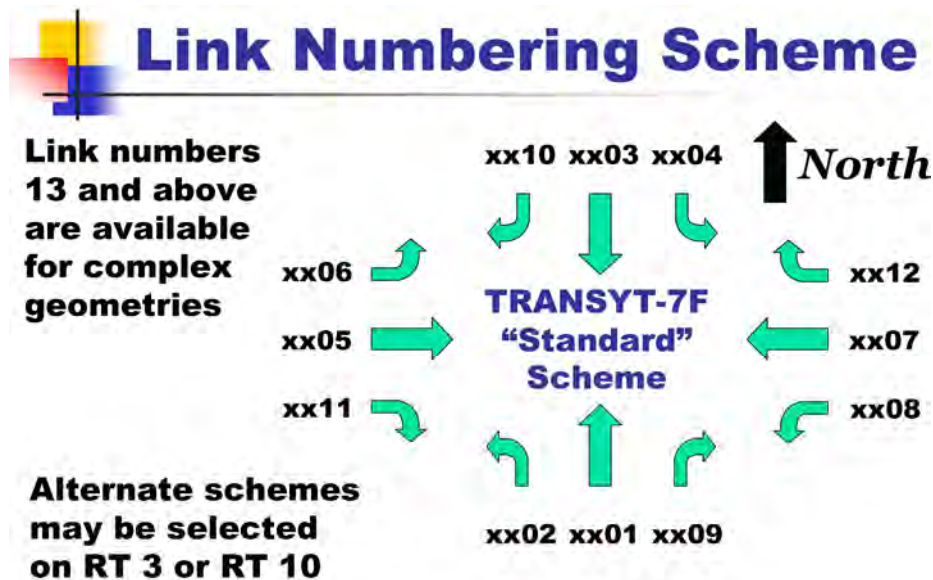


A Linear Arterial Street



A Two-Dimensional Network

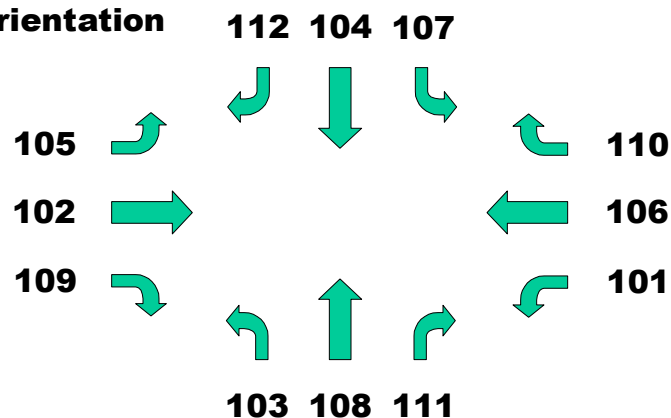
The link/node numbering is entirely arbitrary in TRANSYT-7F; however, the user is strongly urged to adopt one scheme (at least in any given data set) and stick with it. The schemes commonly recommended for use in TRANSYT-7F are illustrated in the figures below, where "xx" indicates a node number between 1 and 99.



In the NEMA scheme illustrated below, node number 1 is assumed, and right-turn links are numbered beyond the standard NEMA phases 1-8.

## TRANSYT-7F "NEMA" Scheme

**Note: these numbers vary with orientation**



These pre-defined schemes may be chosen on the Edit > Optional > Global > Network Parameters screen (or record type 10). If a pre-defined scheme has not been chosen on the Global > Network Parameters screen, the program checks to see whether a unique link numbering scheme has

been coded by the user on the Global > Link Numbering Scheme screen (or record type 3). If no link numbering scheme has been chosen in either location, the program assumes that the TRANSYT-7F Standard scheme is being used.

Most of the TRANSYT-7F software is designed to implement a link numbering scheme having numbers 1 through 12. Under this scheme, link numbers 13 and higher would be available for extra lanes, movements, and approaches, beyond the typical 12 turning movements (L, T, and R for NB, SB, EB, and WB). Also, the TRANSYT-7F Standard and NEMA schemes described earlier both use numbers 1 through 12 as part of their scheme. Definition of numbers 13 and higher within the link numbering scheme is not recommended.

A TRANSYT-7F link can consist of one or more traffic movements, and may cover one or more lanes. Separate links are required in TRANSYT for any movements that:

- have separate lanes (or bays) exclusively for their use,
- have significantly different lane utilization or queuing than the adjacent lanes,
- move in a unique set of signal phases, or
- are turns and throughs using a shared lane and the turn is opposed by a through movement.

Link assignments can be specified by the user on the Edit > Lanes screen, or on the Edit > Traffic screen. Typically the Lanes screen will assign a specific link number for each turning movement. However, there are special cases in which the user might want to make additional adjustments.

For example, it is allowable for shared throughs and right-turns to be modeled using a single link, as long as they always move during the same phase(s). Their combined volumes and saturation flow rates could be coded on the Traffic screen. As demonstrated later, multiple downstream destinations can still be accounted for on the Edit > Feeders screen. Another example is a lane-by-lane analysis. If there are multiple lanes for the same turning movement, such as dual exclusive left-turn lanes, each lane can be modeled separately by defining a unique link number for each lane. This type of lane-by-lane coding and link number assignment can be handled automatically by selecting "Lane-by-Lane" on the Lanes screen.

## **Lane Configuration Screen (associated with record types 28, 29, and 291)**

This input screen establishes lane channelizations and usage for the traffic network. It can also be used to establish saturation flow rates, link lengths, traffic volumes, and turn pocket lengths. For most traffic network analyses, this type of data will often be coded first by the user, because other aspects of the analysis depend on them heavily. Data from this screen can also be coded on the Traffic and Queuing screens, so it is even possible to skip this screen entirely. However, the Lane Configuration screen is (in some ways) more efficient and intuitive, and will serve as an effective "pre-processor" for the data on those other two screens. This screen can be sent to the printer, or to an image file, by pressing Ctrl-P. The Edit > Lanes > Intersection View option is illustrated below.



**7F Lane Configuration at University and 13th**

103

104

Lane #   
☒ Lane Group  
☐ Lane-by-Lane

Distance (ft)

	Left	Thru	Right
Volume (vph)	45	360	45
Peak Hour Factor	0.90	0.90	0.90
Turn Bay Storage (veh)	12		

106

105

Lane #   
☒ Lane Group  
☐ Lane-by-Lane

Distance (ft)

	Left	Thru	Right
Volume (vph)	45	360	45
Peak Hour Factor	0.90	0.90	0.90
Turn Bay Storage (veh)	12		

Lane #   
☒ Lane Group  
☐ Lane-by-Lane

Distance (ft)

	Left	Thru
Volume (vph)	45	360
Peak Hour Factor	0.90	0.90
Turn Bay Storage (veh)	12	

Node

Lane #   
☒ Lane Group  
☐ Lane-by-Lane

Distance (ft)

	Left	Thru	Right
Volume (vph)	45	360	45

Add...

The Edit > Lanes > Intersection View option geometrically displays four approaches on one large screen. In this case, it is best to set the computer at a high screen resolution, so that all information can be simultaneously visible. It is also possible to display a separate screen that focuses on one approach by selecting Edit > Lanes > Northbound, Southbound, Eastbound, or Westbound. For intersections with five or more approaches, lane configuration data for the "extra" approaches must be coded on the Edit > Traffic screen.

#### Lane #

This input field indicates a lane number for which the user could change lane use, or switch between "lane group" and "lane-by-lane" analysis. Lane number 1 is always the leftmost lane on the approach. This field will thus affect the appearance and behavior of the adjacent Lane Use combo box,

as well as the Lane Group radio buttons below. Clicking on any lane in the Lane Use Diagram will automatically update the lane number in this field. It is allowable to enter a lane number value that equals the current number of approach lanes plus one, in the event that a new lane must be added.

### Lane Use

This combo box provides a list of lane channelization options for the currently selected lane number. These options include any combination of the letters L (left), T (thru), and R (right), which represent the three available turning movements. If there are more than three turning movements on the approach, such as there might be for a 5-leg or 6-leg intersection, these "diagonal" or "extra" movements can be coded on the Traffic screen. The Lane Use combo box dynamically re-populates itself to prevent the coding of illogical turning movements, such as a right-turn lane to the left of a thru lane. Whenever the lane use is changed, the Lane Use Diagram is updated automatically.

If the user selects a lane number value that equals the current number of approach lanes plus one, the Lane Use will become blank. If the user then codes a new lane use code, the impact of this depends on whether these turning movements already exist on the approach. If a brand-new turning movement is added, one or more new links will be added to the dataset, with a default volume of 100 vehicles per hour (vph). If the turning movement already exists on the approach, but an extra lane for that turning movement has been added, the volume will remain the same, but the saturation flow rate for that link will be factored up accordingly.

If the user selects an existing lane number value, but then changes the Lane Use to become blank, the impact of this depends on whether any more of these turning movements still exist on the approach. If an existing turning movement is completely deleted, its link will be deleted to the dataset. If the turning movement still exists on the approach, but in a different lane, the volume will remain the same, but the saturation flow rate for that link will be factored down accordingly. It is perfectly acceptable to delete all lanes on any given approach, for example, to create a T-intersection. However, to delete an entire intersection, this should be done either on the Map View, or by selecting Edit > Optional > Intersection > Delete.

### Lane Group and Lane-by-Lane

These radio buttons allow the user to control whether the software will perform HCM-style "lane group" analysis, or a more precise "lane-by-lane" analysis. This decision is only applicable when a given turning movement is allowed in more than one lane of the approach. In the situation where a turning movement is confined to its own single, exclusive lane, a lane group analysis would end up being a lane-by-lane analysis anyway. In these situations, the two radio buttons are automatically disabled. The lane-by-lane selection is also automatically disabled for the innermost turn lane, if there are multiple exclusive turn lanes.

When lane utilization is fairly even within each lane of the shared lane group, the entire lane group may be defined and modeled fairly accurately as a single link. Highly unequal lane utilizations are modeled more accurately when a link is defined to represent each individual lane. Thus, for a multilane lane group in which the same turning movement is allowed from multiple lanes, these radio buttons are enabled so the user can choose their desired style of analysis.

In general, the software runs more efficiently when a single link number can be used to represent multiple lanes. This allows for lane group modeling, which is consistent with Highway Capacity Manual [12] procedures, and many other software packages. Attempting to model every lane throughout the network as a lane-by-lane analysis is certainly discouraged. However, for special network locations in which traffic utilization of a given lane is significantly different from other lanes in the same lane group, lane-by-lane modeling becomes necessary for achieving realistic results. Note that secondary shared turning movements are considered to be part of the through-movement lane group, whereas exclusive turn lanes are considered to be a separate lane group in and of themselves.

If the "lane-by-lane" selection is made, this impacts the model on several levels. First, the Lane Use Diagram toward the bottom of the screen is updated to reflect a brand-new link number within the dataset. Since movement numbers 1 through 12 are reserved for standard turning movements within the link numbering scheme, the new link number will be set to (node number)\*100 + (movement number 13 or higher). Secondly, the original traffic volume and saturation flow rate will be automatically subdivided between the new and old links. Finally, the user will need to ensure that the new link is eventually accommodated in the timing plan and feeding link information. Since the node summary reports only display results for the 12 standard turning movements, it will be necessary to refer to the legacy output report to view results for movement numbers 13 and higher.

To accommodate "extra" links 13 and higher in the timing plan, it is sometimes necessary to use the Edit > Optional > Timing screen, which is more flexible than the graphical Timing View screen. On the other hand, if the timing plan is coded after a lane-by-lane analysis has been requested for that node on the Lane Configuration screen, the Timing View screen can automatically include those extra links within the timing plan. In order to perform phasing sequence optimization with "extra" links, it is necessary to code those links as "associated links" (Edit > Optional > Global > Associated Links, or record type 5).

For feeding link information, extra links 13 and higher can be coded on the Edit > Feeders screen. Since there is a limit of four feeding links per downstream link, it is sometimes necessary to omit one or more feeding links from the list. In this case, the user should keep in mind that feeding volumes are factored up proportionally to their feeding volumes until they match the downstream volume, for the purpose of establishing the flow profile. If four feeding links are already coded with the appropriate feeding flows, it should be possible to omit additional feeding links without significantly disrupting the flow profile. If more than four feeding links exist, it is OK to code only one of multiple thru links as feeding the entire approach volume (assuming no queue spillback) because this would result in the same flow profile. If queue spillback exists on these links, it is probably best to omit the feeding link(s) with the lowest volume-to-saturation flow rate (v/s) ratio.

Regarding standard turning movement numbers 1 through 12, the numbers reserved for left or right turning movements are always assumed (by the Lane Configuration screen) to belong to any shared lane. This means that if an exclusive right-turn lane exists next to a shared through and right-turn lane, the exclusive turn lane will automatically be assigned an extra link number (13 and higher). In addition, extra link numbers are not allowed to cover more than one lane, so any new extra lanes added will result in their own brand-new link number.

If a lane-by-lane analysis is currently selected, but the user selects "Lane Group" to revert back to a lane group analysis, the Lane Configuration screen will attempt to restore the original lane group volume and saturation flow rate. However, round-off error will sometimes occur.

For a tutorial example regarding lane-by-lane analysis, refer to the documentation section entitled Lane-by-Lane Analysis Example.

#### Distance (ft or m)

This input field specifies the distance from the upstream node to the subject node, for the given approach. Usually, the approach distance is assumed to be equal for all turning movements (left, thru, and right). If one of the turning movements has a significantly different link length, this distance can be coded on the Edit > Traffic screen. However, turn pockets are considered to have the same overall link length as the adjacent thru movement. For a turn bay or turn pocket, the full signal-to-signal distance is also specified as the link length, in order to achieve proper simulation of platoon dispersion. Turn pocket lengths should only be indicated by using the Turn Bay Storage field. If turning movements on the same approach have different approach distances, which should be a rare condition, then the displayed approach distance will be the thru movement's link length. If there is no thru movement, then the displayed approach distance is the right-turn link length (if right-hand drive), or the left-turn link length (if left-hand drive).

When the approach links to be coded are external links, it is not necessary to code a Distance. If no Distance is coded, no total travel will be calculated for the links, and the links will not be considered in the calculation of network-wide average speed. Total travel time on the links will be equal to the total delay estimated by TRANSYT. Fuel consumption will be estimated for the links based on the default network-wide link speed, which is the value coded on the Edit > Optional > Global > Network Parameters screen (or record type 10). If queuing on the external links is important to the analysis, Distance should be specified.

When a Distance is provided for external links, total travel will be estimated, and the links will be considered in the calculation of network-wide average speed. Total travel time on the links will include both the delay and the time spent at cruise speed. The fuel consumption estimate will be based on the speed provided for feeding link #1 on the Edit > Feeders screen (or record type 28 field 9), or the network-wide value (Edit > Optional > Global > Network Parameters screen, or record type 10).

### Volume (vph or vpp)

This input field specifies the traffic volume demand for the three standard turning movements. The Edit > Traffic screen must be used to code diagonal turn volumes, for intersections with five or more approaches. Any volumes coded here for Left, Thru, and Right, will be saved and automatically loaded by the Traffic screen. Therefore it is not necessary to code volumes twice, in both the Lanes and Traffic screens.

Traffic volume is typically specified in units of vehicles per hour, regardless of the duration of simulation. This volume is specified as the total number of vehicles, including heavy vehicles. Later on, the saturation flow rate (and queuing headways) can be used to reflect the percentage of heavy vehicles in the analysis. Traffic volume is specified as the desired input demand, and not necessarily the number of vehicles discharged. When coding traffic volume, the user should not try to predict any changes to the number of arriving or discharging vehicles based on queue spillback, or congestion. The traffic model will take care of this.

If a lane-by-lane analysis is being performed, such that multiple link numbers are being used to represent a single turning movement, this input field displays the combined volume of all links within that turning movement. In this situation, it would be important to use the Traffic screen to code the exact volumes and saturation flow rates for each lane.

If the overall data set uses vehicle per hour (vph) coding, then the "unadjusted" hourly volumes are coded here. By contrast, the Traffic screen displays peak "adjusted" volumes. For example, suppose that 360 vehicles per hour were actually counted in the field for one particular turning movement. In this case, a volume of '360' would be coded in the Lane Configuration screen, regardless of any volume fluctuations or "peaking" within the hour.

If the overall data set uses vehicle per time period (vpp) coding, volumes arriving during the current time period are coded here, or on the Traffic screen. The "current time period" is displayed in the Time Period Quick Jump. For example, suppose that 100 vehicles were actually counted in the field during a 15-minute time period, for one particular turning movement. In this case, a volume of '100' would be coded in the Lane Configuration screen. Vehicle per time period (vpp) coding can be requested on the Options > Preferences screen. Refer to the Preferences section of the documentation for more information.

When tabbing through the input fields, the Lane Configuration screen initializes link-specific input data (for links with nonzero volume) by automatically generating record types 28, 29, and 291 within the TRANSYT-7F input data file. These record types must be present before any data is written to them. These record types are usually initialized early in the coding process by using File > New, or by adding a new turning movement in the Lane Configuration screen, or by coding a nonzero volume in the Traffic screen. The program is capable of running without record types 29 and 291, but it is preferable to have

these record types present for all existing links at all times (even if empty), in case they are ever needed. If a link does not exist, such as the southbound thru link number at a T-intersection with no southbound approach, no record types should be used to represent these links.

### Peak Hour Factor

This input field specifies the peak hour factor (PHF) for the three standard turning movements. When the peak hour factor is lower than 1.00, it implies that the user wishes to model the peak period within the hour, which experiences a somewhat higher flow rate than the hourly volume would indicate. It is usually calculated using the following formula:

$$PHF = \frac{\text{hourly volume (vph)}}{4 \times (\text{peak 15 minute volume})}$$

The peak hour factor should be coded as 1.00 under the following conditions:

- the duration of analysis is 60 minutes or more
- a multi-period analysis
- volumes are being specified in units of vehicles per time period (vpp)

For example, suppose that 360 vehicles per hour were actually counted in the field for one particular turning movement, such that a volume of '360' has been coded in the Lane Configuration screen. If 100 vehicles were counted for this turning movement in the field during the peak 15-minute time period, a PHF of '0.90' would be coded in the Lane Configuration screen.

Ultimately, the TRANSYT model is only responsive to the adjusted volume, and the duration of analysis. The input fields on this screen for PHF and unadjusted volume are only provided to assist the user in determining the adjusted volumes. If this assistance is not needed, it is perfectly acceptable to code the adjusted volumes directly into the Lanes or Traffic screens, and to accept the default PHF values of 1.00. For more advice on which types of volume and PHF coding are recommended for certain types of network conditions, refer to the Peak Hour Factor documentation in Appendix B.

### Turn Bay Storage (veh)

This input field specifies the queuing capacity for the left-turn and right-turn movements. Queuing capacities for each link can alternatively be coded on the Edit > Optional > Queuing screen, if desired. If the turning movement uses a full-length lane stretching from the upstream node to this node (i.e., there is no "turn pocket"), this field can be left blank, in which case the queuing capacity will be automatically estimated by the program as a function of the overall link length (displayed in the Distance field, above).

When a value is coded in this field, queue spillover input data are automatically coded by the program, if indeed a thru movement does exist on this approach. This data includes adjacent link numbers, queue blockage percentages, and turn bay flags. The Queuing screen can be loaded to view this data. If the thru movement has one lane, the queue spillover blockage percentage is assumed to be 99%, and the queue spillback blockage percentage is assumed to be 1%. If the thru movement has two lanes, the blockage percentage is assumed to be 49%. If the thru movement has three lanes, the blockage percentage is assumed to be 32%, and so on. If the user has manually coded a queue spillover percentage between 2% and 98%, based on their own judgment, these queue spillover percentages will not be automatically overwritten.

The Turn Bay Storage fields are disabled under certain circumstances, such as when no thru movement exists, or when a turning movement is shared. However, for advanced modeling, the user can still code queuing capacities for any or all movements on the Queuing screen. Advanced modeling in this sense might include queue spillover of a right-turn pocket into a full-length left-turn lane, or coding a thru movement queuing capacity that was inconsistent with the link length.

### Lane Use Diagram

The Lane Use Diagram is capable of displaying lane channelizations for up to seven lanes on the same approach. In each lane, below the arrows indicating what turning movements are allowed from each approach, the link number is listed. In the case of a shared lane, the primary link number (usually the thru movement) is listed. The format of each link number is: (node number)\*100 + (movement number), where the movement number reflects the user's chosen link numbering scheme.

By left-clicking on a lane inside the Lane Use Diagram, the user can automatically update the "Lane #" data field. By right-clicking on a lane inside the Lane Use Diagram, the user can easily add or delete any lane on the active approach. When using the "Intersection View" option, the active approach is determined based on which input fields have focus (i.e., location of the mouse cursor).

### Add/Delete

By selecting the Add/Delete button, the user can easily add or delete any lane on the active approach. This has the same effect as right-clicking on a lane inside the Lane Use Diagram, but provides the option of using the keyboard instead of the mouse.

## **Traffic Screen (associated with record types 28 and 29)**

The Traffic screen (Edit > Traffic) is illustrated below. This screen can be sent to the printer, or to an image file, by pressing Ctrl-P.

**TRANSYT-7F**

File Edit View Run Window Options Help

Output Quick Jump - Tin

FourNode.tin

**Traffic at University and 13th**

Eastbound			Westbound			Northbound			Southbound	
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru
106	105	111	108	107	112	102	101	109	104	103
Total Volume (vph)										
50	400	50	50	400	50	50	400	50	50	400
Saturation Flow Rate (vphg)										
1805	1900	0	1805	1900	0	1805	1900	0	1805	1900
Link Length (ft)										
1320	1320	1320	1320	1320	1320	1320	1320	1320	1320	1320
Mid Block Source Volume (vph)										
0	0	0	0	0	0	0	0	0	0	0
Start-Up Lost Time (sec)										
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Extension of Effective Green Time (sec)										

### Total Volume (vph or vpp)

Traffic volume is typically specified in units of vehicles per hour, regardless of the duration of simulation. This volume is specified as the total number of vehicles, including heavy vehicles. Later on, the saturation flow rate (and queuing headways) can be used to reflect the percentage of heavy vehicles in the analysis. Traffic volume is specified as the desired input demand, and not necessarily the number of vehicles discharged. When coding traffic volume, the user should not try to predict any changes to the number of arriving or discharging vehicles based on queue spillback, or congestion. The traffic model will take care of this.

An option is also available (through Options > Preferences) to specify volumes in units of vehicles per time period (vpp). Refer to the Preferences Screen section of the documentation for more details.

If one peak hour factor (PHF) can be assumed to apply to the entire network, the Volume Multiplier (Edit > Optional > Weighting screen, or record type 36) can be used to implement this. For a more detailed analysis involving different PHFs applied to different links, the adjusted volumes on this screen are generally equal to 4 times the peak 15-minute link volumes. Analysis period durations of 15 minutes (on the Edit > Analysis screen, or record type 1) are usually recommended for most cases. Multi-period simulation is yet a third option for analyzing traffic volume variations. When applied properly, multi-

period simulation is the most effective way to analyze severely oversaturated conditions.

Traffic volume turning movement counts should preferably be made in 15-minute interval groupings. These field counts are typically made during the morning peak, evening peak, and midday peak periods. The analyst should reconsider making any traffic counts during potentially unusual periods, such as Friday afternoons and Monday mornings.

Total volume is an important input parameter because it initializes all link-specific input data. Whenever the user clicks the "OK" button, the Traffic Screen initializes link-specific input data (for links with nonzero volume) by automatically generating record types 28, 29, and 291 within the TRANSYT-7F input data file. These record types must be present before any data is written to them. These record types are usually initialized early in the coding process by using File > New, or by adding a new turning movement in the Lane Configuration screen, or by coding a nonzero volume in the Traffic screen. The program is capable of running without record types 29 and 291, but it is preferable to have these record types present for all existing links at all times (even if empty), in case they are ever needed. If a link does not exist, such as the southbound thru link number at a T-intersection with no southbound approach, no record types should be used to represent these links.

Similarly, link-specific information and record types are un-initialized by coding a traffic volume of 0. The GUI also automatically removes un-initialized links from the signal timing plan (Edit > Timing screen), and removes un-initialized links from the list of feeding links (Edit > Feeders screen). The user should manually remove un-initialized links from coded progression routes, if any, on the Edit > Optional > Route screen.

Note that the TRANSYT traffic model requires traffic volumes to be at least 10 vph for any movement that exists. For this reason, if a value between 1 and 9 is coded here, a value of 10 will be coded by the program.

#### Saturation Flow Rate (vphg)

TRANSYT requires the adjusted saturation flow rate applicable to the entire link. For example, if a link has two lanes and an ideal saturation flow rate of 1900 vehicles per lane per hour of green (vplphg), the unadjusted saturation flow rate for the entire link would be 3800 vphg, and the adjusted saturation flow rate for the entire link could be a somewhat lower value such as 3600 vphg. These downward adjustments to the saturation flow rates are dictated by non-ideal conditions such as narrow lanes, uphill approaches, and the presence of heavy vehicles.

The saturation flow rate is used by the traffic flow simulation model to affect the departure of queues from a link. It is important that a reasonably accurate measure of the saturation flow rate be provided for each link. Experience has shown that saturation flow rates will be similar for links that have similar characteristics (such as location, geometry, turning movements, driver characteristics, etc.).

Suggested values for saturation flow rates (which should be confirmed by field data) are shown below. These guidelines assume right-hand drive, and protected turning movements. The reduction in saturation flow rates shown for protected turns accounts for the somewhat more sluggish departure of turning vehicles from queues. Presumably, it is more difficult for turning vehicles to stay in their lane, thus larger headways are required. Assuming right-hand drive, right-turners are typically assumed to travel at a speed of approximately 9 mph (14.4 kmph), at 85% of the thru movement saturation flow rate. Left-turners are typically assumed to travel at a speed of approximately 15 mph (24.0 kmph), at 95% of the thru movement saturation flow rate.



Typical Conditions (right-hand drive)	Saturation Flow (vphg/lane)		
	Left	Thru	Right
Conservative -- drivers avoid close headways	1520	1600	1360
Normal -- average driver patterns	1615	1700	1445
Aggressive -- drivers tailgate	1710	1800	1530
Buses (exclusive bus links)	475	700	360

More detailed guidelines are available in the HCM2000 [12] regarding analytical estimation (chapter 16, p. 11) and field measurement (chapter 16, p. 158) of saturation flow rates. Two of the TRANSYT-7F input screens (Satflow View, Saturation Flow Rate Calculator) implement some of these HCM guidelines, and these screens are documented later in this chapter.

The most appropriate method of saturation flow rate data entry might depend on the number of adjustments necessary. If conditions are fairly simple, adjusted saturation flow rates could be estimated using the general guidelines listed in the table above. Shared lane group links might require a volume-weighted average, such as the ideal saturation flow rate value times 0.925, if half of the traffic turns right. It is not necessary to code permitted movement saturation flow rates, because these capacities are determined by the permitted movement models. If numerous saturation flow adjustments are necessary for the protected movements, it might be best to use the one of the saturation flow rate calculation screens (Satflow View or Saturation Flow Rate Calculator), or to import HCS files (which would automatically import the adjusted saturation flow rates).

For links representing permitted turning movements, the saturation flow rate is significantly lower, since these vehicles must wait for gaps in traffic on the opposing links. Saturation flow rate determination is accomplished automatically by the internal permitted movement model. Thus, the saturation flow rate input field is not applicable for permitted-only movements. The "maximum flow rate" (MFR) for periods when the permitted movement is unopposed may be adjusted (from the internal default value) on the Edit > Optional > Permitted screen.

For protected-permitted left-turns, the saturation flow rate to be used during only the protected phase is coded on the Traffic screen. If multiple links are specified as being shared (Edit > Optional > Global > Shared Lanes screen, or record type 7), the saturation flow rate must only be coded for the primary link, and this saturation flow rate will apply collectively to all of the shared-lane links.

#### Link Length (ft or m)

TRANSYT expects the link length to be coded as the distance from upstream stop line to downstream stop line. In the case of a turn bay or turn pocket, the full signal-to-signal distance should be specified as the link length anyway, in order to achieve proper simulation of platoon dispersion. A turn bay storage (Edit > Lanes) or queuing capacity (Edit > Optional > Queuing) could be coded if the user wishes to simulate queue spillover, although this would have no effect on results if the queues were not long enough to fill the turn pocket.

For "external" links that do not have a nearby upstream signal, it is not necessary to code a link length. If no link length is coded, no total travel will be calculated for the link, and the link will not be considered in the calculation of network-wide average speed. The total travel time on the link will be equal to the total delay estimated by TRANSYT. Fuel consumption will be estimated for the link based on the default network-wide link speed, which is the value coded on the Edit > Optional > Global > Network Parameters screen (or record type 10). If the amount of queuing on an external link is important to the analysis, an appropriate link length should be specified.

When a link length is coded for an external link, total travel will be estimated, and the link will be considered in the calculation of network-wide average speed. Total travel time on the link will include both the delay, and the time spent at cruise speed. Fuel consumption estimates will be based on the

speed provided for feeding link #1 on the Edit > Feeders screen (or record type 28 field 9), or the network-wide value (Edit > Optional > Global > Network Parameters screen, or record type 10).

#### Mid-Block Source Volume (vph or vpp)

Parking lots, shopping centers, freeway off-ramps and other facilities that serve as mid-block sources of traffic demand are modeled as uniform input flows to the link. These sources should not be overlooked, because they may have a significant effect on the modeling of traffic flow. The value used for the uniform flow should be the total output volume from the source. TRANSYT automatically accounts for flow losses into such facilities, so it is not necessary to code any mid-block "sink" volumes that might exist in the field. As a general guide, a mid-block source should be modeled if its output constitutes 10% or more of the total volume on a link, or if it adds more than two vehicles per cycle.

Mid-block source volume is assumed to possess a uniform distribution when arriving at the intersection. It is only necessary to code this value for internal links on which feeding links have been coded, or will be coded, in the Edit > Feeders screen. External links, which have no nearby upstream signal and no feeding links, should not have any value coded in this field. On external links, all traffic volume is automatically simulated with a uniform distribution.

The volume here represents the total number of vehicles that enter the link in the middle of the block, and are assumed by TRANSYT to travel half the link length. These vehicles are assumed to enter the link at a constant rate throughout the cycle and, therefore, contribute an equal amount of flow to each time step of the flow profile. Mid-block flows may only be added to a link, and not subtracted from a link. The input flow rate adjustment automatically accounts for flow losses along a link. If a mid-block source flow is substantial and actually travels a distance that is significantly different than half the link length, then an unsignalized (single-phased) node could be modeled at the proper location. Mid-block source flow would then be modeled as an external link feeding the unsignalized node.

Mid-block source volume is a significant input parameter in TRANSYT, because non-progressed flow is not assumed to be uniform flow. Instead, progressed feeding flows will be factored up or down by the traffic model to exactly match the downstream flow, in the absence of any coded mid-block source volume. This differs from external links at which all volume is assumed to be uniform flow.

Links with significant mid-block flows have much less potential benefit from progression, and the optimization process is likely to treat them much differently. Omitting them from the input data could lead to inaccurate modeling and compromised output results.

#### Start-Up Lost Time (sec)

By default, this field displays the global or network-wide start-up lost time specified on the Edit > Optional > Global screen (or record type 1). If the link-specific value is modified to be something other than the global value, then the "addition to or subtraction from" the start-up lost time will be saved into the data set (onto record type 29). This entry may also be used to delay the start of effective green; for example, to account for delays to turning movements awaiting passage of pedestrians (if they are not explicitly modeled).

#### Extension of Effective Green Time (sec)

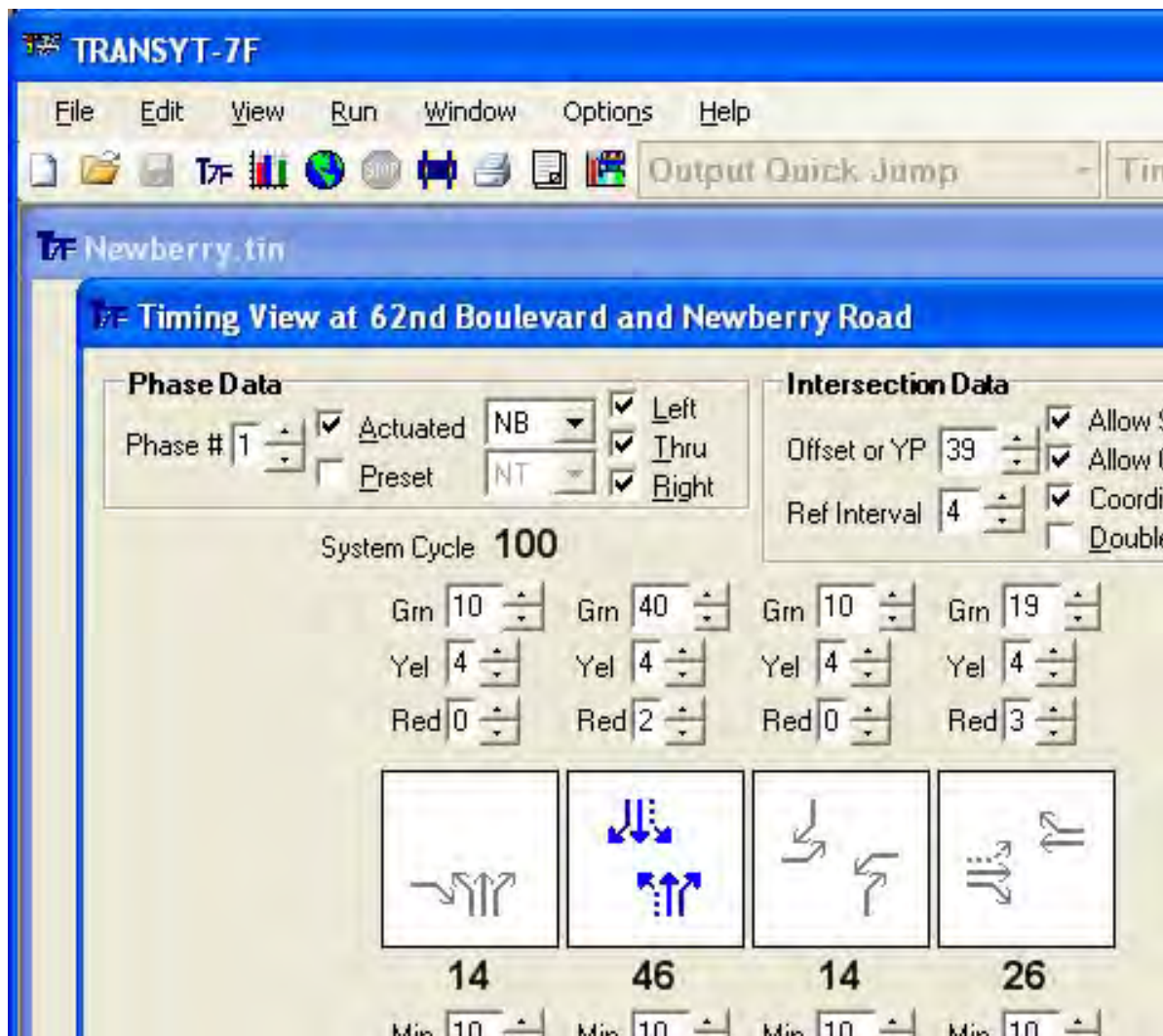
By default this field displays the global or network-wide extension of effective green time specified on the Edit > Optional > Global screen (or record type 1). If the link-specific value is modified to be something other than the global value, then the "addition to or subtraction from" the extension of effective green time will be saved into the data set (onto record type 29). Do not include additional time for sneakers. This is done automatically if sneakers are coded on the Edit > Optional > Permitted screen (or record type 29).

#### Record Types and Bottleneck Links

Record types 28 and 29 are automatically generated by the Traffic screen, but record types '-28' and '-29' can also be specified (using the record type editor) in the data file, in order to simulate bottleneck links. Bottleneck links are not associated with any numbered nodes, and bottleneck record types must be located after all intersection-specific record types in the data file (i.e., further down in the input file). Refer to the TRANSYT-7F sample file Daytona.tin, which is installed inside the Example folder, for an example of coding bottlenecks. If any bottlenecks are coded, the maximum number of nodes is reduced by one. There is a limit of 50 separate bottleneck links.

## Timing View Screen (associated with record types 1X, 18, 2Y, and 2Z)

The Timing View screen allows graphical coding and viewing of signal timing plans, with features such as the "phasing diagram" and "bar diagram". There are also "spinners" available for quickly coding interval durations, and a prominent display of the split durations, where a split is defined as the summation of green, yellow, and all-red interval durations. For additional timing plan-related information, refer to the documentation section for the Edit > Optional > Timing screen, or Traffic Signal Timing Elements and Characteristics. This screen can be sent to the printer, or to an image file, by pressing Ctrl-P. The Timing View (Edit > Timing) screen is illustrated below.



A "non-graphical" Timing screen is also available by clicking on the "Coding" pushbutton at the bottom of the Timing View screen, or by selecting Edit > Optional > Timing. The non-graphical Timing screen also displays the link numbers moving during each phase, the interval durations, and the actuated flags. All of this information is automatically updated by the Timing View screen. For coding complicated timing plans, it may become necessary to use the non-graphical Timing screen. Examples of such complex timing plans (possibly requiring use of the non-graphical Timing screen) include:

- intersections with more than 4 approaches
- intersections with more than 3 movements per approach
- two separate and unequal green displays in a single cycle for any link
- links with continuous, 100% green operation throughout the cycle
- intersections with exclusive pedestrian phases
- intersections where lane-by-lane analysis is being performed
- phases with multiple green (fixed and variable) intervals defined

If the timing plan is coded after a lane-by-lane analysis has been requested for that node on the Lane Configuration screen, the Timing View screen automatically includes those extra links within the timing plan. Otherwise, adding "extra" links 13 and higher (for lane-by-lane analysis, or for 5+ intersection approaches) into the timing plan may require using the non-graphical Timing screen, which is more flexible than the Timing View screen.

#### Add/Delete

By selecting the Add/Delete button, the user can easily add or delete any phase at the active intersection. This has the same effect as right-clicking on a phase inside the Phasing Diagram, but provides the option of using the keyboard instead of the mouse.

#### Export

The Export button is enabled for direct CORSIM analyses, or when a related HCS-Signals file exists in the same folder as the TRANSYT input (\*.TIN) file. The Export button would allow exporting of cycle length, splits, and offsets to the corresponding node in the CORSIM input (\*.TRF) file. To export network-wide timing to CORSIM, select File > Export instead. The Export button would also allow exporting of cycle length and splits to the corresponding HCS file. In this case, the Intersection Name in the TIN file would need to be consistent with the HCS filename. This occurs when HCS files are imported into TRANSYT-7F, or when right-clicking on a node in the Map View to "Link to HCS". When "one-touch" CORSIM animation and one-touch HCS analysis are used to generate CORSIM and HCS files from scratch, all data are exported, and there is no need to use the Export button.

#### **Phase Data** (associated with record types 2Y and 2Z)

TRANSYT currently does not support changes to the phasing sequence between different time periods. As such, when a multi-period analysis file is loaded, input data within the Phase Data frame can only be coded when time period #1 is selected.

#### Phase Number

This input field is used to indicate which phase will be edited by the user. Subsequently, the user can specify whether or not the phase is actuated, and which movements are active during that phase. This field can also be used to add another phase to the timing plan. For example, if there are four phases in the timing plan, a fifth phase can be added by specifying '5' in this field, and coding a new movement for this phase using the Left/Thru/Right checkboxes.

#### Actuated

This checkbox is used to indicate whether or not this phase is actuated. In the phasing diagram, non-actuated phases are drawn using dark blue arrows, whereas actuated phases are drawn using light gray arrows. Within the TRANSYT model, if any individual movement is capable of actuating or terminating the phase, the entire phase is considered actuated. In the phasing diagram, the arrows for every individual movement will be colored gray for an actuated phase, even if some of the movements are non-actuated.

#### NB/SB/EB/WB

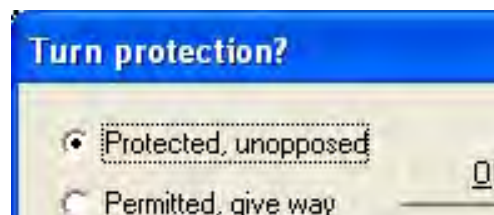
This combo box is used to indicate which approach will be edited by the user. Subsequently, the user can specify which movements are active (during the "current" phase) from this approach.

#### Left/Thru/Right

These checkboxes are used to indicate which movements are active during the current phase. These checkboxes can also be used to add or delete phases from the timing plan. For example, if there are four phases in the timing plan, a fifth phase can be added by specifying '5' in the Phase Number field, and coding a new movement for phase #5 using these checkboxes. To delete a phase from the timing plan, simply uncheck all boxes to indicate that there are no active movements during the given phase. The phasing diagram will update itself accordingly.

The Timing View screen does not allow the user to code movements within the timing plan unless a link has already been defined for that movement. Therefore, when starting a brand-new file, it is necessary to begin by specifying traffic volumes within the Edit > Lanes or Edit > Traffic screens before coding the timing plan.

It is usually obvious whether the individual movements in the timing plan are protected or permitted. For example, thru movements are almost always protected, except at sign-controlled approaches. When right-hand drive is in effect, left-turns are always permitted if the opposing thru movement runs during the same phase. In cases of "ambiguous turn protection", the dialog box below will pop up, so the user can specify the exact type of turn protection they want.



The case of a left-turn opposed by a right-turn, but not opposed by a thru movement, gives rise to ambiguous turn protection. For right-hand drive, the left-turn might be coded as a permitted movement if it yields to the opposing right-turn, but it might be coded as a protected movement if there are multiple receiving lanes. Another case of ambiguous turn protection arises for turning movements that may or may not yield to thru movements on the conflicting approach. In cases such as this, when the Left/Thru/Right checkbox is turned on, the Turn Protection dialog box will pop up, so the user can specify the exact type of turn protection they want.

#### Preset Phasing

This feature allows the input coding of complete phasing sequences to be faster and easier. If the preset phasing feature is not used, the user must code the phasing sequence by manually selecting the individual active movements for each phase. However, with preset phasing, the user can select from

a list of predefined phasing sequences, and the software will instantly code the entire phasing sequence. Certain complex phasing sequences must still be manually coded by the user, although these can sometimes be handled after letting the software initially code the "baseline" preset phasing sequence.

User inputs associated with preset phasing are located in the "Phase Data" frame toward the upper left-hand corner of the Timing View screen. The preset phasing feature must be used with caution, because it can overwrite any existing or previously coded phasing sequence. Because of this, the "Preset" checkbox is toggled off by default. Turning on the Preset checkbox enables the adjacent combo box that contains a predefined list of available phasing sequences. The available phasing sequences on this list are a function of the existing movements that the user has coded on the Edit > Lanes and Edit > Traffic screens. The available phasing sequences are also a function of the currently active approach direction from the directional combo box, which specifies either "NB" (northbound), "SB" (southbound), "EB" (eastbound), or "WB" (westbound). When a previously coded timing plan is initially loaded into the Timing View screen, the combo box may display one of the preset phasing sequences, if one has been detected.

The preset phasing sequences are specified using the same "phase codes" employed by the phasing sequence optimization methodology. For example, "ETW" implies lead-lag phasing that begins in the eastbound direction. One exception to this is the phase code of "G", which represents a single-phase "green ball". This phase code is available for preset phasing, even though it is not applicable to phasing sequence optimization.

When a preset phasing sequence is selected by the user from the combo box, it is automatically coded beginning at the currently active phase number, specified in the Phase Number field (upper left-hand corner of the Timing View screen). For example, if the user selects "NTS" when phase #3 is displayed, the new phasing sequence will be automatically coded within phases 3 through 5. However, if the existing timing plan only contains 4 phases, the final phase 5 would not be written. In some cases it may be desirable or necessary to manually add or delete phases, prior to using preset phasing. In addition, when the preset phasing sequence is automatically written, the phase specifications are affected by the Actuated checkbox. That is, if the Actuated checkbox is toggled on at the time when a preset phasing sequence is chosen, then the entire preset phasing sequence will be coded as actuated. In timing plans that contain semi-actuated control, it may be necessary to manually specify the major street non-actuated phase, after using the preset phasing feature. Even so, coding the entire phasing sequence or timing plan should usually be much faster when using preset phasing.

In certain cases, the preset phasing feature will automatically code permitted movements for the user. In other cases, the user may have to code permitted movements manually, after the preset phasing sequence has been displayed. For example, assuming a right-handed system, suppose that all three turning movements (L, T, R) have been previously defined on the Edit > Traffic screen in the northbound and southbound direction, and the user has selected the preset phasing sequence of "ST". This phasing sequence specifies southbound movements only in phase #1, followed by both through movements in phase #2. In this case, the software would automatically code the northbound permitted left-turn in phase #2, because this would be the only phase in which it could possibly move. However, if the user wishes to optionally analyze permitted southbound left-turns as well in phase #2, they would have to code that manually.

The preset phasing combo box is automatically disabled when a two-way stop-controlled (TWSC) intersection is in effect, or when a direct CORSIM analysis is in effect. When a TWSC intersection is in effect, the Timing View screen displays only one "phase", and all existing movements should be coded manually for this phase. When a direct CORSIM analysis is in effect, the phasing sequence should not be changed in the TRANSYT input (\*.TIN) file, because this would create an inconsistency with the associated CORSIM input (\*.TRF) file.

## **Intersection Data** (associated with record types 1X and 18)

### Offset or YP

The offset or yield point for the current intersection is displayed in this field. The user can code existing offsets to evaluate them via simulation, or simply leave the offsets coded as zero, and allow the program to optimize them. The term "offset" is applicable to coordinated, non-actuated signals. The term "yield point" is applicable to coordinated, actuated signals. Yield points are typically referenced to the major street through movement yellow interval, because this is the guaranteed point during the cycle where control can be yielded to the actuated phases. When a multi-period analysis file is loaded, this is the only input field within the Intersection Data frame that can be coded uniquely across multiple time periods. The Offset or YP input field and the Reference Interval input field are both automatically disabled for uncoordinated signals.

#### Reference Interval

TRANSYT allows offsets and yield points to be referenced to any interval. For coordinated actuated controllers, it is recommended to reference yield points to the major street through movement yellow interval, as mentioned above. This is consistent with the operation of the controller, and allows for consistency between analysts as well. If some other reference interval is chosen, the model results will still be correct, but it may be necessary to "translate" these values prior to implementation in the field.

The reference interval setting provides a relationship between signalized intersections, with the intent of achieving good progression, and minimizing stops. For example, suppose the timing plan at the "master controller" or "master intersection" begins at time zero, and the reference interval at the master controller is interval #1. If the offset at a downstream intersection is 40 seconds, and is referenced to interval #7, this means that interval #7 downstream begins exactly 40 seconds after interval #1 begins at the master controller.

#### Allow Split Optimization

This checkbox specifies whether hill-climb or genetic algorithm optimization of splits will be allowed at this intersection. This checkbox is only applicable if split optimization has been requested (on the Edit > Analysis screen) for the entire network. If this checkbox is not selected, but the splits are being modified anyway during a run, this is probably due to the initial timing model. In order to de-activate the initial timing model for one node (but not others), it is necessary to remove the intersection node number from the optimization node list. If this checkbox is disabled, it implies that a "simulation-only" run is currently requested on the Edit > Analysis screen.

#### Allow Offset Optimization

This checkbox specifies whether or not genetic algorithm optimization of offsets will take place at this intersection. This checkbox is only applicable when optimization of offsets has been requested globally on the Edit > Analysis screen. If this checkbox is not selected, but the offsets are being modified anyway during a run, this is possibly due to cycle length optimization, or the initial timing flag. If hill-climb optimization has been selected, offset optimization cannot be prevented through the use of this checkbox (except for grouped nodes, and the master controller). If this checkbox is disabled, it implies that 1) a "simulation-only" run is currently requested on the Edit > Analysis screen, or 2) the Uncoordinated Network checkbox from the Edit > Analysis screen has been turned on, or 3) there is only one intersection in the data file, or 4) this intersection is the master controller.

#### Coordinated Signal

This checkbox specifies whether the controller at this intersection is coordinated with other controllers in the network. By default this checkbox should be turned on at every intersection, so that platoon dispersion can be simulated properly. Flow profiles can be observed by using the Profile View screen.

Whenever the Coordinated Signal checkbox is turned off, this means the flow profile will

simulated as perfectly flat or uniform, and there will never be an incoming platoon. Therefore this checkbox should only be turned off if there are no nearby upstream signals, or if the subject intersection has a different cycle length than the background system cycle length. Also note that the initial timing flag cannot be used to generate initial timings for uncoordinated signals.

Uncoordinated signals are allowed to have cycle lengths that are different from the network background cycle length. When using genetic algorithm optimization, it is possible to optimize the cycle length and splits at an uncoordinated intersection independently of the network cycle length. However, it is not possible to achieve reliable progression through intersections that are not coordinated. If this checkbox is disabled, it implies that the Uncoordinated Network checkbox from the Edit > Analysis screen has been turned on.

### Double Cycled

This checkbox specifies whether the controller at this intersection is to be double-cycled. It is common to double-cycle at minor intersections where traffic demand is light. This means to operate the local controller on a cycle length exactly half of the system cycle length. For example, if the system cycle length is 120 seconds, the local double cycle length is 60 seconds, and interval times coded on the Timing screen should sum to 60 seconds. Delays, and even possibly stops, will almost certainly be reduced. However, the phasing sequence cannot be optimized.

If progression is not needed, double-cycling is not needed. Instead the Coordinated Signal checkbox should be turned off, and an appropriate isolated cycle length should be chosen, possibly by Uncoordinated Optimization. Refer to the Uncoordinated Optimization section of the documentation for more details.

At double-cycled nodes, the user is limited to coding a maximum of three phases, because TRANSYT uses three more for the second (half) cycle. Likewise, only code the minimum phase durations for one half cycle. To model a double-cycled node with four phases or more, simply code it as a single-cycled node, but with the phase durations doubled.

If this checkbox is disabled, it implies that 1) the Uncoordinated Network checkbox from the Edit > Analysis screen has been turned on, or 2) there is only one intersection in the data file, or 3) this intersection is the master controller. For more information, refer to the Double Cycling section of the documentation.

**Timing Data** (associated with record types 1X, 18, 2Y, and 2Z)

### System Cycle, Local Cycle

The Timing View screen displays the cycle length at the current signalized intersection, but does not allow the user to directly edit the cycle length value. Instead, the cycle length value is dynamically updated in response to user inputs. The displayed value represents the cycle length that will be simulated by TRANSYT. The cycle length that will be simulated depends on whether or not this is a coordinated signal. If the Coordinated Signal checkbox is turned on, and the Uncoordinated Network checkbox (in the Edit > Analysis screen) is turned off, the signal is assumed to be coordinated.

If the signal is coordinated, the system cycle length will always be displayed. If the initial timing flag is activated in the Edit > Analysis screen, green times will be re-designed by the initial timing model prior to simulation, and the user-coded green times will be ignored. If the initial timing flag is deactivated, TRANSYT will try to simulate the user-coded green times. If a discrepancy exists between the system cycle length and the local cycle length, the local cycle length will be displayed directly to the right of the system cycle length. If the user-coded green times add up to a value that is lower than the system cycle length, the model will simulate extra green time during the final phase, to avoid a fatal error. If the user-coded green times add up to a value that is higher than the system cycle length, the model will issue a fatal error message.



If the signal is uncoordinated, the local cycle length will be displayed. The displayed cycle length will be the sum of all user-coded green, yellow, and all-red intervals on the Timing View screen.

### Major Street Direction

This combo box is only enabled and visible for sign-controlled intersections. This combo box specifies whether the major street through movements, which do not yield to any other movements, are located in either the East-West or North-South direction.

### Grn

This input field is used to indicate the green interval duration. Pressing <PageUp> or <PageDown> (to switch between intersections) when the cursor lies within one of the interval fields (Grn, Yel, Red) causes the cursor to switch to the "Node Jump" combo box, in case the number of phases changes (such that the original interval field doesn't exist).

If the initial timing flag is on, user-coded green times will be ignored, and could even be coded as zero seconds. If the initial timing flag is off, user-coded green times will simulated explicitly unless interval times don't sum to the cycle length. If interval times sum to a value lower than the cycle length, the final-phase green time will be artificially increased. If interval times sum to a value higher than the cycle length, a fatal error message will be issued.

TRANSYT treats user-coded green times just like HCS does; they are considered average times. The only exception to this is when the "estimation" procedure is used, in which case the user-coded green times are treated as maximums. TRANSYT provides minimum phase times for optimization in separate input fields, and for the estimation procedure they are interpreted as actuated minimum phase times.

### Yel

This input field is used to indicate the yellow interval duration. If in reality there is no yellow or all-red change interval, a dummy yellow interval must be coded nonetheless. Overall minimum phase times must be at least 2 seconds to account for a 1-second green interval and a 1-second yellow interval. If the user wishes to simulate an overlap phase as small as 1 second, this can be accomplished if the start-up lost times and extension of effective green times are correctly coded. On the surface, it is necessary to either specify a minimum 2-second overlap, or no overlap. Refer to the Overlap Phasing section of the documentation for more details.

### Red

This input field is used to indicate the all-red interval duration. The user can optionally combine the yellow and all-red times into the yellow interval, and code an all-red interval duration of zero, because combining these interval durations into one should not have an effect on the modeling or the results. The maximum value that can be coded is 5 seconds. If the user wishes to model a longer interval with no vehicles moving, this can be accomplished by coding a "dummy" phase with green time but no moving links.

### Phasing Diagram

Along with the bar diagram, the phasing diagram is one of the primary features of the Timing View screen. The phasing diagram provides a graphical representation of the signal timing plan. On the phasing diagram, non-actuated phases are illustrated using dark blue arrows, whereas actuated phases are illustrated using light gray arrows. Protected movements are illustrated using arrows with solid lines, whereas permitted movements are illustrated using arrows with dashed lines. Right-hand driving is illustrated using arrows on the right-hand side, whereas left-hand driving is illustrated using arrows on the left-hand side.

By left-clicking on a phase inside the Phasing Diagram, the user can automatically update the "Phase #" data field. By right-clicking on a phase inside the Phasing Diagram, the user can easily add or delete any phase at the active intersection.

After the timing plan has been coded and saved, the record type data (1X, 18, 2Y, and 2Z) are updated in the background. Record type data may also be viewed by clicking on the "Coding" pushbutton. The phasing diagram is limited to illustrating four approaches, however, so any active movements on approaches 5+ are not illustrated. In order to code complex timing plans involving more than three lane groups per approach, or more than four approaches, it is necessary to click on "Coding" to access the original Edit > Optional > Timing screen. From the original Timing screen, it is possible to specify any number of links moving during any given phase.

When coding a brand-new phasing sequence, certain left-turn and right-turn movements are drawn with dashed arrows and assumed to be permitted movements, based on the presence of opposing or conflicting thru movements. However, the user should pay close attention to the situation where left-turn and right-turn movements may or may not yield to each other. For example, suppose that a left-turn and an opposing right-turn move together in the same phase, but with no thru movements on either side. Depending on the number of receiving lanes, these movements could be either permitted, or protected. When a turning movement yields to an opposing turn movement, the user should open the Edit > Optional > Permitted screen, to code the appropriate opposing link number(s).

### Min

This input field is used to indicate the minimum phase time, where the phase time is the summation of green, yellow, and all-red intervals. The minimum phase time parameter is primarily intended to act as a constraint on the optimization process. However, the software does not allow timing plans evaluated via simulation-only to violate the minimum phase times, either. When a multi-period analysis file is loaded, minimum phase times can only be coded when time period #1 is selected.

Typically the minimum phase times should reflect driver expectation and/or ped/bike crossing time requirements. In some cases, experienced users have inflated minimum phase times for critical movements, in order to speed up the optimization process. This is often a good idea, but is sometimes risky. Occasionally, green time must be taken away from important movements, in order to optimize performance of the entire network.

## **Bar Diagram**

Along with the phasing diagram, the bar diagram is one of the primary features of the Timing View screen. The bar diagram provides a time-space representation of the signal timing plan, moving from left to right. The horizontal distance of the bar diagram represents the duration of the cycle length, and the bar is subdivided into smaller pieces that represent interval durations. The width of each interval bar is proportional to each interval duration, even though the width of the entire bar diagram always spans the entire Timing View screen. The bar diagram is also consistent with the phasing diagram, in that the initial intervals illustrated on the left side are associated with the first phase, and the final intervals on the right side are associated with the final phase.

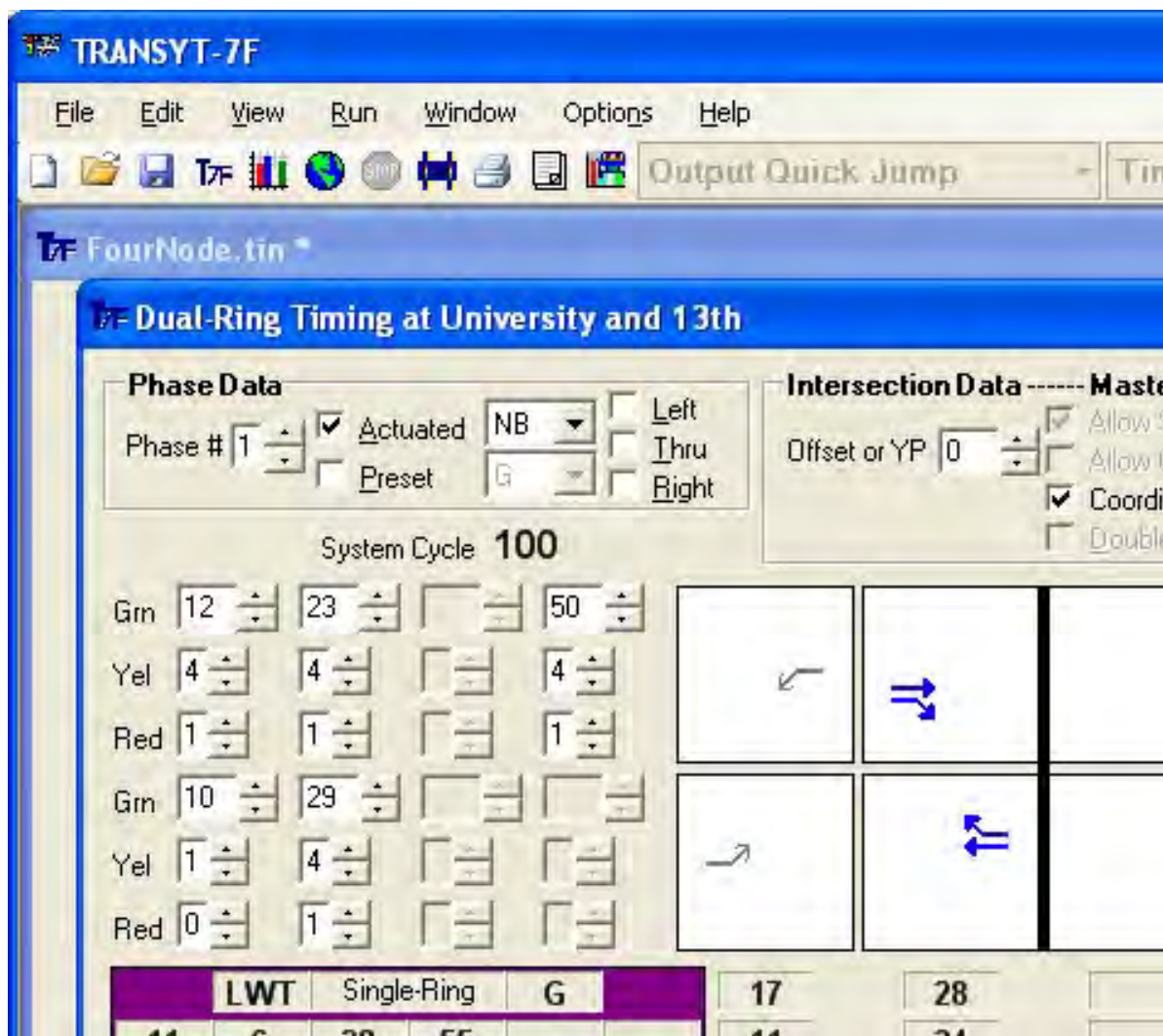
The bar diagram uses the colors green, yellow, and red to indicate the green intervals, yellow intervals, and all-red intervals, respectively. Optionally, the yellow and all-red intervals can be illustrated using a "cross hatch" pattern, using the Options > Preferences screen. By default, the bar diagram is displayed using only the colors green, yellow, and red, to distinguish between the different signal intervals. The "cross hatch" option provides another mechanism besides color for distinguishing between the different signal intervals. When the cross hatch option is activated, yellow intervals are drawn using a single diagonal line as the fill pattern, in addition to the yellow coloring. All-red intervals are drawn using double diagonal lines as the fill pattern, in addition to the red coloring.

The TRANSYT model does not distinguish between yellow and all-red intervals. The software does require a minimum yellow time of 1 second, however. All-red intervals are optional and can be coded with a duration of zero seconds.

The reference interval is illustrated on the bar diagram using a vertical arrow, and the word "Ref". This arrow indicates the point within the signal cycle to which the offset or yield point value is referenced. The offset or yield point is assumed to reference the beginning of the reference interval, and thus the arrow will point to the left-hand side of the reference interval. Within the model, the reference interval settings (and the offset and yield point settings, for that matter) have no effect on the performance of uncoordinated signals.

## **Dual-Ring Timing Screen (associated with record types 1X, 18, 2Y, and 2Z)**

The Dual-Ring Timing screen allows 8-phase NEMA-style coding and viewing of signal timing plans. Because TRANSYT-7F only stores signal timing data in single-ring format, this screen displays a dual-ring "interpretation" of the timing plan. In many cases, the dual-ring timing is an exact representation of the single-ring timing, with no room for ambiguity. In other cases, such as with "tight" overlap phasing, the screen simply displays one of several possible interpretations. If the user changes timing and tries to save data, either by switching node numbers or clicking on the "OK" button, the screen pops up a dialog box showing the equivalent single-ring timing plan about to be stored. The Dual-Ring Timing (Edit > Timing > Dual-Ring) screen, illustrated below, can be sent to the printer (or to an image file) by pressing Ctrl-P.



By default, phases 1-2-5-6 (left side of the barrier) are used to indicate east-west phases, whereas phases 3-4-7-8 (right side of the barrier) are used to indicate north-south phases. Also by default, the equivalent single-ring timing plan is displayed towards the bottom of the screen (purple area). These default behaviors can be reversed on the Options > Preferences screen.

The sample screen image above illustrates tight overlap phasing in the east-west direction. The westbound (WB) left-turn has only 2 seconds more green time than the eastbound (EB) left-turn. When the combined yellow and all-red time (5 seconds) is greater than or equal to the overlap time (2 seconds), translating the timing plan to single-ring format becomes more complicated. Single-ring phase #1 gets 10 seconds of green plus 1 second of yellow. The yellow time is reduced below what would be observed in the field, to make the overlap time shorter than the combined yellow and all-red time. Single-ring phase #2 gets 1 second of green, 4 seconds of yellow, and 1 second of all-red. This causes the WB left-turn to receive exactly 2 seconds more green time than the EB left-turn (EB yellow and all-red are used as green time by the WB left-turn). For printing and reporting purposes, the yellow and all-red times could be increased (to 4 and 1) in dual-ring phase #5, and the green time could be reduced (to 25) in dual-ring phase #6. However, this would have no effect on the equivalent single-ring timing plan.

Start-up lost time (SLT) and extension of effective green time (EEG), coded on the Edit > Traffic screen, play an important role in the precise modeling of tight overlap phasing. As illustrated by the two

bar diagrams, SLT for the westbound thru and right-turn would clearly need to be increased by 4 seconds in the above example, to prevent them from moving too early in the cycle. Also, EEG for the eastbound left-turn could be increased by 1 second, if the user wanted to model a 1-second overlap.

The Dual-Ring Timing screen is designed to handle a fixed set of standard phasing sequences at a traffic signal, and is not designed to handle unsignalized intersections. For coding complicated timing plans, it may become necessary to use the single-ring Timing View screen, or the non-graphical Timing screen. Examples of complex timing plans could include:

- intersections with more than 4 approaches
- intersections with more than 3 movements per approach
- two separate and unequal green displays in a single cycle for any link
- links with continuous, 100% green operation throughout the cycle
- intersections with exclusive pedestrian phases
- intersections where lane-by-lane analysis is being performed
- phases with multiple green (fixed and variable) intervals defined

If the timing plan is coded after a lane-by-lane analysis has been requested for that node on the Lane Configuration screen, the Dual-Ring Timing screen automatically includes those extra links within the timing plan. Otherwise, adding "extra" links 13 and higher (for lane-by-lane analysis, or for 5+ intersection approaches) into the timing plan may require using the non-graphical Timing screen, which is more flexible than the Dual-Ring Timing screen.

#### **Phase Data** (associated with record types 2Y and 2Z)

TRANSYT currently does not support changes to the phasing sequence between different time periods. As such, when a multi-period analysis file is loaded, input data within the Phase Data frame can only be coded when time period #1 is selected.

##### Phase Number

This input field is used to indicate which phase will be edited by the user. Subsequently, the user can specify whether or not the phase is actuated, and which movements are active during that phase. The user can also change the active phase number by clicking on one of the eight white phase boxes, from the phasing diagram.

##### Actuated

This checkbox is used to indicate whether or not this phase is actuated. In the phasing diagram, non-actuated phases are drawn using dark blue arrows, whereas actuated phases are drawn using light gray arrows. Arrows for every individual movement will be colored gray in an actuated phase, even if some of the movements are non-actuated.

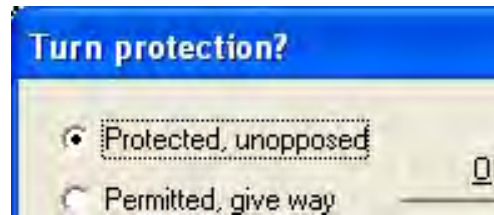
##### NB/SB/EB/WB

This combo box is used to indicate which approach will be edited by the user. Subsequently, the user can specify which movements are active (during the "current" phase) from this approach.

##### Left/Thru/Right

These checkboxes are used to indicate which movements are active during the current phase. The input screen does not allow the user to code movements within the timing plan unless a link has already been defined for that movement. Therefore, when starting a brand-new file, it is necessary to begin by specifying traffic volumes within the Edit > Lanes or Edit > Traffic screens before coding the timing plan.

When coding a new dual-ring phasing sequence, it is often unclear whether turning movements in the timing plan will ultimately be protected or permitted. In such cases, the dialog box below will pop up, so the user can specify the exact type of turn protection they want.



### Preset Phasing

This feature allows the input coding of complete phasing sequences to be faster and easier. If the preset phasing feature is not used, the user must code the phasing sequence by manually selecting the individual active movements for each phase. However, with preset phasing, the user can select from a list of predefined phasing sequences, and the software will instantly code the entire phasing sequence. Certain complex phasing sequences must still be manually coded by the user, although these can sometimes be handled after letting the software initially code the "baseline" preset phasing sequence.

User inputs associated with preset phasing are located in the "Phase Data" frame toward the upper left-hand corner of the Dual-Ring Timing screen. The preset phasing feature must be used with caution, because it can overwrite any existing or previously coded phasing sequence. Because of this, the "Preset" checkbox is toggled off by default. Turning on the Preset checkbox enables the adjacent combo box that contains a predefined list of available phasing sequences. The available phasing sequences on this list are a function of the existing movements that the user has coded on the Edit > Lanes and Edit > Traffic screens. The available phasing sequences are also a function of the currently active approach direction from the directional combo box, which specifies either "NB" (northbound), "SB" (southbound), "EB" (eastbound), or "WB" (westbound). When a previously coded timing plan is initially loaded into the Dual-Ring Timing screen, the combo box may display one of the preset phasing sequences, if one has been detected.

The preset phasing sequences are specified using the same "phase codes" employed by the phasing sequence optimization methodology. For example, "ETW" implies lead-lag phasing that begins in the eastbound direction. One exception to this is the phase code of "G", which represents a single-phase "green ball". This phase code is available for preset phasing, even though it is not applicable to phasing sequence optimization.

In certain cases, the preset phasing feature will automatically code permitted movements for the user. In other cases, the user may have to code permitted movements manually, after the preset phasing sequence has been displayed. For example, assuming a right-handed system, suppose that all three turning movements (L, T, R) have been previously defined on the Edit > Traffic screen in the northbound and southbound direction, and the user has selected the preset phasing sequence of "ST". This phasing sequence specifies southbound movements only, followed by both through movements. In this case, the software would automatically code the northbound left-turn as permitted, because it could not possibly be protected. However, if the user wishes to optionally analyze southbound left-turns as protected-permitted, they would have to code that manually.

The preset phasing combo box is automatically disabled when a two-way stop-controlled (TWSC) intersection is in effect. When a direct CORSIM analysis is in effect, the phasing sequence should usually not be changed in the TRANSYT input (\*.TIN) file, because this would create an inconsistency with the associated CORSIM input (\*.TRF) file.

## **Intersection Data** (associated with record types 1X and 18)

### Offset or YP

The offset or yield point for the current intersection is displayed in this field. The user can code existing offsets to evaluate them via simulation, or simply leave the offsets coded as zero, and allow the program to optimize them. The term "offset" is applicable to coordinated, non-actuated signals. The term "yield point" is applicable to coordinated, actuated signals. Yield points are typically referenced to the major street through movement yellow interval, because this is the guaranteed point during the cycle where control can be yielded to the actuated phases. When a multi-period analysis file is loaded, this is the only input field within the Intersection Data frame that can be coded uniquely across multiple time periods. The Offset or YP input field is automatically disabled for uncoordinated signals.

### Allow Split Optimization

This checkbox specifies whether hill-climb or genetic algorithm optimization of splits will be allowed at this intersection. This checkbox is only applicable if split optimization has been requested (on the Edit > Analysis screen) for the entire network. If this checkbox is not selected, but the splits are being modified anyway during a run, this is probably due to the initial timing model. In order to de-activate the initial timing model for one node (but not others), it is necessary to remove the intersection node number from the optimization node list. If this checkbox is disabled, it implies that a "simulation-only" run is currently requested on the Edit > Analysis screen.

### Allow Offset Optimization

This checkbox specifies whether or not genetic algorithm optimization of offsets will take place at this intersection. This checkbox is only applicable when optimization of offsets has been requested globally on the Edit > Analysis screen. If this checkbox is not selected, but the offsets are being modified anyway during a run, this is possibly due to cycle length optimization, or the initial timing flag. If hill-climb optimization has been selected, offset optimization cannot be prevented through the use of this checkbox (except for grouped nodes, and the master controller). If this checkbox is disabled, it implies that 1) a "simulation-only" run is currently requested on the Edit > Analysis screen, or 2) the Uncoordinated Network checkbox from the Edit > Analysis screen has been turned on, or 3) there is only one intersection in the data file, or 4) this intersection is the master controller.

### Coordinated Signal

This checkbox specifies whether the controller at this intersection is coordinated with other controllers in the network. By default this checkbox should be turned on at every intersection, so that platoon dispersion can be simulated properly. Flow profiles can be observed by using the Profile View screen.

Whenever the Coordinated Signal checkbox is turned off, this means the flow profile will be simulated as perfectly flat or uniform, and there will never be an incoming platoon. Therefore this checkbox should only be turned off if there are no nearby upstream signals, or if the subject intersection has a different cycle length than the background system cycle length. Also note that the initial timing flag cannot be used to generate initial timings for uncoordinated signals.

Uncoordinated signals are allowed to have cycle lengths that are different from the network background cycle length. When using genetic algorithm optimization, it is possible to optimize the cycle length and splits at an uncoordinated intersection independently of the network cycle length. However, it is not possible to achieve reliable progression through intersections that are not coordinated. If this checkbox is disabled, it implies that the Uncoordinated Network checkbox from the Edit > Analysis screen has been turned on.

### Double Cycled

This checkbox specifies whether the controller at this intersection is to be double-cycled. It is common to double-cycle at minor intersections where traffic demand is light. This means to operate the local controller on a cycle length exactly half of the system cycle length. For example, if the system cycle length is 120 seconds, the local double cycle length is 60 seconds, and interval times coded on the Timing screen should sum to 60 seconds. Delays, and even possibly stops, will almost certainly be reduced. However, the phasing sequence cannot be optimized.

If progression is not needed, double-cycling is not needed. Instead the Coordinated Signal checkbox should be turned off, and an appropriate isolated cycle length should be chosen, possibly by Uncoordinated Optimization. Refer to the Uncoordinated Optimization section of the documentation for more details.

If this checkbox is disabled, it implies that 1) the Uncoordinated Network checkbox from the Edit > Analysis screen has been turned on, or 2) there is only one intersection in the data file, or 3) this intersection is the master controller. For more information, refer to the Double Cycling section of the documentation.

### **Timing Data** (associated with record types 1X, 18, 2Y, and 2Z)

#### System Cycle, Local Cycle

The Timing View screen displays the cycle length at the current signalized intersection, but does not allow the user to directly edit the cycle length value. Instead, the cycle length value is dynamically updated in response to user inputs. The displayed value represents the cycle length that will be simulated by TRANSYT. The cycle length that will be simulated depends on whether or not this is a coordinated signal. If the Coordinated Signal checkbox is turned on, and the Uncoordinated Network checkbox (in the Edit > Analysis screen) is turned off, the signal is assumed to be coordinated.

If the signal is coordinated, the system cycle length will always be displayed. If the initial timing flag is activated in the Edit > Analysis screen, green times will be re-designed by the initial timing model prior to simulation, and the user-coded green times will be ignored. If the initial timing flag is deactivated, TRANSYT will try to simulate the user-coded green times. If a discrepancy exists between the system cycle length and the local cycle length, the local cycle length will be displayed directly to the right of the system cycle length. If the user-coded green times add up to a value that is lower than the system cycle length, the model will simulate extra green time during the final phase, to avoid a fatal error. If the user-coded green times add up to a value that is higher than the system cycle length, the model will issue a fatal error message. If the signal is uncoordinated, the local cycle length will be displayed.

#### Major Street Direction

This combo box is only visible for sign-controlled intersections. This combo box specifies whether the major street through movements, which do not yield to any other movements, are located in either the East-West or North-South direction.

#### Grn

This input field is used to indicate the green interval duration. If the initial timing flag is on, user-coded green times will be ignored. If the initial timing flag is off, user-coded green times will be simulated explicitly unless interval times aren't consistent with the cycle length. If interval times are too low, the final-phase green time will be artificially increased. If interval times are too high, a fatal error message will be issued.

TRANSYT treats user-coded green times just like HCS does; they are considered average times. The only exception to this is when the "estimation" procedure is used, in which case the user-coded green times are treated as maximums.



If the user wishes to simulate an overlap phase as small as 1 second, this can be accomplished if the start-up lost times and extension of effective green times are correctly coded. On the surface, it is necessary to either specify a minimum 2-second overlap, or no overlap. Dual-ring coding of overlap phasing was discussed in more detail earlier on in this section. The Overlap Phasing section of the documentation provides more details on how to code single-ring overlap phasing.

### Yel

This input field is used to indicate the yellow interval duration. If in reality there is no yellow or all-red change interval, a dummy yellow interval must be coded nonetheless.

### Red

This input field is used to indicate the all-red interval duration. The user can optionally combine the yellow and all-red times into the yellow interval, and code an all-red interval duration of zero, because combining these interval durations into one should not have an effect on the modeling or the results. The maximum all-red value that can be coded is 5 seconds. If the user wishes to model a longer interval with no vehicles moving, this can be accomplished in the Timing View screen by coding a "dummy" phase with green time but no moving links.

### Phasing Diagram

The phasing diagram provides a graphical representation of the signal timing plan. On the phasing diagram, non-actuated phases are illustrated using dark blue arrows, whereas actuated phases are illustrated using light gray arrows. Protected movements are illustrated using arrows with solid lines, whereas permitted movements are illustrated using arrows with dashed lines. Right-hand driving is illustrated using arrows on the right-hand side, whereas left-hand driving is illustrated using arrows on the left-hand side.

By left-clicking on a phase inside the Phasing Diagram, the user can automatically update the "Phase #" data field. By right-clicking on a phase inside the Phasing Diagram, the user can easily clear (remove) all movements from that phase.

After the timing plan has been coded and saved, the record type data (1X, 18, 2Y, and 2Z) are updated in the background. The phasing diagram is limited to illustrating four approaches, however, so any active movements on approaches 5+ are not illustrated. In order to code complex timing plans involving more than three lane groups per approach, or more than four approaches, it is necessary to access the original Edit > Optional > Timing screen. From the original Timing screen, it is possible to specify any number of links moving during any given phase.

When coding a brand-new phasing sequence, certain left-turn and right-turn movements are drawn with dashed arrows and assumed to be permitted movements, based on the presence of opposing or conflicting thru movements. However, the user should pay close attention to the situation where left-turn and right-turn movements may or may not yield to each other. For example, suppose that a left-turn and an opposing right-turn move together in the same phase, but with no thru movements on either side. Depending on the number of receiving lanes, these movements could be either permitted, or protected. When a turning movement yields to an opposing turn movement, the user should open the Edit > Optional > Permitted screen, to code the appropriate opposing link number(s).

When importing a CORSIM input file into TRANSYT-7F, the phasing sequence is only imported for signals with a background cycle length (pre-timed or semi-actuated). For fully-actuated signals, maximum green times are imported for all 8 dual-ring phases, but dual-ring movements associated with each phase are not saved into the TIN file. As a result, CORSIM fully-actuated movements can only be displayed on the Dual-Ring Timing screen immediately after importing the TRF file. If the TIN file is then closed and opened at a later date, CORSIM fully-actuated movements will not be displayed on the Dual-

Ring Timing screen.

## **Bar Diagrams**

The bar diagrams provide a time-space representation of the signal timing plan, moving from left to right. The horizontal distance of a bar diagram represents the duration of the cycle length, and the bar is subdivided into smaller pieces that represent interval durations. The width of each interval bar is proportional to each interval duration, even though the width of the entire bar diagram always spans the entire Dual-Ring Timing screen. The bar diagram is also consistent with the phasing diagram, in that the initial intervals illustrated on the left side are associated with the first phase, and the final intervals on the right side are associated with the final phase. The upper bar diagram represents dual-ring phases 1 through 4. The lower bar diagram represents dual-ring phases 5 through 8.

The bar diagram uses the colors green, yellow, and red to indicate the green intervals, yellow intervals, and all-red intervals, respectively. Optionally, the yellow and all-red intervals can be illustrated using a "cross hatch" pattern, using the Options > Preferences screen. By default, the bar diagram is displayed using only the colors green, yellow, and red, to distinguish between the different signal intervals. The "cross hatch" option provides another mechanism besides color for distinguishing between the different signal intervals. When the cross hatch option is activated, yellow intervals are drawn using a single diagonal line as the fill pattern, in addition to the yellow coloring. All-red intervals are drawn using double diagonal lines as the fill pattern, in addition to the red coloring.

The TRANSYT model does not distinguish between yellow and all-red intervals. The software does require a minimum yellow time of 1 second, however. All-red intervals are optional and can be coded with a duration of zero seconds.

## **Satflow View Screen (associated with record types 28 and 292)**

This input screen was implemented to assist the user in determining adjusted saturation flow rates. As explained later, some adjustment factors are computed and displayed in exact accordance with Highway Capacity Manual (HCM2000) [12] procedures, whereas other adjustment factors are simply approximated. Detailed guidelines for analytical estimation of saturation flow rates are summarized in HCM chapter 16, page 11. If the user already knows the adjusted saturation flow rates, they can be coded directly on the Edit > Traffic screen, and this screen can be skipped.

Saturation flow rates are used by the macroscopic simulation model to affect the departure of queues from a link. Along with the volume, travel time between nodes, number of lanes, and timing plan, saturation flow rate is one of the most important factors in determining traffic network performance. Because of this, it is important that reasonable estimates of the saturation flow rate are provided by the user.

### Satflow View at University and 13th

Eastbound			Westbound			Northbound			Southbound	
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru
106	105	111	108	107	112	102	101	109	104	103
Base Saturation Flow Rate (vplphg)										
1900	1900		1900	1900		1900	1900		1900	1900
Number of Lanes										
1	1		1	1		1	1		1	1
Lane Width (ft)										
12	12		12	12		12	12		12	12
Heavy Vehicles (%)										
0	0		0	0		0	0		0	0
Grade (%)										
0	0		0	0		0	0		0	0
Parking Prohibited										
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Parking Maneuvers Per Hour										
0	0		0	0		0	0		0	0
Number of Buses Stopping										
0	0		0	0		0	0		0	0
Area Type										
Other <span style="border: 1px solid black; padding: 2px;">▼</span>										
Lane Utilization										
Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>	Egl <span style="border: 1px solid black; padding: 2px;">▼</span>
Type of Turning Movement										
LT <span style="border: 1px solid black; padding: 2px;">▼</span>	TH <span style="border: 1px solid black; padding: 2px;">▼</span>	RT <span style="border: 1px solid black; padding: 2px;">▼</span>	LT <span style="border: 1px solid black; padding: 2px;">▼</span>	TH <span style="border: 1px solid black; padding: 2px;">▼</span>	RT <span style="border: 1px solid black; padding: 2px;">▼</span>	LT <span style="border: 1px solid black; padding: 2px;">▼</span>	TH <span style="border: 1px solid black; padding: 2px;">▼</span>	RT <span style="border: 1px solid black; padding: 2px;">▼</span>	LT <span style="border: 1px solid black; padding: 2px;">▼</span>	TH <span style="border: 1px solid black; padding: 2px;">▼</span>
Ped-Bike influence on Left-Turns										
Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>
Ped-Bike influence on Right-Turns										
Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>	Noi <span style="border: 1px solid black; padding: 2px;">▼</span>

The Satflow View screen (Edit > Satflow) is illustrated above. This screen can be sent to the printer, or to an image file, by pressing Ctrl-P. When doing this, it is best to set the computer at a high screen resolution, so that all of the information can be simultaneously visible. Also note that a summary text report of saturation flow rate adjustment factors, called "Node Satflow Summary", can be accessed

from the Output Quick Jump. The Node Satflow Summary report only contains information if adjustment factors have been saved into the TRANSYT-7F input (\*.TIN) file (record type 292) by pressing the Apply button. Default settings for this screen can be specified by using the Saturation Flow Rate Calculator, which stores settings such as the lane width and percent grade into memory on the computer. The Saturation Flow Rate Calculator screen, which is described later in this chapter, can be accessed by selecting Edit > Optional > Capacities.

As shown at the bottom of the screen, the adjusted saturation flow rates (Calculated Value) are simply the product of the base saturation flow rate, the number of lanes, and eleven "adjustment factors". The Stored Values are the current values within the TIN file. Clicking on the Apply button will cause the calculated values from this screen to be saved into the TIN file, thus becoming the new Stored Values.

Techniques for field measurement of saturation flow rates are discussed on HCM chapter 16, page 158. Time permitting, it is always advisable to field-measure saturation flow rates at a few key locations in the network. Although the HCM recommends a "base" (unadjusted) saturation flow rate of 1900 vehicles per lane per hour of green (vplphg), this value is not likely to be realistic for all cities or city sections. If there is a consistent and predictable discrepancy between field-measured and HCM-estimated saturation flow rates, this might indicate that an adjustment to the base saturation flow rate is warranted. Once the user has calibrated the base saturation flow rate for a certain jurisdiction, the HCM analytical estimation procedures could then be used for adjusting the updated base saturation flow rate, at all other (non-field-measured) network locations.

In the Traffic Screen section of the documentation, rule-of-thumb (non-HCM) suggested values for adjusted saturation flow rates are available. The most efficient method of data entry might depend on the number of adjustments necessary. If conditions are fairly simple, adjusted saturation flow rates could be estimated using the general guidelines listed in the rule-of-thumb table. If numerous saturation flow adjustments are necessary for the protected movements, it might be easiest to use the Satflow View screen, or the HCS file import feature (which automatically imports the adjusted saturation flow rates).

This input screen does not estimate adjusted saturation flow rates for permitted-only links, or for secondary shared links, because any saturation flow rates coded for these links would be ignored by the model. For links representing permitted turning movements, the saturation flow rate is significantly lower, since these vehicles must wait for gaps in traffic on the opposing links. Saturation flow rate determination is accomplished automatically by the internal permitted movement model. The "maximum flow rate" (MFR) for periods when the permitted movement is unopposed may be adjusted (from the internal default value) on the Edit > Optional > Permitted screen. For protected-permitted turns, the saturation flow rate to be used during only the protected phase can be estimated on this screen. If multiple links are specified as being shared (Edit > Optional > Global > Shared Lanes screen, or record type 7), the screen only provides a value for the primary link, and this saturation flow rate applies collectively to all shared-lane links.

The following adjustment factor calculations are taken directly from the HCM:

$$f_w = 1 + \frac{(W - 12)}{30}, \text{ where } W \text{ is the lane width in feet}$$

$$f_{HV} = \frac{100}{100 + \%HV(E_T - 1)}, \text{ where } E_T \text{ equals 2.0 passenger cars per heavy vehicle}$$

$$f_g = 1 - \frac{\%G}{200}, \text{ where negative is downhill}$$

$$f_p = \frac{N - 0.1 - \frac{18N_m}{3600}}{N}, \text{ where } f_p \geq 0.05, \text{ and } f_p = 1.00 \text{ for no parking}$$

$$f_{bb} = \frac{N - \frac{14.4N_B}{3600}}{N}, \text{ where } f_{bb} \geq 0.05$$

$$f_a = 0.90 \text{ in CBD, } 1.00 \text{ in all other areas}$$

$$f_{LT} = 0.95 \text{ for exclusive left turn lanes, if right hand drive}$$

$$f_{RT} = 0.85 \text{ for exclusive right turn lanes, if right hand drive}$$

When implementing these adjustments, left-hand drive is handled differently than right-hand drive. For example, in the case of an exclusive right-turn lane under left-hand drive, the adjustment factor becomes 0.95, and is listed as the right-turn adjustment factor. Metric units are also handled differently for the lane width adjustment factor. Lane widths are accepted in meters, with decimal points allowed. The adjustment factor should be the same for equivalent English and metric lane widths.

The lane utilization adjustment factor is handled differently by the Satflow View screen, relative to the HCM. The combo box allows for selection of "equal", "slightly unequal", "moderately unequal", and "highly unequal" vehicle utilization of each lane. These selections result in adjustment factor values of 1.00, 0.95, 0.90, and 0.85, respectively. If a value of '1' is present in the Number of Lanes field, the lane utilization combo box is automatically disabled, and the lane utilization adjustment factor is assumed to be 1.00. If a value greater than '1' is present in the Number of Lanes field, and the user has not yet coded a lane utilization, the default lane utilization adjustment factor is assumed to be 0.95 ("slightly unequal").

The lane utilization adjustment factor computation from the HCM is as follows:

$$f_{LU} = \frac{v_g}{v_{g1}N}$$

where,

$v_g$  = unadjusted demand flow rate for the lane group, vph

$v_{g1}$  = unadjusted demand flow rate on the single lane in the lane group with the highest volume

$N$  = number of lanes in the lane group

The pedestrian-bicycle blockage adjustment factors for left-turns and right-turns are also handled differently by the Satflow View screen, relative to the HCM. The combo boxes allow for selection of "none", "light", "medium", and "heavy" pedestrian-bicycle blockage of the subject link. These selections result in adjustment factor values of 1.00, 0.95, 0.90, and 0.85, respectively. Through links should only be affected by pedestrian-bicycle blockage if they are primary links within a shared lane group, due to impedance of the turning vehicles. The HCM computation of the pedestrian-bicycle blockage adjustment factor is fairly detailed, and readers are referred to the HCM to review these details.

## Feeders Screen (associated with record type 28)

Input data on this screen are primarily applicable to internal links having an upstream intersection that may be affecting traffic flow patterns. The link connection information specified here affects simulation of platoon dispersion, as well as fuel consumption and travel time measurements. The information specified here also affects simulation of queue spillback, when step-wise simulation is used. The Feeders screen (Edit > Feeders) is illustrated below.

The screenshot shows the TRANSYT-7F software interface. The title bar reads 'TRANSYT-7F'. The menu bar includes File, Edit, View, Run, Window, Options, and Help. Below the menu bar is a toolbar with various icons. The main window title is 'FourNode.tin'. The active screen is titled 'Link Connections at University and 13th'. It displays a table of link connections for four directions: Eastbound, Westbound, Northbound, and Southbound. Each direction has three lanes: Left, Thru, and Right. Below the table, there are input fields for 'Upstream Node Number, or Feeding Link #1' and 'Feeding Link #2'. For each feeding link, there are input fields for 'Feed Volume (vph)' and 'Cruise Speed (mph)'.

Eastbound			Westbound			Northbound			Southbound	
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru
106	105	111	108	107	112	102	101	109	104	103

Upstream Node Number, or Feeding Link #1

0	0	0	207	207	207	0	0	0	303	303
---	---	---	-----	-----	-----	---	---	---	-----	-----

Feed Volume (vph)

0	0	0	50	300	50	0	0	0	50	300
---	---	---	----	-----	----	---	---	---	----	-----

Cruise Speed (mph)

0	0	0	35	35	35	0	0	0	35	35
---	---	---	----	----	----	---	---	---	----	----

Feeding Link #2

0	0	0	0	202	0	0	0	0	0	308
---	---	---	---	-----	---	---	---	---	---	-----

Feed Volume (vph)

0	0	0	0	50	0	0	0	0	0	50
---	---	---	---	----	---	---	---	---	---	----

Cruise Speed (mph)

Link connections can be drawn graphically on the Map View. When link connections are drawn graphically, data from the Feeders screen are coded automatically.

### Feeding Link #1-#4

Upstream feeding link numbers from upstream intersections are specified here. Press the Next Data or Prior Data pushbuttons in order to view and enter data for the other two feeding links, since the Feeders screen only displays information for two feeding links at a time. Changing the feeding link number to and from zero will disable and enable the associated fields Feed Volume and Cruise Speed.

### Feed Volume (vph or vpp)

The amount of volume passed on from the upstream feeding link to the subject link is specified here. Ideally the sum of the upstream feeding volumes and mid-block flows should be equal to the total link volume specified on the Edit > Traffic screen. The program issues a warning message if the flows are moderately unbalanced, and issues an error message if the flows are severely unbalanced.

An option is also available (through Options > Preferences) to specify volumes in units of vehicles per time period (vpp). Refer to the Preferences Screen section of the documentation for more details.

### Cruise Speed (mph or kmph)

The cruise speed specified here primarily affects the simulation of platoon dispersion, but fuel consumption and travel time measurements are also affected. It is important to note that the proper speed entries for simulation-only runs represent the actual average cruise speeds. The effects of delay due to the signal should not be included when estimating the input values. For optimization runs, the speed entries should be the desired speeds. The coded values for different feeding links will generally be the same, unless there is a significant difference in travel time of turning traffic compared to through traffic. If desired, the user can code unique approach speeds for external links in the Cruise Speed field under Feeding Link #1 (or record type 28 field 9).

For most practical purposes, indirect modeling of bus operations (through saturation flow rate adjustment) is recommended. However, a technique for simulating bus operations more explicitly is described below.

For exclusive bus links, speeds are coded as 'xxyy', where xx is the average bus cruise speed, and yy is the bus stop dwell time between 0 and 99 seconds. Average cruise travel time cannot be coded for buses. The proper dwell time value is the total stopped time at the bus stop. Deceleration and acceleration rates to and from a stop are automatically accounted for by TRANSYT, although these rates can be calibrated on the Model Coefficients screen (record type 8, model 4). This information is used to more accurately predict the arrival of buses at the intersection when separate bus links have been coded.

### Automatic Flow Balancing

This input screen features "automatic flow balancing". Simply type in the upstream node number in the Feeding Link #1 field, and the software will calculate the required flows (based on the link-numbering scheme) and fill in the remaining link connection information fields, including all feeding links, flows, and cruise speeds. Cruise speeds are automatically obtained from the TRANSYT-7F graphical user interface (GUI), within the Options > Cruise Speed dialog screen. When link connections are drawn graphically, the Feeders screen is called from the Map View, and automatic flow balancing is performed.

The automatic flow balancing logic produces the most common type of traffic flow pattern. It assumes that vehicles do not turn onto the side street (a.k.a. minor street, cross street) at the downstream intersection if they originated on the side street at the upstream intersection. If a significant number of vehicles turn onto the side street at the downstream intersection after originating on the upstream side street, feeding flows should be coded manually on the Feeders screen.

Note that for Feeding Links #2-#4, it is necessary to type in the entire upstream link number, instead of just the node number. However, typing this data may not be necessary, because the link number is automatically filled in after coding the upstream node number in the Feeding Link #1 field.

Feeding link flows may vary between time periods when performing multi-period simulation. The automatic flow balancing logic also supports multiple time periods. Within the GUI, there is always a certain "active" time period that is currently selected. The active time period can be changed by using the combo box at the top of the GUI, or by going to Options > Time Period > Select. The Feeders screen will always display the feeding link volumes for the active time period, and will always apply automatic flow

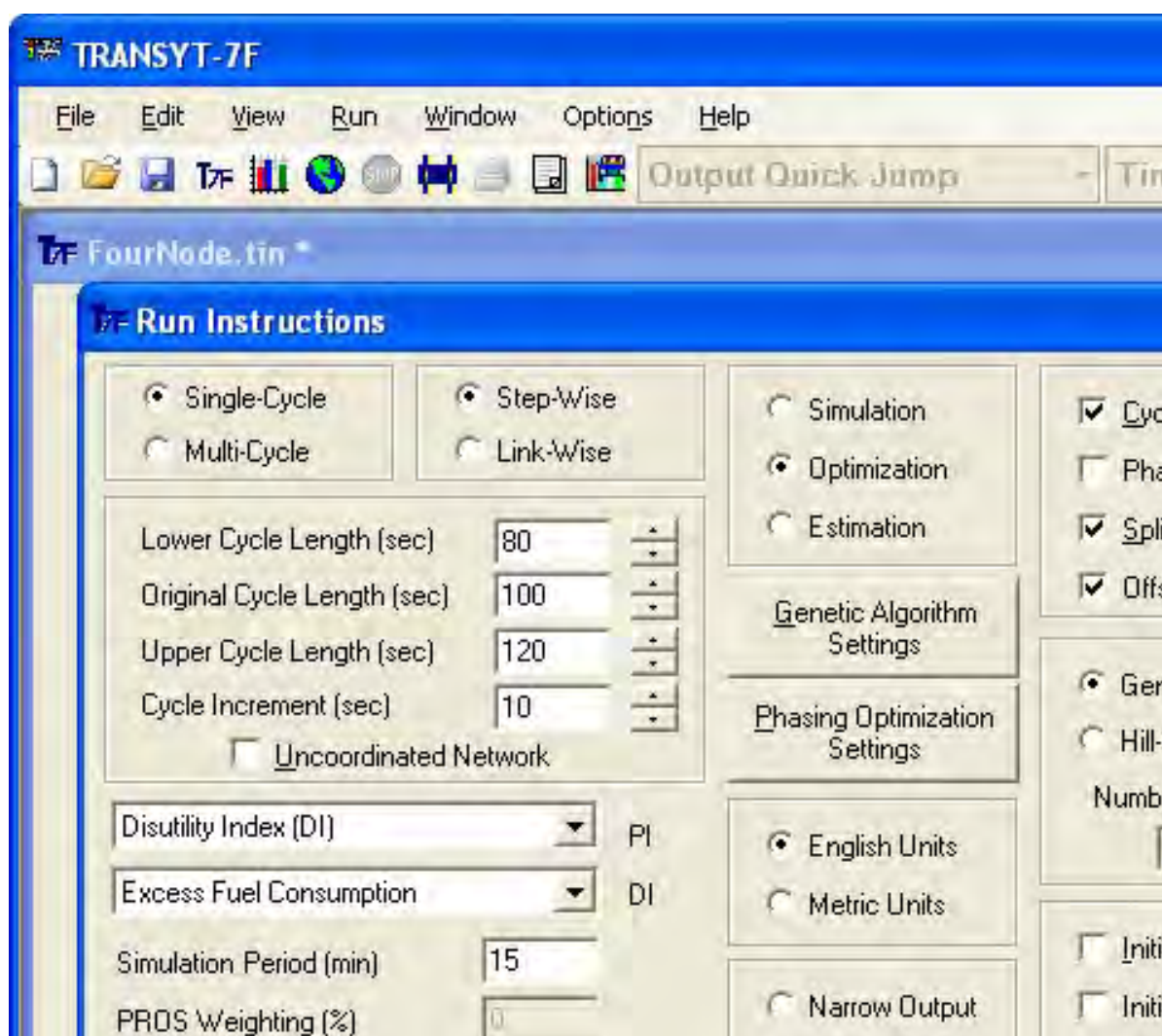


balancing for the active time period. When coding link connections in the Map View, this invokes automatic flow balancing for all time periods of a multi-period input file.

The automatic flow balancing methodology can handle mid-block source volumes as well. Refer to the Automatic Flow Balancing section of the documentation for more details.

## Analysis Screen (associated with record types 1, 1X, 10, 5X, and 9X)

The Analysis screen (Edit > Analysis) is illustrated below.



### Single-Cycle / Multi-Cycle (associated with record type 5X)

The duration of TRANSYT simulation is partially specified here. The duration of the analysis, in units of minutes, is also specified on the Analysis screen (record type 1). However, the program must be told whether or not to simulate only one cycle of traffic flow. The model is capable of using only one cycle of traffic patterns to estimate performance for the entire analysis period.



Multi-cycle simulation is primarily applicable to oversaturated conditions, where the extent of queue blockage is affected by the number of simulated cycles. When conditions are steady-state and completely undersaturated, results may be nearly identical under single-cycle and multi-cycle simulation, because the traffic flow patterns do not change between cycles.

Single-cycle simulation can be performed using either step-wise or link-wise simulation. Multi-cycle simulation can only be performed using step-wise simulation, so if multi-cycle simulation is selected, the option to select link-wise simulation will be automatically unselected. When multi-cycle simulation is selected, the number of simulated cycles is based on the coded Simulation Period.

#### Step-Wise / Link-Wise (associated with record type 5X)

The methodology of TRANSYT simulation is specified here. In link-wise simulation, each link is simulated to completion individually before switching to the next link. This is accomplished by simulating the protected-only external links first, then the permitted external links, and then steadily moving towards the middle of the network and simulating the internal links whose performance is affected by the outer links. In step-wise simulation, each link within the network is simulated for one step or second at a time before switching to the next step or second.

Link-wise simulation cannot be selected in conjunction with:

- multi-cycle simulation
- multi-period simulation
- actuated estimation

In general, step-wise simulation allows for more realistic simulation of queue blockage, back-of-queue movement, and intra-link travel time, whereas link-wise achieves faster program running times on the computer. Refer to the Simulation of Traffic Flow section of the documentation for more details.

#### Simulation / Optimization / Estimation (associated with record type 5X)

The question of whether to allow TRANSYT-7F to simulate or optimize the overall traffic signal timing plan is specified here. Traffic-actuated signal timing estimation is now available as well, although this cannot be selected in conjunction with multi-cycle and multi-period simulation.

When simulation is specified, the initial timing plan specified on the Edit > Timing screen (or record type 1X) is evaluated using a single simulation run. The exception to this is when the initial timing flag is turned on. When requesting simulation only, the upper cycle length will automatically be set equal to the lower cycle length.

TRANSYT-7F optimization is based on iterative simulation runs. The simulation run having the best Performance Index is remembered, such that the signal timing plan in effect at that time is recommended as optimal.

The selection of single-cycle vs. multi-cycle simulation, link-wise vs. step-wise simulation, and simulation vs. optimization, on the Analysis screen, produces corresponding codes in the TRANSYT-7F input data file. The codes are written onto record type 5X as follows:

'5X' (link-wise simulation) where X may be:

- 1 = single-cycle simulation
- 2 = single-cycle optimization

'-5X' (step-wise simulation) where X may be:

- 1 = single-cycle simulation
- 2 = single-cycle optimization
- 6 = multi-cycle simulation

7 = multi-cycle optimization

Cycle Length / Phasing Sequence / Splits / Offsets (associated with record type 5X)

The signal settings to be optimized are specified here. These settings are not applicable to simulation or estimation runs. If genetic algorithm optimization is used, the software can optimize any of these signal settings individually, and can optimize any combination of these signal settings simultaneously. However, the optimization process is often more efficient when one or more signal settings can be held constant. If hill-climb optimization is used, phasing sequence optimization is not permitted. Moreover, an offsets-only optimization run can only be accomplished by special coding techniques (refer to the section on Hill-Climb Optimization for instructions on this).

The signal settings to be optimized are stored on record type 5X, field 8, columns 36-39, as follows:

Column 36: optimize cycle length (0 = no, 1 = yes)

Column 37: optimize phasing sequence (0 = no, 1 = yes)

Column 38: optimize splits (0 = no, 1 = yes)

Column 39: optimize offsets (0 = no, 1 = yes)

For isolated intersections and uncoordinated intersections, offsets are not applicable and cannot be optimized. Phasing sequence optimization is primarily effective at improving progression for coordinated intersections. On the other hand, split and cycle length optimization is potentially beneficial at all intersections, isolated and coordinated.

Lower Cycle Length (sec) (associated with record type 1)

This input field indicates the one and only network cycle length when performing simulation-only runs, or when performing optimization runs constrained to a single cycle length. Frequently the network-wide cycle length is the same for every signalized intersection in the network; however, some signals may operate with a double cycle. If any signals are to be double cycled, all candidate cycle length values to be evaluated (in a cycle length optimization run) must be even numbered. When a multi-period analysis file is loaded and the run-type is simulation-only, the Lower Cycle Length can be coded uniquely among different time periods, although this is not recommended for most practical applications.

If the "Uncoordinated Operations" checkbox is switched on, all signalized intersections in the system are considered to be uncoordinated. Uncoordinated operations can be optimized if desired. When uncoordinated operations are in effect, the user-specified signal timing plans on the Edit > Timing screen (or record type 1X) indicate the local isolated cycle lengths. To simulate some intersections as coordinated and some intersections uncoordinated, it is best to specify coordinated operations for the network, and then indicate the uncoordinated intersections by using the "Coordinated Signal" checkbox (on the Edit > Timing screen) for those intersections.

This input parameter may also be coded on the Edit > Optional > Global > Run Controls screen.

Original Cycle Length (sec) (associated with record type 5X)

This field is applicable to genetic algorithm cycle length optimization runs. If cycle length optimization has not been selected, or if genetic algorithm optimization has not been requested, TRANSYT-7F ignores the value within this field.

When running traditional TRANSYT optimization, the Original Cycle Length affects the first individual in the first generation of optimization. The first individual contributes to the optimization process, and is also used to create "Initial" results for the output file. Typically the first individual in the first generation of optimization simply reflects the user-coded initial timing plan. However, if the initial timing flag is turned on, coordinated nodes on the optimization node list immediately adopt the Original

Cycle Length. If the user wishes to run genetic algorithm cycle length optimization without the initial timing flag selected, it is necessary to code an Original Cycle Length consistent with the coordinated nodes' individual cycle lengths (on the Edit > Timing screen). If any local coordinated cycle length exceeds the Original Cycle Length, there may be error messages related to the sum of interval durations. If the Original Cycle Length exceeds any local coordinated cycle length, extra green time may be added to the final phase of the timing plan. If the Original Cycle Length is close or identical to the global optimum cycle length, this could speed up or assist the optimization process. Refer to the section on Cycle Length Optimization (in Chapter 5) for more details.

When running grouped node optimization, the Original Cycle Length affects the "difference of offset", which must be maintained between grouped nodes. This would affect both traditional TRANSYT optimization and direct CORSIM optimization. Refer to the section on Grouped Nodes (in Chapter 5) for more details.

#### Upper Cycle Length (sec) (associated with record type 1)

This input field is only applicable to cycle length optimization runs. When the user wishes to constrain the analysis or optimization to a single cycle length only, the maximum cycle length will automatically be set equal to the minimum cycle length. If the "Uncoordinated Operations" checkbox is switched on, TRANSYT-7F ignores the value within this field.

This input parameter may also be coded on the Edit > Optional > Global > Run Controls screen.

#### Uncoordinated Network (associated with record type 1)

If the "Uncoordinated Network" checkbox is switched on, all signalized intersections in the system are marked as uncoordinated. All intersections are simulated with uniform flow profiles, because vehicle platoons are never expected to arrive at any specific window within the signal cycle. Uncoordinated operations can only be optimized using the genetic algorithm search technique. When uncoordinated operations are in effect, the user-specified signal timing plans on the Edit > Timing screen (or record types 1X and 18) indicate the local isolated cycle lengths. To simulate some intersections as coordinated and some intersections uncoordinated, it is best to specify coordinated operations for the network, and then indicate the uncoordinated intersections by using the "Coordinated Signal" checkbox (on the Edit > Timing screen) for those intersections.

This input parameter may also be coded on the Edit > Optional > Global > Run Controls screen. On the Global screen, if a negative minimum cycle length value is coded, the software assumes uncoordinated operations.

#### Cycle Length Increment (sec) (associated with record type 1)

If the minimum and maximum network cycle length fields are coded to indicate a range of cycle lengths to be evaluated, this field indicates the interval between candidate cycle lengths to be evaluated. Even if there is no cycle length range in effect, the cycle length increment value controls the cycle length "spinner" values on the Analysis screen. The cycle length increment should be evenly divisible into the range, otherwise the maximum cycle length will not be evaluated when running optimization. TRANSYT-7F ignores the value within this field if the minimum and maximum cycle length values are equal, or if the maximum network cycle length is coded as zero or blank. The cycle length increment is also ignored if the Uncoordinated Operations checkbox is turned on.

This input parameter may also be coded on the Edit > Optional > Global > Run Controls screen.

#### Genetic Algorithm Settings and Phasing Optimization Settings

Pushbuttons are available in the middle of the Analysis screen for specifying detailed Genetic Algorithm Settings and Phasing Optimization Settings. Refer to the Genetic Algorithm Optimization and

Phasing Sequence Optimization sections of the documentation for a description of these settings.

#### Genetic Algorithm / Hill-Climb (associated with record type 5X)

The selection between the available optimization algorithms is specified here. This input data is not applicable to simulation-only runs.

The optimization algorithm request is stored on record type 5X, field 8, column 40, as follows:

- 0 or blank -- Genetic Algorithm (elitist method)
- 1 -- Genetic Algorithm (non-elitist method)
- 2 -- Hill-Climb Algorithm
- 3 -- Traffic-Actuated Signal Timing Estimation

Selection of the genetic algorithm (instead of hill-climb) is required when optimizing:

- phasing sequence
- multi-period operations
- uncoordinated operations

#### Number of Generations (associated with record type 5X)

If genetic algorithm optimization has been requested, this input field specifies the maximum number of optimization generations. In general, a larger number of generations allows for more thorough optimization, but also requires longer program running times on the computer. Theoretically, every traffic network has a certain number of generations that would be needed to locate the global optimum solution. The number of generations needed to locate the global optimum solution would tend to become larger under the following conditions:

- large traffic networks
- multiple signal settings (cycle length, phasing sequence, splits, offsets) optimized simultaneously
- initial timing plan not close to the global optimum solution

The maximum number of generations can also be coded on the Edit > Analysis > Genetic Algorithm Settings screen. However, the maximum number of generations is provided on the Analysis screen itself, because it is the most commonly used and modified G.A. input parameter.

#### Performance Index (associated with record type 5X)

Also known as the objective function, the performance index (PI) allows the user to define their preferences regarding performance of the traffic network. Simply stated, TRANSYT-7F develops a signal timing plan that produces an optimal value of the PI. TRANSYT-7F offers numerous PI's to choose from, which can reflect anything that the user desires including delay, progression, stops, fuel consumption, queuing, and throughput. These PI options are illustrated below.

$$PI = \left[ \begin{array}{l} DI \text{ only} \\ PROS \text{ only} \\ PROS \& DI \\ PROS / DI \\ DI \cdot \sum_i \frac{\text{Average back of queue on link } i}{\text{Queuing capacity on link } i} \\ \text{Throughput only} \\ \text{Throughput \& DI} \\ \text{Throughput / DI} \end{array} \right]$$

In general, TRANSYT-7F always attempts to maximize the PI, unless the disutility index (DI) has been selected as the PI. Since the DI can be a combination of delay, stops, queuing, and fuel consumption, this value must be minimized. Other objective functions that involve progression or throughput must be maximized.

The DI is a combination of vehicle delay, stops, and fuel consumption. Weighting factors are available in order to place more emphasis on any of these three DI components if desired. Delay-only optimization results in excessive stops and fuel consumption. Excess fuel consumption (minimization) is considered to be a good compromise between bandwidth and delay-based optimization. Although the DI has been shown over the years to be a practical and effective combination of minimizing delay and stops, it does not necessarily produce a wide bandwidth.

Progression opportunities (PROS) represent the ability of vehicles to progress through multiple intersections without stopping. Points are scored every time a vehicle is able to traverse additional intersections. Therefore it is possible to achieve PROS without having a wide bandwidth stretching from the beginning of an arterial street to the end. In order to use this particular PI, it is necessary to define a route or list of links (on the Map View, or the Edit > Optional > Route screen) on which progression is desired.

It is also possible to select a combination of PROS and DI in order to achieve a compromise between the two objectives. For example, if the PI is defined as PROS/DI, the program attempts to simultaneously maximize the numerator (PROS) and minimize the denominator (DI). For optimizing bandwidth and progression speeds, PROS/DI is recommended. PROS-only is rarely acceptable, because although it often produces a good bandwidth, the minor movements are allowed to fail. Refer to the PROS Optimization section of the documentation for more details.

Except for multi-period optimization of the DI, Throughput and Queuing Ratio are the sometimes most appropriate PI's for standard (single-period) optimization of severely oversaturated conditions. Throughput, which is analogous to the number of vehicles discharged, would not be an effective PI for undersaturated conditions, in which the number of vehicles discharged is always equal to the input volume demand. For more details, refer to the section entitled "Optimization Under Congested Conditions".

Note that performance index options PROS & DI and Throughput & DI were designed for use with hill-climb optimization. As such, these performance index options are not recommended when using genetic algorithm optimization.

#### Disutility Index (associated with record type 5X)

The exact disutility index definition is specified here. In general the selection of (minimizing) excess fuel consumption indicates a compromise between minimizing vehicle delay and maximizing

bandwidth. Refer to the Disutility Index section of the documentation for more details.

PROS Weighting (%) (associated with record type 5X)

This input field indicates the relative weighting of PROS (on the designated arterial streets) versus the DI. This field is applicable when PROS/DI has been selected as the PI. The range for this value is from 1% to 1000%. If a value of 0% is coded, the default value of 100% is applied.

Simulation Period (min) (associated with record type 1)

This is the period of time (in units of minutes) for which network traffic conditions are assumed to remain constant. This is particularly critical for oversaturated conditions. The value coded should represent the length of time in which oversaturation (represented by the input volume) actually exists. When a multi-period analysis file is loaded, the Simulation Period can be coded uniquely among different time periods, although this is not recommended for most practical applications.

The period length specified here is always used directly in the calculation of random delay, d2. The period length is only used as the duration of simulation if multi-cycle step-wise simulation (or optimization) has been requested.

This input parameter may also be coded on the Edit > Optional > Global > Run Controls screen.

Initialization Period (sec) (associated with record type 5X)

The Simulation Initialization Period Length is used to initialize traffic flow patterns, flow profiles, and queue lengths. The default simulation initialization period length is equal to the network background cycle length, even when coded as zero. For most practical applications, it is recommended that the Initialization Period Length be left alone, or set equal to the background cycle length. On the Analysis screen, this input field is only enabled for simulation-only runs.

Maximum v/c Ratio (associated with record type 5X)

When throughput-only is selected as the PI, this field indicates the maximum volume-to-capacity ratio which, when exceeded on any given link, applies a penalty to the PI.

v/c Ratio Penalty (associated with record type 5X)

When throughput-only is selected as the PI, this field indicates the penalty to the PI whenever the maximum v/c ratio is exceeded on any given link.

English Units / Metric Units (associated with record type 1)

The choice between English units and metric units is specified here. This choice governs certain pieces of input data (e.g., cruise speed mph / kmph) and certain pieces of output data (e.g., fuel consumption gallons / liters).

This input parameter may also be coded on the Edit > Optional > Global screen.

Narrow Output / Wide Output (associated with record type 1)

Appearance of the TRANSYT legacy report or output file is specified here. The narrow output format is 80 columns wide and is better suited to printing, whereas the wide output format is 132 columns wide and provides more useful information. This relates to the output file generated by the macroscopic simulation engine (T7FWin2.exe).

This input parameter also affects the appearance and content of the "Node Output Summary" text

report. The summary reports are always 80 columns wide, but when "Wide Output" is selected, this produces the "detailed" node summary report. When "Narrow Output" is selected, this produces the "short" node summary report. Node summary reports display results for the "standard" twelve intersection movements (i.e., left, through, and right for NB, SB, EB, and WB). When intersections have complex geometry, or when performing a lane-by-lane analysis, be aware that overall intersection results (even on the node summary report) may account for additional movements or approaches. When the user needs to view movement-specific results for the additional movements or approaches, it is necessary to obtain these results from the legacy report.

#### Initial Timing Flag (associated with record type 1)

If neither of the initial timing flags are selected, this indicates that the user will provide offset (or yield point) values, as well as the durations of all fixed and variable intervals on the Edit > Timing screen (or record types 1X and 18). Initial timing flag 1 indicates that the user will provide the durations of all fixed intervals on the Timing screen (record types 1X and 18). Variable green interval durations will be provided by the program according to a model that equalizes the degree of saturation for all critical links.

Note that "variable" intervals are green intervals that can be optimized. On the other hand, "fixed" intervals are not subject to optimization. Fixed intervals include yellow, all-red, and occasionally green (e.g., flashing don't walk) intervals. Refer to the Edit > Timing screen documentation for additional details.

When using actuated control, initial timing flag 1 assigns green time to the actuated phases to achieve a target degree of saturation. All unused green time is then donated to the major street non-actuated phase. The target degree of saturation model is effective for optimization runs, and facilitates location of the optimal actuated phase times. To ensure that the optimized actuated phase times materialize in the field, it is necessary to set the force-offs and maximum greens appropriately. Refer to the Traffic-Actuated Control section of the documentation for more details.

When using the actuated estimation model, initial timing flag 1 will cause force-offs or maximum greens to be estimated prior to the analysis. This is not necessarily recommended. The estimation model is most appropriate for simulation runs, and evaluation of existing force-offs and maximum greens. The estimation model computes traffic-actuated phase times that are likely to materialize in the field, based on the user-specified maximum green and force-off settings.

Initial timing flag 2 is the same as initial timing flag 1, except the user-specified offsets and yield points are retained. For optimization runs, initial timing flag 1 eliminates the need to specify offsets (or yield points) and variable green interval durations at signals to be optimized. This information must be provided, however, for signals that will not be optimized (i.e., nodes not listed on the optimization node list, or "grouped" nodes at which splits are fixed). The initial timing model (initial timing flag 1) is automatically applied by the program whenever cycle length optimization is performed. Initial timing flag 2 is automatically applied by the program whenever phasing sequence optimization is performed without cycle length or split optimization.

When a multi-period simulation file is loaded, the initial timing flags can be coded uniquely among different time periods by using the record type editor. Although the initial timing flag can be used for multi-period simulation runs, it cannot be used for multi-period optimization runs. Thus, if multi-period cycle length or phasing sequence optimization is requested, optimization of splits becomes mandatory. Also note that the initial timing flag cannot be used to generate initial timings for uncoordinated signals.

Initial timing flags may also be coded on the Edit > Optional > Global > Run Controls screen. For more information, refer to the documentation on the Initial Timing Model.

#### Ignore Actuated Flags (associated with record type 10)

When this check box is selected, actuated flags in the data file are ignored, meaning that phase durations and vehicle delays will be computed under the assumption of pre-timed control. This input can

be useful for quickly comparing actuated vs. pre-timed operation, without having to add or delete the actuated flags throughout the data file. This check box affects the input value that is written to the Network-Wide Target Degree of Saturation for Actuated Controllers field (Edit > Optional > Global > Network Parameters screen or record type 10).

Run Info (associated with the Run Title Record)

The Run Title (and other analysis-specific information) can be optionally specified here, for inclusion within the TIN file, and within output reports.

## Global Screen (associated with record types 1-10)

**Run Controls** (associated with record type 1)

Minimum Network Cycle Length

This input field indicates the one and only network cycle length when performing simulation-only runs, or when performing optimization runs constrained to a single cycle length. If a positive value is coded within this field, the program assumes that the network-wide cycle length is the same for every coordinated intersection in the network; however, some signals may operate with a double cycle. If any signals are to be double cycled, all candidate cycle length values to be evaluated (in a cycle length optimization run) must be even numbered.

If a negative value is coded within this field, all signalized intersections in the system are considered to be uncoordinated. When uncoordinated, the user-specified signal timing plans on the Edit > Timing screen (or record types 1X and 18) indicate the local isolated cycle lengths.

Maximum Network Cycle Length

This input field is applicable to cycle length optimization runs. When the user wishes to constrain the analysis or optimization to a single cycle length only, the maximum cycle length coded should be equal to the minimum cycle length coded, or left blank. If the minimum network cycle length is coded as negative to indicate uncoordinated operations, TRANSYT-7F ignores the value within this field.

Cycle Length Increment

If the minimum and maximum network cycle length fields are coded to indicate a range of cycle lengths to be evaluated, this input field indicates the interval between candidate cycle lengths to be evaluated. The cycle length increment should be evenly divisible into the range, otherwise the maximum cycle length will not be evaluated. TRANSYT-7F ignores the value within this field if the minimum and maximum cycle length values are equal, or if the maximum network cycle length is coded as zero or blank. The cycle length increment is also ignored if the minimum network cycle length is coded as negative to indicate uncoordinated operations.

Left-Hand Drive Flag

This flag primarily affects phasing sequence optimization, when it is necessary to determine whether leading or lagging left-turns or right-turns will be examined. However the flag may have also affected brand-new data sets created within the TRANSYT-7F GUI. The initial assumption for brand-new data sets is that through and right-turn movements will occupy a shared lane, assuming right-hand driving. If the left-hand drive flag is turned on, then brand-new data sets will specify that through and left-turn movements occupy shared lanes. In addition, this flag controls appearance and behavior of the graphical timing plan entry (Timing View) screen.



A value of '1' in this field indicates left-hand driving, whereas any other value indicates right-hand driving. The value in this field is automatically generated by check boxes inside the New File and Phasing Sequence Optimization screens.

### Simulation Stepsize

This input field indicates the seconds-per-step resolution to be used by the simulation model. TRANSYT produces more accurate measures of effectiveness when a small seconds-per-step resolution is used. This setting is only applicable to link-wise simulation, and the only advantage of the larger simulation stepsize is faster program running times on the computer. When using step-wise simulation, the simulation stepsize must be 1 second. If another stepsize besides 1 second is coded in conjunction with step-wise simulation, the software may automatically use a stepsize of 1 second, internally.

### Network-Wide Startup Lost Time

This is the time lost at the beginning of green while drivers react and vehicles begin to move. This value may be calibrated for specific links on the Edit > Traffic screen (or record type 29). The value in this field may be a "real number" coded as an integer. In this case the program divides the number by 10 until the result is less than 10. For example, to code a network-wide startup lost time of 2.75 seconds, code '275' here.

### Network-Wide Extension of Effective Green

This is the time from the end of green until the last vehicle (on average) crosses stop line. This value may also be calibrated for specific links on the Edit > Traffic screen (or record type 29). The value in this field may be a "real number" coded as an integer. In this case the program divides the number by 10 until the result is less than 10. For example, to code a network-wide extension of effective green time of 3.2 seconds, code '32' here.

### Network-Wide Stop Penalty

This is the factor, "K", which is multiplied by the total number of stops per second in the calculation of the disutility index ( $DI = \text{delay} + "K" * \text{stops}$ ). The stop penalty may be changed for specific links on the Edit > Optional > Weighting screen (or record type 38).

If this field is left blank, coded as zero, or coded as '-1', then TRANSYT automatically determines the stops and delay coefficients to minimize excess fuel consumption. If the coded value is '-2', then TRANSYT will produce a signal timing plan that minimizes network-wide vehicle delay without considering stops. This will generally result in more green time being assigned to the minor street than would be otherwise if stops were also considered.

### Output Level Flag

In addition to the narrow vs. wide output specification on the Edit > Analysis screen, this field controls the appearance and content of the TRANSYT legacy report, or output file. Regarding simulation-only runs, a coded value of '0' indicates that the signal timing tables will be suppressed, whereas a '1' causes the signal timing tables to be printed.

Regarding hill-climb optimization runs (without cycle length optimization), the final performance table indicating performance under the optimal timing plan is always included. A coded value of '0' indicates that signal timing tables will be suppressed, and the initial performance table (output under the initial timing plan) will be suppressed. A coded value of '1' allows the signal timing tables to be included, and a coded value of '2' allows the initial performance table to be included.

This input does not affect the appearance of genetic algorithm optimization reports, cycle length optimization reports, actuated estimation reports, or summary text reports.

### Initial Signal Timing Flag

A coded value of '0' indicates that the user will provide offset (or yield point) values, as well as the durations of all fixed and variable intervals on the Edit > Timing screen (or record types 1X and 18). A coded value of '1' indicates that the user will provide the durations of all fixed intervals on the Timing screen (record types 1X and 18). Variable green interval durations will be provided by the program according to a model that equalizes the degree of saturation for all critical links.

Note that "variable" intervals are green intervals that can be optimized. On the other hand, "fixed" intervals are not subject to optimization. Fixed intervals include yellow, all-red, and occasionally green (e.g., flashing don't walk) intervals. Refer to the Edit > Timing screen documentation for additional details.

When using traffic-actuated control, option '1' assigns green time to the actuated phases to achieve a target degree of saturation. All unused green time is donated to the major street non-actuated phase. The target degree of saturation model is effective for optimization runs, and facilitates location of the optimal actuated phase times.

When using the actuated estimation model, option '1' will cause force-offs or maximum greens to be estimated prior to the analysis. This is not necessarily recommended. The actuated estimation model is most appropriate for simulation runs, and evaluation of existing force-offs and maximum greens. The actuated estimation model computes traffic-actuated phase times that are likely to materialize in the field, based on the user-specified maximum green and force-off settings.

A coded value of '2' is the same as '1', except the user-specified offsets and yield points are retained.

For optimization runs, a coded value of '1' eliminates the need to specify offset or yield point values and variable green interval durations at signals to be optimized. This information must be provided, however, for signals that will not be optimized (i.e., nodes not listed on the optimization node list or "grouped" nodes at which splits are fixed). The initial timing model (option '1') is automatically applied by the program whenever cycle length optimization is performed (except in the case of simultaneous genetic algorithm optimization of cycle length and splits). The initial timing model (option '2') is automatically applied by the program whenever phasing sequence optimization is performed without cycle length optimization (except in the case of simultaneous genetic algorithm optimization of phasing sequence and splits).

Also note that the initial timing flag cannot be used to generate initial timings for uncoordinated signals, and cannot be used for multi-period optimization.

The initial timing flag may also be coded on the Edit > Analysis screen. If the initial timing flag is set on the Analysis screen, the corresponding value will be written onto record type 1, and displayed the next time the Global screen is loaded.

### Analysis Period Length

This is the period of time (in units of minutes) for which network traffic conditions are assumed to remain constant. This is particularly critical for oversaturated conditions. The value coded should represent the length of time in which oversaturation (represented by the input volume) actually exists.

The period length specified here is always used directly in the calculation of random delay, d2. The period length specified here is only used as the duration of simulation if multi-cycle step-wise simulation (or optimization) has been requested on the Edit > Analysis screen (or record type 5X).

If this value is coded as negative, traffic volumes from the Traffic and Feeders screens (or record types 28 and 29) are interpreted in units of vehicles per time period (vpp), instead of vehicles per hour

(vph). In this case, the time period duration is taken as the absolute value of the coded number.

#### Signal Timing Units Flag

0: Offsets (or yield points) and interval durations are specified on the Edit > Timing screen (or record types 1X and 18) in units of seconds.

1: Offsets (or yield points) and interval durations are specified on the Edit > Timing screen (or record types 1X and 18) in units of percent of cycle. This option is not fully supported in the current version of the software.

#### Speed/Travel Time Flag

0: Average cruise speeds are specified on the Edit > Feeders screen (or record type 28) in units of miles per hour (mph) or kilometers per hour (kmph).

1: Average cruise speeds are specified on the Edit > Feeders screen (or record type 28) in units of seconds. This option is not fully supported in the current version of the software.

#### English/Metric Units Flag

0: Link lengths and average cruise speeds are coded in U.S. customary (English) units (e.g., ft and mph).

1: Link lengths and average cruise speeds are coded in SI (metric) units (e.g., m and kmph).

#### Analysis Type

After opening a CORSIM input (\*.TRF) file, a corresponding TRANSYT-7F input (\*.TIN) file is generated, which contains the processing and optimization settings. By default, a value of '4' is written into the Analysis Type field, to indicate that "direct" CORSIM processing or optimization is being performed. If the user selects File > Convert to Macroscopic Analysis, minor modifications are made to the interface and the TIN file, to support a subsequent macroscopic analysis. In this event, the Analysis Type is changed to '3', to indicate that the user can later select File > Convert to Microscopic Analysis. For more information on the conversion between macroscopic and microscopic analysis, refer to the CORSIM Processing Methodology section of the documentation.

When the Analysis Type is '4', this enables the ability to export signal timing plans from TRANSYT (TIN) to CORSIM (TRF). Timing plans are exported by selecting File > Export Network-Wide Timing to CORSIM, or by clicking the Export button for a given node in the Timing View screen.

If a traditional TRANSYT analysis (with no linkage to CORSIM) is being performed, this field should be coded as '0', or left blank. This prevents the user from exporting timing from an old TRANSYT input file to a CORSIM input file, which is dangerous because there are usually inconsistencies between the phasing sequences and reference intervals. When a TRANSYT input file is automatically generated by loading a CORSIM input file, the phasing sequences and reference intervals are more likely to match each other. For an older TRANSYT input file, the user can manually create a linkage to CORSIM by coding an Analysis Type of '3' or '4', but this is not recommended.

#### **Optimization Node List** (associated with record type 2)

This list specifies the node numbers for which signal timing is to be optimized during an optimization run. It is recommended that node numbers have a maximum of two digits. Unsignalized intersections may be omitted from the list to save coding time, but including these nodes will have no effect on results. The screen allows for specification of 15 node numbers, but additional node numbers may be specified by pressing the "Duplicate RT" pushbutton to generate a duplicate record. Blank fields are ignored.

If the run is strictly a simulation run, this node list may be omitted, unless the initial timing model has been requested, in which case the node list is needed to indicate nodes at which initial splits must be

calculated. For simulation-only runs where the initial timing model has not been requested, the node list will be ignored. The initial timing model is requested on the Global > Run Controls screen (or record type 1 field 11).

If two or more intersections are closely spaced such that a fixed relationship between them is desired, the signal timing of these nodes can be held constant during hill-climb optimization. If two or more nodes are grouped, the secondary node number(s) must be negative and must immediately precede the primary node number (which has a positive number) in the list. For example, if the list contains 2 -6 8 10, node 6 will be grouped with node 8 (note: do not refer to grouped nodes with negative numbers elsewhere in the data file). The offset difference between the grouped nodes will be held constant during the optimization run. Offsets and phase lengths for the nodes surrounding the group will be optimized normally. In a cycle length optimization run, all nodes should be included on the list and grouped nodes are ignored.

When using genetic algorithm optimization, currently the only way to maintain grouped nodes is to omit these nodes from the optimization node list, or to switch off the "Allow Offset Optimization" checkbox (on the Edit > Timing screen) for the nodes in question.

The optimization node list also specifies the sequence in which signals will be optimized during the hill-climb optimization process. The sequence may not have an effect on genetic algorithm optimization. Nodes may be repeated on the node list to force them to be optimized more than once. This may reduce the chance of particular nodes being trapped in a local optimum solution. This limits the total list of nodes to 200, although only 99 may actually be analyzed.

### **Link Numbering Scheme** (associated with record type 3)

This screen allows specification of a customized link numbering scheme, which primarily affects appearance of the program's user interface and printed report only. Up to 12 link numbers will be reserved for the 12 basic labels of northbound, southbound, eastbound and westbound left, through and right. Input data screens such as Traffic, Feeders, Permitted, Queuing and Actuated automatically display the appropriate link numbers on the interface, next to the movement direction labels, for better understanding. The record type editor also uses the link numbering scheme to display movement labels that correspond with specific link numbers.

The canned, predefined link numbering schemes (TRANSYT-7F Standard and NEMA) may be requested using a single input value on the Global > Network Parameters screen (or record type 10) instead of having to specify the 12 entries on this screen. The TRANSYT-7F Standard scheme is applied by default if the user has not specified any link numbering scheme within the data file. The link numbering scheme itself has no direct effect on results. Results for additional links beyond the initial 12 are reported in the output file according to their link number only.

According to the TRANSYT-7F Standard scheme, the number '1' is reserved for the northbound through movement. Thus link number 401 would indicate the northbound through movement at intersection #4. The link numbering scheme also illustrates why node numbers should be two digits or fewer to allow the program to operate successfully. For example, permitted link number 55 at node number 77 would have to be coded as a five digit number (-7755) on the Edit > Optional > Timing > Phase #1-#7 screen, and the record type format supports a maximum of five digits per field.

According to all of the predefined schemes (TRANSYT-7F Standard, and all four NEMA orientations), numbers 1 through 12 are reserved for the basic 12 turning movements, such that link number 413 would indicate an "extra" link at node #4. In this case, the extra link could be used to model an extra approach, at an intersection having more than four approaches, or could be used to model an unusual lane configuration on one of the original four approaches.

Many aspects of the TRANSYT-7F GUI assume that numbers 1 through 12 represent the basic 12 turning movements, and that numbers 13 and higher are used for either lane-by-lane analysis, or for

minor "diagonal" approach numbers five and higher. Because of this, it is strongly recommended that any movement numbers coded here fall within the range of 1 through 12.

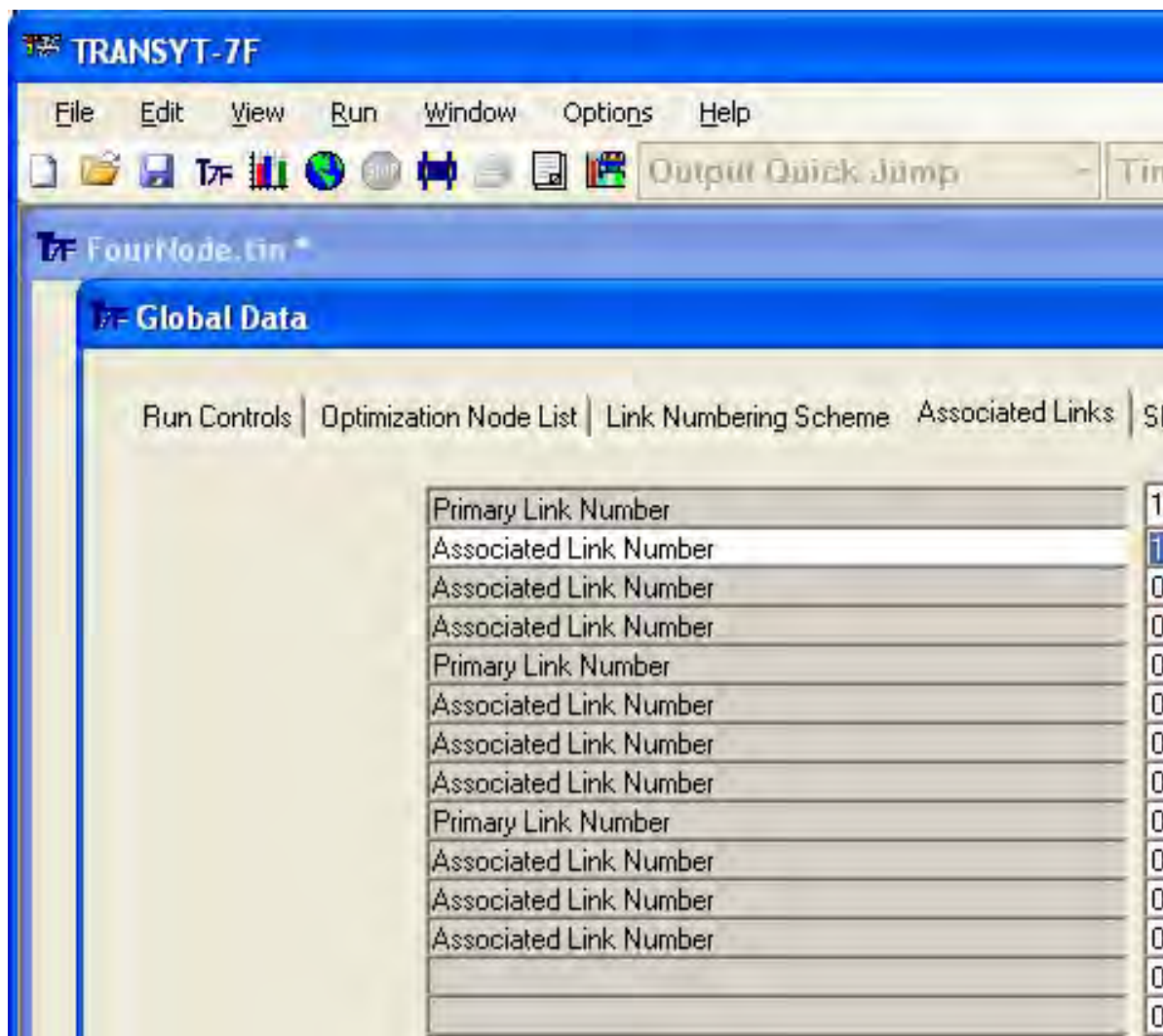
### **Associated Links** (associated with record type 5)

Aside from phasing sequence optimization, this input data is not used or required by TRANSYT-7F. Associated links must only be coded when optimizing the phasing sequence for complex geometric conditions. Similar to a secondary shared stopline link, an associated link moves with its primary link. For example, assuming right-hand drive, a protected right-turn from an exclusive lane may be associated with protected left-turns on the adjacent approach.

For simple data sets, it is not necessary for the user to code associated links. The program should automatically handle any links corresponding with the link numbering scheme. Again assuming right-hand drive, right-turn links are assumed to be associated with the neighboring through movement, and protected right-turns from an exclusive lane are assumed to be associated with protected left-turns on the adjacent approach.

However, TRANSYT allows for a large number of links to be coded at one intersection. When reconstructing the phasing sequence, the program may not know where certain links belong in the signal timing plan. This is particularly true of links not included on the link numbering scheme. For example, if links 1 through 12 are defined by the link numbering scheme, and link 113 has been coded by the user, it may be necessary to code this as an associated link prior to phasing sequence optimization. The screen image below illustrates an association of link 113 with link 101.

If phasing sequence optimization has been requested on the Edit > Analysis screen, "extra" link numbers 13 and higher must be declared as associated links at all nodes, including those at which phasing sequence optimization has been disallowed (Edit > Analysis > Phasing Optimization Settings). If associated links are not coded properly, or not coded when they are needed, the optimization run might abort and issue some error messages (e.g., "Link 113 not found in network link list"). Another way to avoid error messages would be to omit intersections from the optimization node list. Intersections with five approaches (or more) can have their phasing sequence optimized without the need for associated links, as long as the movements from approach #5 (and higher) move in their own exclusive phase.



During genetic algorithm optimization, if the program issues an error message that says "Relevant data could not be extracted from TRANSYT output", the first thing to do is simply run the original data set without phasing sequence optimization. If the original data set runs fine and contains no errors, this may indicate a problem with reconstructing the phasing sequence and timing plan. This could be caused by not coding an associated link, or by leaving the "Right-Hand Driving" checkbox toggled on when the data file simulates left-hand driving, etc.

If a link has been coded as a secondary shared stopline link (Global > Shared Lanes or record type 7), it should not be necessary to code this again as an associated link. The list of permitted shared lane groups is scanned prior to optimization, and secondary shared links are automatically "associated" with their primary links. For example, if links 1 through 12 are defined by the link numbering scheme, and link 113 has been coded by the user as a secondary shared link, it is not necessary to code this as an associated link prior to phasing sequence optimization.

Note that the Global > Run Controls screen (or record type 1 field 5) can be used to indicate left-hand driving. Otherwise, right-hand driving is assumed. Alternatively, the user could toggle the "Right-Hand Driving" checkbox on the Edit > Analysis > Phasing Sequence Optimization screen. This input data is only used by TRANSYT-7F when running phasing sequence optimization, or when using the Timing View screen.

### **Shared Lanes** (associated with record type 7)

The primary link in a group of shared-lane links is typically specified as the through movement link. Left-turn and right-turn movements are typically considered to be secondary links. Only primary links can have bandwidth constraints, or PROS analysis, or be listed on the Edit > Optional > Route screen (record types 41, 42, and 61). Any secondary link is expected to move in the same phase(s) as its associated primary link. A secondary link does not need to have a saturation flow rate specified on the Edit > Traffic screen (record type 28), but the link length must be specified to achieve proper simulation of platoon dispersion and queue spillback.

Input data from this screen are not allowed for modeling unsignalized intersections. Refer to the Sign Control Examples section of the documentation for more details.

For more details on the modeling of shared lanes, including permitted shared lanes, refer to the Shared Lanes Methodology section of the documentation.

### **Model Coefficients** (associated with record type 8)

This screen enables users to change default values and behavior of the following models:

- fuel consumption model
- total operating costs model
- stops reduction model
- permitted movement models
- platoon dispersion model
- upstream filtering and metering factor (I-factor) model
- inclusion of random stops within the disutility index (DI)
- shared lanes model

The table below describes the coding to override the global permitted movement model, and to calibrate other miscellaneous models. For additional instructions on how to calibrate the major models listed above, please refer to those specific documentation sections.

The coding of this data must properly reflect the level of numerical precision ("Prec"). For example, in order to code RATDEL as 1.3 based on E4 precision, it would be necessary to code "13000". Moreover, negative coded values are interpreted as negative. For example, a parameter value of 0.012345 at E5 would be coded as "-1234" or "-1235", depending on user preference. If this screen is used to calibrate a model, it is necessary to code all of the fields associated with that model, with the exception of Penic fuel consumption model [32] number '11'. Any parameter not coded will be set to zero.

Note that "Fld 2" in the table below (for record type 8) is associated with the first field of the Global > Model Coefficients screen.

Fld	Variable	Description	Prec	Range	Default
<b>2</b>	<b>'4'</b>	<b>Miscellaneous Model Parameters</b>			
3	IPERMD	network-wide permitted movement model number*	E0	0-50	0
4	BETA	parameter "beta" within the platoon dispersion model	E5	1-99999	0.8 (80000)
5	ACCDEC(1)	average acceleration rate for general traffic (fpss)	E3	1-99999	5 (5000)
6	ACCDEC(2)	average deceleration rate for general traffic (fpss)	E3	1-99999	10 (10000)
7	ACCDEC(3)	average acceleration rate for general traffic (fpss)	E3	1-99999	5 (5000)
8	ACCDEC(4)	average deceleration rate for general traffic (fpss)	E3	1-99999	10 (10000)
9	RATDEL	conversion factor between total delay and stop delay (total delay is divided by this factor)	E4	1-99999	1.3 (13000)
* if '10' is coded, this automatically selects FHWA/TRC models 11-16. Models 17-19 cannot be chosen as global default models.					

In TRANSYT, the default permitted movement model for signalized intersections is one developed by the Australian Road Research Board (ARRB) [27]. For the overall network, the user may override this model and choose among *most* of the available permitted movement models as the new global default model. The desired global permitted movement models can be selected on the Global > Model Coefficients screen (or record type 8), using the "Global Number" in the table below.

In this table below, model numbers 17, 18, and 19 are designed for minor stop-controlled movements, and cannot be selected as the global model. In addition, the table below assumes right-hand drive (e.g., "RTOR" to indicate the right-turns-on-red model), such that the "Permitted-only left-turn" model could be chosen to model permitted-only right-turns in a country with left-hand drive.

Model Type	Speed (mph):	1 Lane		2+ Lanes		Global No.
		<40	>=40	<40	>=40	
ARRB Model			0			0
FHWA/TRC Model (all)						10
Permitted-only left-turn		11	11	12	13	
Protected + permitted left-turn		14	14	15	16	
Unsignalized left-turn (artery)		17	17	17	17	
RTOR & unsig. X-street thru		18	18	18	18	
Unsignalized X-street left-turn		19	19	19	19	
Texas A&M Model						
Field study parameters (all)			21			21
Simulation parameters (all)			22			22
Minnesota Model		31		32		30
Ohio State Model		41		42		40

Although the disutility index (DI) has been updated to include random stops, the release 8.2 (and earlier) DI that includes uniform stops only can be accessed by coding a '6' in the first field (Model Number). Refer to the Disutility Index section of the documentation for more details.

If multiple models are being calibrated or adjusted, it is perfectly acceptable to insert multiple Model Coefficients records into a single data set. Simply click on the "Duplicate RT" button prior to coding the new data.

**Network Parameters** (associated with record type 10)



### Node Number of Master Controller

All offsets (or yield points) in the network will be referenced to the coded reference interval of this controller. A '0' coded in this field implies that offsets are referenced to an arbitrary system offset time base. The offset (or yield point) of the master controller, typically set to 0 seconds by most users, will be held constant during the optimization process.

### Network-Wide Yellow Change Interval Length

This value is used when the initial timing model has been requested (Global > Run Controls screen or record type 1, field 11), and the yellow interval durations are not coded on record types 1X/18. This value is also used when creating new phases on the Edit > Timing screen. The default value is 4 seconds. Yellow change intervals are mandatory for each phase in TRANSYT, and the duration must be at least 1 second.

### Network-Wide All-Red Clearance Interval Length

This value is used when the initial timing model has been requested (Global > Run Controls screen or record type 1, field 11), and the all-red interval durations are not coded on record types 1X/18. This value is also used when creating new phases on the Edit > Timing screen. The default value is 1 second. All-red clearance intervals are optional for each phase in TRANSYT. It is also perfectly acceptable to combine the yellow and all-red interval times into the yellow interval time only. This achieves the same results from the traffic model as if the yellow and all-red interval times were coded separately.

### Network-Wide Approach Saturation Flow Rate

The value coded here, or the default value of 1700 vehicles per lane per hour of green, is used to determine the number of lanes on a link, in order to calculate queuing capacities. For example, a TRANSYT link with a coded saturation flow rate of 5000 is assumed to have 3 lanes, after dividing by three and rounding.

### Network-Wide Approach Speed for External Links

The value coded here, or the default value of 30 mph (48 kmph), is used in the calculation of fuel consumption on external links. This value may also be used to compute travel times, if external links are coded as part of a progression route. If desired, the user can code unique approach speeds for external links on the Edit > Feeders screen (or record type 28 field 9).

### Network-Wide Platoon Dispersion Factor

The platoon dispersion factor (PDF) is used by the TRANSYT platoon dispersion model in propagating traffic from one intersection to the next. The recommended default value is 0.35, based on empirical studies. The coded value must be an integer value, i.e.,  $PDF * 100$ . The PDF can also be changed for individual links on the Edit > Optional > Weighting > Platoon Dispersion Factors screen (or record type 39).

Research from PRC Engineering [17] and the University of Florida [20] suggests that the following values could be used to obtain reasonable estimates for the PDF. The default value of 0.35 implies moderate friction (light turning traffic, light pedestrian traffic, 11- to 12-ft or 3.3- to 3.6-meter lanes, possibly divided, well-designed CBD or fringe arterial street). A value of 0.5 implies heavy friction (combination of parking, moderate to heavy turning traffic, moderate to heavy pedestrian traffic, narrow lane width, traffic flow typical of urban CBD). A value of 0.25 implies low friction (no parking, divided, turning provision 12-ft or 3.6-meter lane width, suburban high-type arterial street).

### Fuel Consumption Multiplier

The value in this field will automatically and globally adjust all fuel consumption estimates. It is recommended to leave this field blank unless reliable data are available to justify the multiplier. The fuel consumption model may also be calibrated on the Global > Model Coefficients screen (or record type 8).

#### Average Vehicle Spacing

This input field is used to code the jam spatial headway (default value 25 feet or 7.6 m) and the optimal spatial headway (default value 75 feet or 22.9 m). Jam spatial headway is the space a vehicle occupies when standing in the queue. Queuing studies [45,46] within the U.S. have indicated that average jam headways are typically 25 feet. Optimal spatial headway is the space a vehicle occupies when departing from a queue. The internal calculation of queuing capacity is always affected by the jam spatial headway and link length. When using step-wise simulation, both variables are used in the simulation and calculation of the back of queue location. When using link-wise simulation, optimal spatial headway is ignored. A sample calculation (jam headway) is listed below:

	vehicle length	empty space	total spacing	fleet %	
small car	14	5	19	0.25	4.75
car	16	6	22	0.70	15.40
truck	35	7	42	0.05	2.10
					<b>22.25</b>

The format of this field is a "packed variable" in the format 'xxxyy' to permit entering 'xxx' as the optimal spatial headway and 'yy' as the jam spatial headway. Any value less than 100 coded in this field will be considered as jam spatial headway, and the optimal spatial headway will be set equal to its default value.

The metric unit range for jam spatial headway is 50-149 decimeters, with a default of 76 decimeters (7.6 meters). However, the packed variable format does not allow values greater than 99 for the jam spatial headway. To overcome this limitation, code an "equivalent" value for jam spatial headway between 0 and 99 using the following formula: Coded Value = Real Value - 50. For example, to model a jam spatial headway of 125 decimeters (12.5 meters), code a value of 125 - 50 = '75'.

#### Orientation Flag

This input field specifies the link numbering scheme, although any data specified within the Global > Link Numbering Scheme screen (or record type 3) would override this field. This field is also used whenever the orientation flag from the Edit > Optional > Route > Route Parameters screen (or record type 60) is left blank, even though other Route Parameters data have been specified.

Values between 1 and 4 indicate that the TRANSYT-7F Standard scheme will be used with orientation 1 = northbound, 2 = southbound, 3 = eastbound, 4 = westbound.

TRANSYT-7F Standard link numbers with northbound orientation:

Northbound left through right are 2, 1, and 9.  
 Southbound left through right are 4, 3, and 10.  
 Eastbound left through right are 6, 5, and 11.  
 Westbound left through right are 8, 7, and 12.

Values between 11 and 14 indicate that the NEMA scheme will be used with orientation 11 = northbound, 12 = southbound, 13 = eastbound, 14 = westbound.

NEMA link numbers with northbound orientation:

Northbound left through right are 5, 2, and 9.

Southbound left through right are 1, 6, and 10.  
Eastbound left through right are 7, 4, and 12.  
Westbound left through right are 3, 8, and 11.

#### Network-Wide Target Degree of Saturation for Actuated Phases

Under the traditional model, traffic-actuated phase lengths will be designed to achieve the target degree of saturation. Unused green time will be reassigned to the major street non-actuated phase. This can be useful prior to simulation or optimization of an actuated system. If the value is coded as negative, all actuated controllers are ignored. The default value is 85, which indicates a target degree of saturation of 85%.

The original target degree of saturation model, employed by TRANSYT for many years, is effective for optimization runs. This model facilitates location of the optimal actuated phase times. To ensure that the optimal actuated phase times materialize in the field, it is necessary to set the force-offs and maximum greens appropriately. Refer to the section on Traffic-Actuated Control for more details.

Target degrees of saturation are ignored by the actuated estimation model.

#### Threshold Degree of Saturation for Double Cycling

TRANSYT is capable of suggesting signalized intersections that may be good candidates for double cycling. Any signalized intersection whose maximum degree of saturation is less than the value coded here (or its default) will be so identified in the output file. The default value is 25, which represents a 25% threshold degree of saturation.

#### Queue Spillback Penalty

When the disutility index (DI) is specified as the performance index (objective function), a queuing penalty can be optionally included in the DI to minimize spillback effects. If step-wise simulation is used, the queuing penalty is the value coded in this field multiplied by the number of simulation time steps in which spillback occurs. If link-wise simulation is used, the queuing penalty is the value coded in this field multiplied by the excess maximum back of queue (the amount by which the maximum back of queue exceeds queuing capacity). The reason for the difference in the queuing penalty definition between the two simulation types is that the excess maximum back of queue is always zero in step-wise simulation, since no vehicle is allowed to leave the upstream stop line if the downstream link is full. The default value is 40, which indicates a queuing penalty of 0.4.

#### Inflation Rate

This field is used to update the cost estimate for future years. A 4% inflation rate would be coded as 400, which results in a multiplier of 1.04. The default value is 0. It is expressed as a ratio of the consumer price index in the desired year, divided by the base year. The base value is 140.3 (all items based on 1982-1984 being 100%). For future updates, see reference [21], which is carried at most public libraries. This factor may also be used by other nations other than the U.S. as a currency converter. For example, to indicate 1.8 German marks for every U.S. dollar, code '180'.

#### Cost of Fuel per Gallon

This field indicates the average cost of gasoline in U.S. dollars per gallon only. The default value is 125, which indicates a price of \$1.25 per gallon. The detailed cost model can be modified by advanced users on the Global > Model Coefficients screen (or record type 8). Part of that model, however, includes gasoline costs, which change frequently and may be changed from the default value using this field. Note: other nations must convert their cost and local currency per volume unit to US\$/gal to work properly in the model.

### Average Vehicle Occupancy

Also related to the cost model, this is the passenger occupancy for the average vehicle in the traffic network. It should take buses and other high-occupancy vehicles into account when applicable. The default value is 120, which indicates an occupancy of 1.20.

## **Route Screen (associated with record types 40-43, and 60-61)**

### **Flow Profile Diagrams** (associated with record type 40)

Flow Profile Diagrams (FPDs) for routes may be requested on the Route > Route Parameters screen (record type 60), but this screen is available for individual or randomly located FPDs. Flow Profile Diagrams Records are used to identify the links for which FPDs are to appear in the program's output file. Flow profile diagrams can now also be plotted by using the Profile View screen.

There is a limit of 300 links for which FPDs may be printed in the legacy report. Links may be repeated if desired. If additional FPDs are requested on the Route > Route Parameters screen, these do not affect the 300 link limit for this screen. Each is considered independently.

In the wide, 132-column output format, four flow profile plots will appear on one output page, with one plot in each quadrant. These are arranged on the page as described below. In the narrow, 80-column output format, only two plots are printed on each page. The plots that would normally be on the left side of the page in the wide output format will be printed first, followed by those plots which would normally appear on the right side of the page, both in the same vertical order.

The FPDs provide a picture of the arrival and departure of traffic at the downstream end of the link throughout the signal cycle. It is recommended that FPDs be printed by TRANSYT for several key links in the network and compared with actual flow profiles in the field (collected over several cycles). This will determine whether TRANSYT has accurately represented traffic behavior actually occurring on the street.

If the FPDs for an arterial street are requested here, it is recommended that the diagrams be arranged so that the links for a single direction along the arterial are aligned along the left sides of the output pages, and the links in the opposing direction are aligned along the right sides. This is the order in which they are generally displayed when requested on the Route > Route Parameters screen, provided that the links on the Route Link or Flow Profile Diagrams screens are arranged in this order. If any data field is left blank, that quarter of the output page will be left blank. If the narrow output format is requested, those plots which would appear on the left side of the page in the wide output will be printed first, followed by those plots that would have been printed on the right side of the page. This permits maintaining directional continuity of the plots.

### **Bandwidth Constraints** (associated with record type 41)

This screen provides link-specific data associated with green bands that will not be disturbed by TRANSYT hill-climb optimization. These green bands will generally be developed by some other process or program. The data on this screen is naturally ignored in a simulation-only run.

There may be a maximum of four bands passing through any given intersection. Normally, the bands will apply only to major arterial streets, and may apply to both directions of travel. Major arterial streets may intersect, so provisions are made for the total of four bands.

The bandwidth constraint is applied to individual links, independent of their relationship to one another, or to other intersections; thus the analyst must ensure that the progression bands are properly coded. Therefore it is suggested that the bandwidth constraint records be arranged in a logical sequence

along each arterial street. Intersecting bands may be defined on separate records, and duplicate records may be generated on this screen by clicking the "Duplicate RT" button.

Note that if any intersection has links included on this screen, the complete timing plan for all intersections must be coded, and the initial timing flag (Edit > Analysis screen or record type 1) must be set to 0. Further, timing for the intersections with constrained links must be consistent with the band coordinates coded on this screen. Bandwidth constraints may be mixed with PROS, but be careful to coordinate the timing on all arterial streets to ensure the timing and the bands are consistent.

#### Link Number Applicable to the Band

This is usually a through-traffic link. It cannot be secondary shared-lane link or a 100% green link.

#### Start Time of the Current Thru Band

The offset time from the system time reference base to the start (or leading-edge) of the through green band.

#### End Time of the Current Thru Band

The offset to the end (or trailing edge) of the through band.

#### **Arterial Routes** (associated with record type 42)

To use the PROS performance index, or to use the time-space diagram screen, it is necessary to identify the individual routes for which PROS will apply. Although progression routes may be coded here, they are coded more easily by using the Map View. Multiple arterial route records are allowed, and there is a maximum of 50 uni-directional routes in a network. There may be up to 25 two-way routes, or any combination of one-ways, as long as the total number of directions is less than or equal to 50.

Contrary to other link list records, link pairs may not be omitted. The list should be continuous; although, for a one-way route (entirely one-way only) one of the links in each pair must be left blank ('0'). Bottlenecks and unsignalized intersection links on the route must be included in the link list to provide for their lengths.

The following guidelines apply to link pairs:

1. Links included on a PROS route need not be through links, but all listed links must be protected and not opposed. Thus protected left-turns and right-turns are allowed. If a turning movement is protected-plus-permitted, PROS are only tabulated for the protective phase(s).
2. All links downstream of other links included in the list must have their upstream links included as an upstream feeding link on the Edit > Feeders screen (or record type 28). In other words, other than the entry links (which do not have to be external links), all downstream links must be connected by link-to-link progression on the Feeders screen.
3. All links listed must be primary (if shared lane), through (or at least not-opposed) links.
4. When link pairs are listed, they must enter the same intersection. One-way streets may be included, but they must be one-way for the entire length. For streets that are in fact two-way for some distance, and then separate into a one-way pair, it would be necessary to define three routes in order to model PROS throughout.
5. Links must be listed in the logical upstream-to-downstream sequence, including any bottlenecks and all-green links.

6. Any one intersection can be included on several routes, but there is a limit of four links maximum at a node.

7. Any given route can have intersections that are common to other routes. That is, any route can have multiple intersecting routes.

8. Routes may be defined in any order. They are optimized simultaneously, so there is no need to sequence them, unlike the Edit > Optional > Global > Optimization Node List (record type 2). However, the overall set of route numbers should be "1-based" and continuous. If a master intersection has been defined in the Edit > Optional > Global > Network Parameters screen (record type 10), and also in a route, it must be in the first route.

9. All-green (100% green) links and bottlenecks may be included, but PROS are not tabulated for these.

Refer to the PROS section of the documentation for figures and examples of proper and improper arteries for PROS.

### **Arterial Weighting** (associated with record type 43)

When using one of the PROS objective functions, individual routes can be given preference, and/or directions on the routes can be given preference, during the optimization process. The weighting factors are proportional. For example, if '66' is coded for one route and '33' for another, the former will have twice the weight applied to PROS than the first. The arterial reference number specified here should be identical to the one from the Arterial Routes screen.

Although this is not a specifically required approach, the weighting factors can be considered as percentages. For example, to achieve 60% of progression in the forward direction, code '60' and '40' as the forward and reverse factors, respectively. If a 50:50 split on a second route is desired, but double the priority as the first route, code '100' in both fields as both the forward and reverse factor.

### **Route Parameters** (associated with record type 60)

In the record type data file, the Route Parameters record type 60 must be followed immediately by the route title, and then Route Link record type 61. The route title is a comment record with letters beginning in column 1. Although the Route Parameters and Route Link record types 60 and 61 are generated in the background by the input editor, the Route Title button can be used to edit the route title between record type 60 and 61. Multiple routes may be requested by multiple groups, where a group consists of record type 60, route title, and then record type 61. Although duplicate record types can be generated by pressing the "Duplicate RT" button, the record type editor should be used to ensure that the proper order (record type 60, route title, and then record type 61) is maintained.

The route-specific outputs available directly from TRANSYT are:

- Time-Space Diagram
- Flow Profile Diagram
- Route Summary Report
- Time-Location Diagram

Flow profile diagrams can also be requested on the Route > Flow Profile Diagrams screen, but they are also accessible here, which simplifies the process of obtaining multiple diagrams and other route-specific summaries for a given route.

If the user selects File > New from TRANSYT-7F, a dialog box appears that contains a field for the "Number of Routes". For each route selected in this dialog box, a template record type 60, route title, and record type 61 should be generated in the new data file.

### Time-Space Diagram Flag

The number 1 indicates that a time-space diagram will be printed for the route defined on the subsequent record type 61 (Route > Route Link screen).

### Time-Space Units Flag

0: The units on the time axis are indicated by the seconds/percent flag from the Edit > Optional > Global > Run Controls screen (record type 1).

1: The units on the time axis are in seconds, regardless of the value of the seconds/percent flag from the Edit > Optional > Global > Run Controls screen (record type 1).

2: The units on the time axis are in percent, regardless of the value of the seconds/percent flag from the Edit > Optional > Global > Run Controls screen (record type 1).

### Time-Space Time Axis Scale Factor

This is the number of seconds (or percent of the cycle length) associated with each character printed on the time axis. If this field is left blank, the default is three seconds per character, or three percent per character, depending on the seconds/percent flag (Edit > Optional > Global > Run Controls screen or record type 1).

### Time-Space Distance Axis Scale Factor

This is the number of feet or meters associated with each character printed on the distance axis. If this field is left blank, the default is 67 feet per line, or 20 meters per line, depending on the English/metric flag (Edit > Optional > Global > Run Controls screen or record type 1).

### Flow Profile Diagram Flag

The number 1 indicates that a series of flow profile diagrams will be printed for each link on the route defined on the subsequent record type 61 (Route > Route Link screen).

### Route Summary Report Flag

The number 1 indicates that a route summary report will be included in the output file.

### Platoon Progression Diagram Flag

The number 1 indicates that platoon progression data will be generated and written to a file. Following completion of the TRANSYT run, this data file can be used as input to the platoon progression diagram (PPD) program to produce the requested plots. The MS-DOS based PPD program is no longer distributed with TRANSYT-7F. This program can be downloaded from the *McTrans* web site, if desired.

### Single Ring Timing Flag

The number 1 indicates that a single ring timing data set will be written.

### Time-Location Diagram Flag

This is a different type of time-space diagram, in which the distances between intersections have been collapsed. If the value is coded as negative, the diagram is written without the PROS values.

1: A time-location diagram will be printed including actual PROS values, and the time units are in steps.

2: Same as 1, except the time units are seconds.

3: Same as 1, except the time units are percent of cycle.

## Route Orientation Flag

This indicates the compass direction of the first link listed on the Route > Route Link screen (record type 61), or the Route > Arterial Routes screen if the arterial reference number is used on record type 61. This orientation is used as the forward direction on all associated time-space diagrams.

0: Use the orientation defined on the Edit > Optional > Global > Network Parameters screen (record type 10 field 10)

1: Northbound

2: Southbound

3: Eastbound

4: Westbound

## **Route Link** (associated with record type 61)

Route Link Records are used to identify the links associated with route outputs. As many Route Link Records as necessary may be included for a single route, subject to the limit of 300 links. The Route Link Record(s) must appear immediately after the route title record (i.e., the comment record indicating the route title).

If a route number is coded on this screen (because link numbers were already coded on the Arterial Routes screen), it should begin at '1', and should not skip any numbers. For example, if there are 4 progression routes in the network, they should be numbered 1, 2, 3, and 4. Unusual route numbers or route names can be documented by using the route title record.

Note that the links must be entered in pairs that end at the same node. There are no restrictions on the link pairs that appear in the list purely for route summary outputs. For example, the route may turn corners, and primary shared-lane links and bus links may be included in the diagram.

The distance between intersections in a time-space diagram is based on the length of the first link in each pair of links coded. One-way streets may be included in the route summaries. Simply code the appropriate link numbers for the one-way streets in the appropriate field of each pair and leave the other field in each pair blank.

If a bottleneck exists on the route, the bottleneck link(s) must be included in the link list in order to maintain the correct space relationship between the signalized intersections. Bottlenecks and unsignalized intersections are permitted in time-space diagram, flow profile diagram, and platoon progression diagram requests, but not single ring timing data set requests.

## **Timing Screen (associated with record types 1X, 18, 2Y, and 2Z)**

This is the non-graphical Timing screen, in which timing plans are coded at the "link level". For most practical purposes, it is preferable to use the more intuitive Timing View screen, in which timing plans are coded graphically. However, the non-graphical Timing screen may be needed for coding complex timing plans that the Timing View screen cannot handle.

### **Phase #1-#7** (associated with record types 2Y and 2Z)

Phasing data records are used to define signal phases for the signalized intersection in question. The intervals that comprise each phase and the links receiving green during each phase are defined. TRANSYT assumes that any given phase starts with the Starting Interval Number and ends with either the Yellow Interval Number or the Red Interval Number. If a Red Interval Number has been coded, this interval ends the phase in question.



Interval numbers must be sequential, and the Starting Interval Number of each phase must always be one interval number more than the yellow (or all-red) interval of the preceding phase. If a green interval actually exists, which is neither the Starting Interval Number nor the Variable Interval Number, it is omitted from the data entry screen. The list of moving links may be in any order, but all links moving in each phase must be included on the list (which may include a "phasing continuation" record, see below for a description of phasing continuation).

For a phase with only two intervals, these intervals are the green and yellow vehicle intervals. The green interval number is both the Starting Interval Number and the Variable Interval Number, although the Variable Interval Number field may be left blank. The Yellow Interval Number is specified accordingly, and it marks the end of the phase. The All-Red Interval Number field is left blank. The following is a two-phase example, with minimum phase times of 20 seconds, moving links 101 through 108, with negative signs indicating the permitted links:

1	1	2	0	20	101	-102	103	-104...
3	3	4	0	20	105	-106	107	-108...

A phase with three or more intervals often contains pedestrian signal indications. The vehicle green/pedestrian WALK interval is typically specified as the Starting Interval Number. This would also normally be the Variable Interval Number. The green/flashing DON'T WALK interval may alternatively be designated as the variable interval, but this is not recommended. The DON'T WALK interval can be assumed to be part of the variable interval, provided that the minimum phase durations reflect what is required by pedestrians.

An exclusive pedestrian phase can also be represented in TRANSYT. The exclusive pedestrian WALK interval is specified as both the Starting Interval Number and the Variable Interval Number. The pedestrian flashing DON'T WALK interval can be specified as the Yellow Interval Number, provided that the interval duration does not exceed 9 seconds, because then the program will divide by 10 and treat as a non-integer number. If the flashing DON'T WALK interval actually exceeds 9 seconds, then a portion of this interval must be modeled as part of the exclusive pedestrian WALK interval, such that the flashing DON'T WALK interval durations can be 9 seconds or less. The optimization process will likely force such an exclusive pedestrian phase length to its minimum specified duration, especially if no pedestrian links with pedestrian traffic are modeled. Because of this, the minimum phase length specified should be sufficiently long to ensure that the variable interval is not reduced to an unsafe length.

Note that multiple green intervals are not currently supported when running phasing sequence optimization. If multiple green intervals have been coded for a single phase, an error message will likely occur. Multiple green intervals are sometimes coded on the Timing screen, to indicate specific pedestrian or fixed green intervals that are separate from the variable interval, where the variable interval is subject to optimization. An alternative technique for enforcing specific pedestrian or fixed green intervals would be to code the appropriate minimum phase time, also on the Timing screen.

If in reality there is no yellow or all-red change interval, a dummy yellow interval must be coded nonetheless. Overall minimum phase times must be at least 2 seconds to account for a 1-second green interval and a 1-second yellow interval. If the user wishes to simulate an overlap phase as small as 1 second, this can be accomplished if the start-up lost times and extension of effective green times are correctly coded. On the surface, it is necessary to either specify a minimum 2-second overlap, or no overlap. Refer to the Overlap Phasing section of the documentation for more details.

Link numbers of all links receiving green are specified in the Moving Link #1-#8 fields. Since the current version of TRANSYT is capable of simulating up to 10 links moving during any given phase, the Phasing Continuation screen is available for specifying additional moving links. Links moving in multiple phases must be specified in each associated phase. If a link receives 100% green, it must be coded in all phases. If a link has 100% green, but is likewise identified as a permitted movement (by a negative link number) in some or all phases, TRANSYT assumes this is a stop/yield-controlled approach during the permitted phase(s).

TRANSYT-7F currently does not support changes to the phasing sequence between different time periods. As such, when a multi-period analysis file is loaded, input data within the Phase #1-#7 tabs automatically applies to all time periods.

#### Starting Interval Number

This is the interval number that starts the green for the given phase. If the value is changed from zero to a nonzero value, a new record type for the given phase is automatically generated within the TRANSYT-7F data file. Similarly, if the value is changed to zero, the record type for the given phase is automatically deleted.

#### Variable Interval Number

This is the interval number that is subject to optimization. It is typically identical to the starting interval number unless a fixed green interval is desired, although the recommended technique for obtaining fixed green is through the minimum phase time input parameter (see below).

#### Yellow Interval Number

The yellow interval number is mandatory and the yellow interval length (specified in the Timing tab) must be at least 1 second.

#### All-Red Interval Number

The all-red interval number is optional, and the all-red interval length (specified in the Timing tab) has the same impact as the yellow interval length, from a modeling standpoint. For example, combining the yellow and all-red intervals into the yellow interval only is acceptable and should have no impact on results.

#### Minimum Phase Time

Primarily applicable to optimization, this represents the minimum sum of the green, yellow, and all-red intervals. Initial user-coded timing plans (specified in the Timing tab) must not violate the minimum phase times, and when the program is working properly, optimized phase times will not violate the minimum phase times.

#### Moving Links #1-#8

Link numbers moving during the phase in question are specified here. Negative link numbers indicate permitted movements. Press the "Next Data" pushbutton to access link numbers #9 and #10. For exclusive pedestrian phases, the user can either leave the link list blank, or code a low-volume "dummy link".

#### Moving Links #9-#10

The current version of TRANSYT allows a maximum of 10 links moving during each phase. New data files created by the TRANSYT-7F GUI always contain space for moving links #9 and #10.

If pressing the "Next Data" button does not load the screen for moving links #9 and #10, this implies that the phasing continuation record (record type 2Z) is not present in the data file. It also implies that the data file was initially created by an older version of the software, or perhaps by a separate preprocessor program. The record type editor can be used to insert record type 2Z manually. This requires a duplicate record type 2Y for the same phase, but with a '2' coded in field 16. Remember that the phasing continuation record type 2Z is not mandatory unless there are more than 8 links moving during a single phase.

### Actuated Flag

A value of 1 in this field indicates that the phase in question is actuated, whereas a value of 0 indicates that it is non-actuated. Within the TRANSYT model, if any individual movement is capable of actuating or terminating the phase, the entire phase is considered actuated.

### **Timing** (associated with record types 1X and 18)

The signal timing at each signalized intersection in the network must be defined in terms of intervals for TRANSYT. A different interval occurs at a node each time a unique set of vehicular or pedestrian signal indications appears during a cycle.

Intervals are classified in TRANSYT as being either fixed or variable. Fixed intervals, as the name implies, are those intervals whose durations are a specified amount of time. For example, yellow and all-red intervals are always fixed intervals. Variable intervals are those intervals whose durations may be changed by TRANSYT-7F during the optimization process. There can be only one variable interval for each signal phase. The vehicle green interval will be classified as variable. If pedestrian signal indications exist, the variable interval is usually the WALK interval. Any vehicular green interval, however, may be designated as the variable interval, which is done on the Timing > Phase #1-#7 screen (or record type 2Y).

If an initial timing plan is furnished by the user, all interval durations and offset (or yield point) values must be specified. When the sum of the interval durations is less than the cycle length, a warning will be printed, but the run may continue. Any unspecified timing may be assigned to the last phase. If the sum of the interval durations is greater than the cycle length, a fatal error message will be issued.

When the initial signal timing flag (Edit > Optional > Global > Run Controls screen or record type 1) is coded as '1', only the durations of fixed green intervals must be specified for the nodes to be optimized. Fixed intergreen interval durations (yellow and all-red) may be left blank for these nodes, in which case TRANSYT will assume the values coded in the Edit > Optional > Global > Network Parameters screen (or record type 10). Initial variable interval durations and offset (or yield point) values will be provided by TRANSYT. For pre-timed controllers, the initial timing model resets all offsets, and calculates phase durations that equalize the degrees of saturation on the critical links. Traffic-actuated controllers have a separate initial timing model, where phase durations are optimized according to the target degree of saturation, and any remaining green time within the cycle is donated to the major street non-actuated phase.

When the initial signal timing flag is coded as '2', initial variable interval durations values are provided by TRANSYT, but offsets coded by the user are preserved. Refer to the Initial Timing Model section of the documentation for mathematical details.

For nodes not included on the Edit > Optional > Global > Optimization Node List screen (or record type 2), interval durations and offset or yield point values must always be provided, and the sum of the interval durations must always equal the cycle length.

TRANSYT references the offset or yield point to the beginning of the specified interval of the controller. The default reference interval if none is specified is interval #1.

If a controller is actuated, the appropriate actuated flags should be specified, and the phase duration timing may be handled in one of three ways:

1. Code average phase lengths as measured in the field. This applies to evaluation of existing conditions, but is not necessary for optimization. When using this procedure, the sum of the interval lengths must equal the cycle length, or an error message may be issued.

2. Allow phase lengths to be initialized by the target degree of saturation method. This is best for optimization runs. Minimum phase times coded by the user should reflect safety and/or optimization requirements.
3. Allow phase lengths to be estimated by the actuated estimation model. This is best for simulation-only runs, or evaluation of existing conditions. Minimum phase times coded by the user should reflect safety requirements.

Optimization will not modify the offset or yield point of the master controller. Offsets or yield points for other nodes are referenced to the offset or yield point of the master controller. The TRANSYT legacy report will display the word "Offset" within pre-timed signal timing tables, and will display the words "Yield Point" within actuated signal timing tables.

An intersection can be designated as having only one phase, which will normally indicate that the intersection is sign-controlled. In this event all links associated with this node will have 100% green time, thus no interval durations need to be coded.

Signalized intersections that are double-cycled will complete two equal local signal cycles in the same time that other nodes in the network will complete one full cycle. This feature should be used when it is desired to operate certain intersections at a shorter cycle length than the system cycle length. TRANSYT can recommend good candidates for this treatment within the legacy output file, based on the threshold degree of saturation for double cycling (Edit > Optional > Global > Network Parameters screen or record type 10).

Signal timing information for double-cycled intersections may be provided for only one half cycle. TRANSYT will automatically repeat the signal timing information for the second half cycle. Only three single-ring phases may exist at a double-cycled intersection, resulting in a maximum of six single-ring phases within the network-wide system cycle length(s).

On record type 1X that specifies Timing, the number of phases for this intersection is automatically written into field 1. The value here indicates the number of single-ring phases at the given intersection, where the maximum number of single-ring phases recognized by the program is 7. For example, record type 17 would indicate 7 phases. Be careful not to confuse the number of single-ring phases with the number of dual-ring phases. Signal controllers frequently have at least 8 dual-ring phases, but the equivalent single-ring plan always has fewer phases, due to concurrent movements.

The node number for each intersection is listed in field 2 of record type 1X. However, an option is available for coding negative node numbers, using the record type editor. When the node number is coded as negative in field 2, a second network-wide output table is written whose results exclude the negative node numbers. This option is only applicable to simulation-only runs and simple hill-climb optimization runs, and is not applicable to genetic algorithm optimization, traffic-actuated signal timing estimation, or cycle length optimization.

#### Offset or Yield Point

The reference interval at the intersection in question begins a certain amount of time, specified by the offset or yield point, after the master intersection's reference interval begins. The term "offset" applies to pre-timed controllers, whereas the term "yield point" applies to coordinated actuated controllers. The offset or yield point value cannot exceed the cycle length value.

When the initial timing flag (from Edit > Optional > Global > Run Controls) is on, this field may be left blank, unless this is the master controller (see Edit > Optional > Global > Network Parameters), and its offset/yield point is not zero. In this case, the actual offset or yield point should be coded.

If the value in this field is coded as negative, then this intersection is treated as uncoordinated, regardless of the upper and lower cycle length ranges specified on the Edit > Optional > Global > Run

Controls screen. In this manner, the user can specify part of the system to be coordinated and the other part to be uncoordinated. The offset or yield point value coded in this field is ignored if this is an uncoordinated intersection.

#### Reference Interval Number

By default the reference interval number is 1 at any given intersection. However, for coordinated actuated controllers, it is preferable to specify the (major street through) non-actuated yellow interval number as the reference interval number, in order to achieve consistency with controllers in the field and/or microscopic simulation programs such as CORSIM.

The program assumes that the offset or yield point will be referenced to the start of the specified interval. This field is ignored if this is an uncoordinated intersection.

#### Duration of Interval #1-11

The first 11 interval durations are specified here, and the interval definitions (green, variable, yellow, all-red) are specified within the Phase tabs. Interval numbers 12 and higher are specified on the second Timing tab. Interval durations are specified in units of seconds.

#### Duration of Interval #12-25

The final interval durations are specified here, and the interval definitions (green, variable, yellow, all-red) are specified within the Phase tabs. Interval numbers 11 and lower are specified on the first Timing tab. Interval durations are specified in units of seconds.

#### Phasing Flag

If it is desired to run this controller in two equal sub-cycles ("double cycle") within the system cycle, code '1' here (column 80 of record type 1X). If a controller is double cycled, the system cycle length (or all cycle lengths in a range to be evaluated) must be even numbered. Phasing sequence optimization cannot be performed at double cycled nodes (see below).

Optional restrictions on phasing sequence optimization are also specified here, although it is much easier to code these restrictions on the Edit > Analysis > Phasing Sequence Optimization screen. Column 79 (of record type 1X) potentially contains the code for North/South phasing restrictions at the intersection in question. Column 80 potentially contains the code for East/West phasing restrictions at the intersection in question. Phasing restriction codes are listed in the table below.

Allow...				
Optimization	Overlap	Lead-Lag	Yellow Trap	Code
No	No	No	No	9
Yes	No	No	No	8
Yes	Yes	No	No	7
Yes	No	Yes	No	6
Yes	No	No	Yes	5
Yes	Yes	Yes	No	4
Yes	Yes	No	Yes	3
Yes	No	Yes	Yes	2
Yes	Yes	Yes	Yes	0 or blank

In addition, column 77 potentially contains the code for North/South leading through movement restrictions at the intersection in question. Column 78 potentially contains the code for East/West leading through movement restrictions at the intersection in question. In these columns, a '1' indicates that leading through movements are not allowed for the approaches in question.

#### Allow Split Optimization

This checkbox specifies whether or not split optimization will take place at this intersection. This checkbox is only applicable if optimization of splits has been requested globally on the Edit > Analysis screen. If this checkbox is not selected, but the splits are being modified anyway during a run, this is probably due to the initial timing model. In order to de-activate the initial timing model, it is necessary to remove the intersection node number from the optimization node list, or turn off the initial timing flag.

When viewing the record types, a negative record type 1X indicates the prevention of split optimization, although viewing the record types is not required.

#### Allow Offset Optimization

This checkbox specifies whether or not genetic algorithm optimization of offsets will take place at this intersection. This checkbox is only applicable if genetic algorithm optimization of offsets has been requested globally on the Edit > Analysis screen. If this checkbox is not selected, but the offsets are being modified anyway during a run, this is possibly due to cycle length optimization, or the initial timing flag.

If hill-climb optimization has been selected, offset optimization cannot be prevented through the use of this checkbox except for grouped nodes, and the master controller.

When viewing the record types, a '1' coded in column 76 on record type 1X indicates the prevention of offset optimization, although viewing the record types is not required.

#### Coordinated Signal

This checkbox specifies whether the controller at this intersection is coordinated with other controllers in the network. By default this checkbox should be turned on at every intersection, so the flow profiles can be simulated properly. Flow profiles can be observed by using the Profile View screen.

Whenever the Coordinated Signal checkbox is turned off, this means the flow profile will be simulated as perfectly flat or uniform, and there will never be an incoming platoon. In other words, turning off this checkbox would have the effect of deactivating any feeding flow data from the Edit > Feeders screen. Therefore this checkbox should only be turned off if there are no nearby upstream signals, or if the subject intersection has a different cycle length than the background system cycle length.

Uncoordinated signals are allowed to have cycle lengths that are different from the network background cycle length. When using genetic algorithm optimization, it is possible to optimize the cycle length and splits at an uncoordinated intersection independently of the network cycle length. However, it is not possible to achieve reliable progression through intersections that are not coordinated. If the Coordinated Signal checkbox is turned off, initial timing cannot be automatically provided for this intersection, and the local cycle length does not need to be equal to the network background cycle length. This checkbox has no effect if there is only one intersection in the data file, or if the Uncoordinated Network checkbox from the Edit > Analysis screen has been turned on.

When viewing the record types, a negative offset value coded on record type 1X indicates an uncoordinated signal, although viewing the record types is not necessary. On record type 1, a negative cycle length indicates an uncoordinated network.

## Queuing Screen (associated with record types 28 and 291)

Note that all input parameters on this screen, except for the spatial headways, are coded automatically when using the Lane Configuration screen (Edit > Lanes).

### Queuing Capacity (veh)

TRANSYT always computes queuing capacity internally, based on the jam spatial headway and link length. To simulate queue blockage and oversaturated conditions, further adjustments to the queuing capacity are made based on degree of saturation and the number of simulated cycles. Therefore, the internal calculation of queuing capacity is somewhat detailed when using step-wise simulation.

This input field can be used to override the internally calculated queuing capacity on individual links. Queuing capacity is specified here in units of the total number of vehicles, across all lanes of the link. In order to modify the queuing capacity calculation globally, it is easier to use either the global value of jam spatial headway on the Edit > Optional > Global > Network Parameters screen (record type 10), or the queuing capacity multiplier on the Edit > Optional > Weighting screen (record type 34).

Queuing capacity affects the simulation of queue spillback and queue spillover, and also provides the analyst with a possible indicator of queue blockage problems. When analyzing short turn pockets, the user has the option of specifying a relatively small queuing capacity based on what the turn pocket could hold. However, be careful about whether queue spillback or spillover is taking place, because spillback is the default behavior. Spillover can only be simulated if other entries on the Queuing screen are used. The overall link length of the turn pocket should still be coded as the full length of the adjacent through lane, in order to achieve proper simulation of platoon dispersion.

If multiple links are specified as being shared (Edit > Optional > Global > Shared Lanes screen, or record type 7), queuing capacity must only be coded for the primary link, and this queuing capacity will apply collectively to all of the shared-lane links. An overall queue length will be estimated for the combined shared-lane group, and reported for the primary link.

### Left Adjacent Link Number or Right Adjacent Link Number

This input field specifies the number of a link (generally a left-turn or right-turn link with a turn bay) that affects and/or is affected by congestion on the "current" link, where the current link is listed in the same column at the top of the Queuing screen. This field should only be coded if the current link is a through link. Refer to the Queue Spillback and Queue Spillover Examples section of the documentation for examples of how this data is typically coded.

### Saturation Flow Rate Reduction due to Left or Right Adjacent Link (%)

This input field specifies the percent reduction in saturation flow rate on the current link, due to queue spillover from the adjacent link. For example, if there are 3 thru lanes, the saturation flow rate reduction (caused by spillover into one of these lanes) might be 33% or higher.

### Turn Bay Flag

This input field indicates whether the current link is a turn bay or not. For example, to specify left-turn bay spillover to the adjacent through lane, code the left adjacent link number and its associated saturation flow rate reduction, for the through link. For the left-turn link, simply code a '1' in this field to indicate that this is a turn bay. This field can be ignored if the user does not wish to model queue spillover.

### Upstream Saturation Flow Rate Reduction #1-#4 (%) (press New Data or Shift-PageDown to access

additional data)

By default, upstream feeding links specified on the Edit > Feeders screen (or record type 28) have their saturation flow rates reduced by 100% in the event of queue spillback. These fields allow the value to be changed to a value other than 100% when appropriate. For example, when the feeding link is an upstream shared lane, this field can indicate that only a portion of the flow is blocked by queue spillback.

Jam Spatial Headway (ft or m\*10) (press New Data or Shift-PageDown to access additional data)

This input field can be used to override the network-wide default value specified on the Edit > Optional > Global > Network Parameters screen (or record type 10). Jam spatial headway is defined as the space a vehicle occupies when standing in a queue. The internal calculation of queuing capacity is always affected by the jam spatial headway and link length. The acceptable range of input data is 15-50 feet or 4.5-15 meters (code '45' to request 4.5 m). If no link-specific value is coded here, the default value applied is either the global value coded on the Edit > Optional > Global > Network Parameters screen, or 25 feet (7.6 meters), if no global value has been coded. Queuing studies [45,46] within the U.S. have indicated that average jam headways are typically 25 feet. A sample calculation (jam headway) is listed below:

	vehicle length	empty space	total spacing	fleet %	
small car	14	5	19	0.25	4.75
car	16	6	22	0.70	15.40
truck	35	7	42	0.05	2.10
					<b>22.25</b>

Optimal Spatial Headway (ft or m\*10) (press New Data or Shift-PageDown to access additional data)

This input field can be used to override the network-wide default value specified on the Edit > Optional > Global > Network Parameters screen (or record type 10). Optimal spatial headway is defined as the space a vehicle occupies when departing from a queue. When using step-wise simulation, both the jam and optimal spatial headway are used in the simulation and calculation of the back of queue location. When using link-wise simulation, optimal spatial headway is ignored. The acceptable range of input data is 45-150 feet or 13.5-45 meters (code '135' to request 13.5 m). If no link-specific value is coded here, the default value applied is either the global value coded on the Edit > Optional > Global > Network Parameters screen, or 75 feet (22.9 meters), if no global value has been coded.

## Actuated Screen (associated with record type 291)

Gap Setting or Unit Extension (sec)

The gap setting indicates an amount of time, in which a presence detector must be unoccupied, before the detector calls for phase termination via gap-out. TRANSYT uses this value in the calculation of random delay. In addition, the signal timing estimation model uses this value in the calculation of traffic-actuated phase times. The default value is 3.0 seconds, and the allowable range is from 2.0 to 5.0 seconds.

### Signal Timing Estimation Data



All input fields in this section of the screen pertain to the actuated estimation model. Refer to the Actuated Estimation Methodology section for more details on the model, and refer to the Actuated Estimation Examples section for tutorial examples. Technical aspects of the actuated estimation model are also summarized in Appendix A [37], and detailed in the supplemental research report "Modeling Traffic-Actuated Control with TRANSYT-7F" [36].

#### Detector Length (ft or m)

This input field indicates the length of the presence detectors in the traffic network. It is used by the signal timing estimation model in the computation of traffic-actuated phase times, although the effect on the overall phase time is usually minor. The detector length is specified in units of feet or meters, and the default value is 30 feet (9 m).

#### Estimation Flags

This input parameter can be ignored for most data sets. This field provides an optional flag to indicate either a barrier-crossing actuated phase, or the coordinated non-actuated phase. Barrier-crossing actuated phases typically have longer phase times due to stochastic effects. All competing links are capable of extending the phase prior to crossing the barrier. Stochastic effects are quantified by the queue calibration factor (fq) along with increased green extension times. Regarding the coordinated non-actuated phase, it is important for the program to know exactly which phase this is, because this has a significant impact on the assignment of green time within the cycle. Normally the software correctly guesses the coordinated non-actuated phase based on the user-specified link numbering scheme and phasing sequence.

By default, actuated through movements that terminate with the subject phase are assumed to be barrier-crossing actuated phases; however, code the barrier-crossing phase number in field 16 to override this assumption. The "barrier" denotes the transition between North-South and East-West phases.

Also by default, the coordinated non-actuated phase is the first non-actuated phase located on record type 2Y. Code 10 + the coordinated non-actuated phase number in field 16 to override this assumption. Non-actuated phases that are not the coordinated phase will not have their phase duration re-calculated during the actuated estimation run. Normally, to model a coordinated actuated system, it is recommended to code all phases as actuated except for the coordinated non-actuated phase.

#### Queue Service Time Model

The queue service time model calculates the amount of time required to serve the initial queue, after the signal has turned green. The "T7F" selection indicates that queue service times will be obtained directly from TRANSYT step-wise simulation. Other models available for selection are existing models from the literature. The "T7F" model has been implemented as the default model within the signal timing estimation model based on its modeling structure and its correlation with CORSIM. Queue service times are taken directly from the simulated uniform queue profile, which can be plotted using Spyglass and the Profile View screen. During this process, an additional adjustment is made to account for stochastic effects. Numerically computed queue service times are reported in the estimation output file "t7fact.out".

#### Green Extension Time Model

The green extension time model calculates the amount of time until a suitable gap has been located at the detector, following the queue service time. Models whose names begin with the letters "GP" were developed exclusively for use with TRANSYT. Models whose names begin with the letters "GL" are existing models from the literature. The "GP1" model has been implemented as the default model within the signal timing estimation model based on its modeling structure and its experimental correlation with CORSIM. Green extension times are computed using applied probability analysis of the simulated input flow rate profile, which can be plotted using Spyglass and the Profile View screen. During

this process, flow profile adjustments are made to account for stochastic effects. Numerically computed green extension times are reported in the estimation output file "t7fact.out".

#### Critical Link Search Flag

The Critical Link Search Flag check box is toggled on by default to indicate that the signal timing estimation methodology will be applied to each candidate link. When complex conditions and overlapping actuated phases prevail, the critical link search is necessary to obtain correct results. When simple conditions prevail, removing the search flag will not affect results, and will allow faster running times on the computer.

#### Global Detector Length (ft or m)

The Global Detector Length field is used to specify the default length of presence detectors throughout the network. The detector length value potentially affects the green extension time. Link-specific detector lengths can be specified at the top of the Actuated screen (or record type 291 field 15).

#### Global Gap Setting (sec)

The Global Gap Setting field is used to specify the default gap setting (a.k.a. unit extension) throughout the network. The gap setting value potentially affects vehicle delays and/or average phase times. Link-specific gap settings can be specified at the top of the Actuated screen (or record type 291 field 14). Although the Global Detector Length and Global Gap Setting do not affect model results directly, these values are 1) automatically assigned to any new links created on the Lanes or Traffic screens, and 2) used by the actuated estimation model, if no link-specific values are listed on record type 291.

#### Convergence Override Flag and Number of Iterations Override

The Convergence Override Flag check box is toggled off by default to indicate that convergence will be achieved when the signal timing plan appears to be stable. If this box is checked on, then the user must specify the desired number of actuated model iterations to be computed, using the Number of Iterations Override Field.

## **Permitted Screen (associated with record type 29)**

Refer to the Permitted Movements section of the documentation for a detailed definition and description of permitted movements. When coding the phasing sequence in the Edit > Timing (View) screen, the software automatically draws permitted movements as dashed arrows, to distinguish between them and protected movements. In the case of a turning movement opposed by another turning movement, with no thru movements present, the user can manually specify that one of the turn movements is permitted (i.e., yields to the other one) by coding the permitted movement's link number as negative on the Edit > Optional > Timing screen.

#### Opposing Link #1-#3

The subject link must yield to the opposing link numbers specified here. Nonzero link numbers will enable the Opposing Flow fields. When creating a brand-new file using the TRANSYT-7F GUI, opposing link numbers are automatically coded. If right-hand drive is in effect, opposing links for the left-turn movements are automatically coded. If left-hand drive is in effect, opposing links for the right-turn movements are automatically coded. Opposing flows (see below) for these opposing links are automatically coded as 100%.

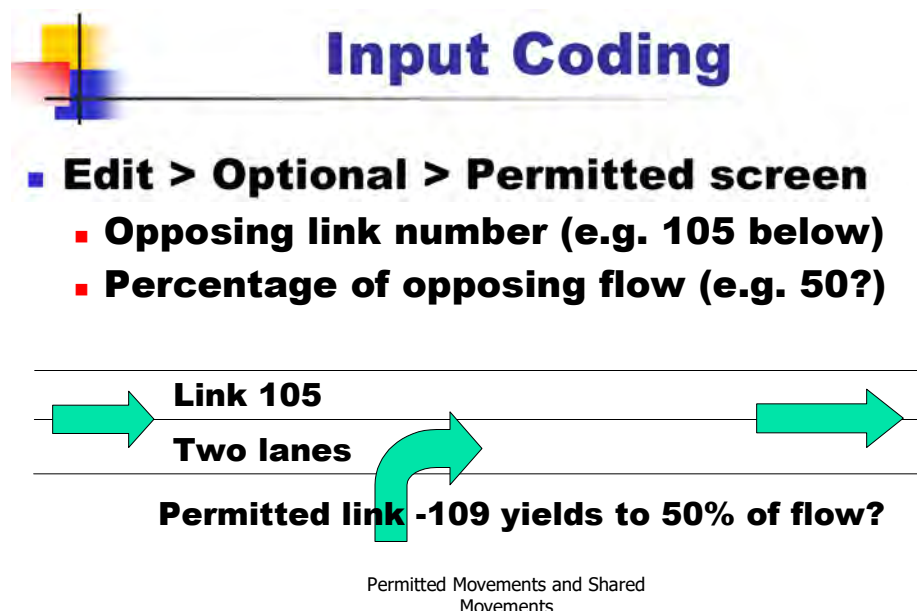
Also when creating a brand-new input file using the GUI, "conflicting" link numbers are

automatically coded. If right-hand drive is in effect, conflicting links for the right-turn movements are automatically coded. If left-hand drive is in effect, conflicting links for the left-turn movements are automatically coded. Opposing flows for these conflicting links are automatically coded as 100%. If there are two through lanes, an opposing flow of 50% might be coded by the user, or 33% for three through lanes, etc.

Note that only non-permitted, unopposed links may be specified as opposing links. The software only exercises the permitted movement model on opposed links versus unopposed links.

#### Opposing Flow (%)

The subject link must yield to a percentage of the opposing flow, which is specified here. For example, assuming right-hand drive, a permitted right-turn movement sometimes does not yield to 100% of the conflicting through volume if there are multiple through lanes. A permitted right-turn lane may also be impeded by an opposite protected left-turn. Generally for permitted left-turns, the only opposing link will be the opposing through link, and 100% of the opposing link's traffic will apply. An exception to this would be an opposing right-turn moving from exclusive right-turn lane, for which the percentage of opposing flow might depend on driver behavior, or the number of receiving lanes. The figure below illustrates a permitted right-turn (represented by link 109) that only yields to traffic in one of the two conflicting through lanes (represented by link 105).



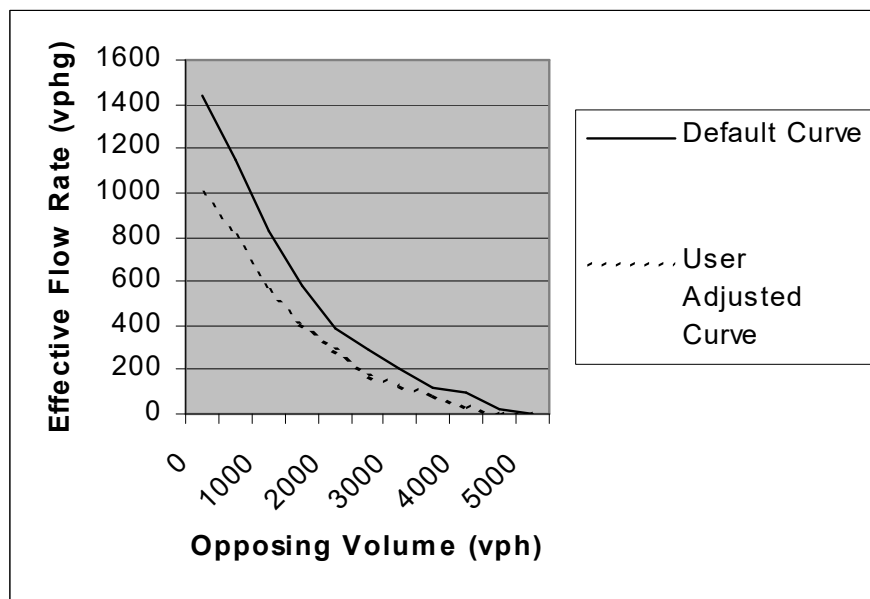
There might be a tendency to try to estimate the percentage of the opposing flow that impedes the permitted link, as a function of the phasing sequence. Do not do this. TRANSYT automatically determines when a permitted link is opposed.

If the node is an unsignalized intersection, and a permitted link yields to 100% of exactly one link's flow, then the permitted link is assumed to be a major-street left-turn movement (assuming right-hand drive).

Permitted Maximum Flow Rate (vph) (press Shift-Page Down or Next Data to access new data)

This represents the maximum flow rate of vehicles that could be discharged during the permitted portion of the phase. Normally the flow rate for permitted movements is computed according to the default permitted movement model parameters, but this parameter allows for calibration, as do the Model Coefficients parameters (on the Edit > Optional > Global > Model Coefficients screen or record type 8).

The replacement value of the maximum flow rate for unprotected, unopposed turns, during the permitted phase(s) has the effect of shifting the default permitted movement model curve "up or down", depending on the value (with respect to the default value). The figure below illustrates the effect of coding a significantly lower value (e.g., 1000 vph) in lieu of the default value (1436 vph).



The Permitted Maximum Flow Rate represents the saturation flow rate when there are no opposing vehicles, but the permitted turning vehicles don't know that there are no opposing vehicles. They must hesitate to confirm that there is a gap to move through. This differs from the standard Saturation Flow Rate, in which vehicles would move through the intersection without hesitation. The Permitted Maximum Flow Rate should be calibrated for dual permitted turn lanes, since the default value assumes a single turn lane.

Sneakers per Cycle (press Shift-Page Down or Next Data to access new data)

Following the permitted green phase, a few extra vehicles may "sneak" through the intersection during the yellow and all-red clearance intervals. TRANSYT provides the saturation flow rate for these vehicles, such that the number of sneakers specified can be served if they are present at that point in the cycle.

The sneakers will apply only to a permitted, opposed movement at the end of a permitted phase that is not followed by a protected green phase. The value coded here will be ignored if a protected phase follows the permitted phase. Assuming right-hand drive, it is acceptable to code a value for sneakers to be used in a later run with a leading left-turn sequence, to compare with a run made with a lagging left-turn sequence.

Due to the record type file format, the Permitted screen automatically stores data in integer format. For example, if a value of 1.5 sneakers is coded on the Permitted screen, then a value of '15' is written onto record type 29.

#### Permitted Model Override (press Shift-Page Down or Next Data to access new data)

The network-wide default permitted movement model, applied by TRANSYT at signalized intersections, is the ARRB model [27]. At unsignalized intersections, TRANSYT uses internal logic to estimate the appropriate permitted movement model to be applied to the traffic movement (link) in question.

This input field is used to override the default permitted movement model number selected by TRANSYT, but only for this particular traffic movement. For example, to model permitted right-turns (assuming right-hand drive) at a signalized intersection, the right-turn permitted movement model number (18) would be specified. To change the default network-wide permitted movement model, it would be necessary to make this change on the Edit > Optional > Global > Model Coefficients screen (or record type 8).

When creating a brand-new input file using the GUI, the right-turn-on-red (RTOR) model number (18) is automatically coded for certain links. If right-hand drive is in effect, the RTOR model is automatically coded for right-turn links. If left-hand drive is in effect, the RTOR model is automatically coded for left-turn links. If the turning movement moves with protection during all phases, and is never permitted, the RTOR model will never be exercised (even if it was coded).

For the ARRB model, the software estimates the number of opposing lanes for permitted movements. Estimation of the number of opposing lanes is based on the opposing links' saturation flow rates. A weighted average, based on the coded percentages of opposing flow, is used to compute the opposing saturation flow rate of up to three opposing links. If the opposing saturation flow rate is lower than 2000 vehicles per hour of green (vphg), then one opposing lane is assumed. If the opposing saturation flow rate is between 2000 and 4400 vphg, then two opposing lanes are assumed. If the opposing saturation flow rate is greater than 4400 vphg, then three opposing lanes are assumed.

To override the estimated number of opposing lanes for the ARRB model, the number of opposing lanes must be coded as a negative value. For example, to specify three opposing lanes regardless of the opposing saturation flow rates, a value of '-3' would be coded in the Permitted Model Override field. Other permitted movement model numbers are listed in the table below, which assumes right-hand drive.

Model Type	Speed (mph):	1 Lane		2+ Lanes		Global No.
		<40	>=40	<40	>=40	
ARRB Model			0			0
FHWA/TRC Model (all)						10
Permitted-only left-turn		11	11	12	13	
Protected + permitted left-turn		14	14	15	16	
Unsignalized left-turn (artery)		17	17	17	17	
RTOR & unsig. X-street thru		18	18	18	18	
Unsignalized X-street left-turn		19	19	19	19	
Texas A&M Model						
Field study parameters (all)			21			21
Simulation parameters (all)			22			22
Minnesota Model		31		32		30
Ohio State Model		41		42		40
HCM 1985 Model (all)			50			50

Refer to the Permitted Movement Methodology section of the documentation for a detailed description of the permitted movement modeling procedure.

## **Weighting Screen (associated with record types 30-39)**

All weighting or parameter modifier records are optional. Note that any given link should not be modified more than once on any individual weighting record. The following general rules apply to all weighting records:

1. They must appear in ascending numerical order (31, 32, 33... 39) in the record type data file.
2. Blank fields are ignored. There is no significance to the order of the links listed.
3. If a weighting factor is specified but all other entries (the link list) are blank, then the weighting factor applies to all links in the network. The exception to this is Platoon Dispersion Factors (record type 39), on which replacement values instead of multipliers are specified.

### **Performance Index Weighting (associated with record type 30)**

The multiplier here, when divided by 100, multiplies the performance index calculation either globally, or for the specified list of links. This input data can be useful when attempting to emphasize or de-emphasize certain links during the optimization process. The acceptable range for the multiplier is 0 to 9999.

### **Link-to-Link Flow Weighting (associated with record type 31)**

To encourage link-to-link progression in the optimization process, this input data establishes specific combinations of links for which special emphasis is applied. This is accomplished by double-counting the disutility index components for all links indicated in the progressive sequence.

If the list of links, to which the feeding link applies, is left blank, the feeding link number is considered global in order to prioritize all through links in a particular direction.

It is possible to overextend the optimization process if this and other weighting factors are overused. It is suggested that this capability be only used for very important routes within the network.

### **Queuing Ratio Weighting (associated with record type 32)**

One of the objective functions that can be selected for optimization is the product of the disutility index and the queuing ratio. This performance index attempts to minimize the disutility index while considering the extension of the back of queue on critical links. These critical links are specified here.

The exponent coded here, when divided by 100, is applied to the queuing ratio (back of queue divided by queuing capacity). The acceptable range for the multiplier is 1 to 9999. This input data should not be used unless the product of the disutility index and the queuing ratio has been selected as the performance index.

### **Throughput Weighting (associated with record type 33)**

Three of the objective functions recommended for optimization of oversaturated conditions involve maximizing throughput on critical links, while maintaining good performance on other links in the system. The critical links for which throughput needs to be maximized are specified here. These links should be those expected to experience a decrease in capacity due to queue blockage effects, either from downstream or adjacent links.

The weight coded here, when divided by 100, is the exponent either applied globally, or to the list

of links. The acceptable range for the multiplier is 0 to 99999. This input data should not be used unless the the performance index includes throughput.

#### **Queuing Capacity Multiplier** (associated with record type 34)

TRANSYT estimates the queuing capacity for each link. The user may optionally override the internally estimated queuing capacity on the Edit > Optional > Queuing screen (or record type 28 field 16). Queuing capacity affects the simulation of queue blockage, and potentially affects the performance index during optimization.

The weight coded here, when divided by 100, is either applied globally, or to the list of links. The acceptable range for the multiplier is from 20 to 100.

#### **Speed Multiplier** (associated with record type 35)

The weight coded here, when divided by 100, either multiplies the average cruise speeds or travel times globally, or for the list of links. The acceptable range for the multiplier is from 50 to 200. Separate records should be used in order to affect bus speeds only, when applicable.

#### **Volume Multiplier** (associated with record type 36)

The volume multiplier may be used to factor up volumes for future years, special events, or sensitivity analysis. The weight coded here, when divided by 100, either multiplies the traffic volumes globally, or for the list of links. The acceptable range for the multiplier is from 5 to 2000.

#### **Delay Weight Multiplier** (associated with record type 37)

The weight coded here, when divided by 100, either multiplies the effect of delay within the disutility index globally, or for the list of links. The range is from 0 to 9999. Specifying a factor greater than 100 causes special preference to be given to the applicable list of links, during the optimization process.

If the factor is global, the disutility index will place more relative importance on delay rather than stops or fuel consumption. Also note that the stop penalty specified on the Edit > Optional > Global > Run Controls screen (or record type 1) can be used instead to automatically indicate that the disutility index coefficients are designed to 1) minimize excess fuel consumption, or 2) consider minimizing delay only, or 3) place extra emphasis on minimizing stops.

If the weighting factor is coded as '0', then the link's delay calculation will not be included in the calculation of the network-wide disutility index. These links are also excluded from all network-wide totals in the output performance tables, and fuel consumption is not calculated for these links. Coding the weighting factor as '0' here is recommended primarily for pedestrian or non-vehicular links.

#### **Stop Weight Multiplier** (associated with record type 38)

The weight coded here, when divided by 100, either multiplies the effect of stops within the disutility index globally, or for the list of links. The acceptable range for the multiplier is from 0 to 9999. Specifying a factor greater than 100 causes special preference to be given to the applicable list of links, during the optimization process. The multiplier here is a further adjustment to the stop penalty on the Edit > Optional > Global > Run Controls screen (record type 1). For example, to further multiply the effect of stops by five, code '500'. If '4' was coded as the stop penalty on the Edit > Optional > Global > Run Controls screen (record type 1), the net effect would be to multiply the stops calculation by  $4 \times 5 = 20$ .

If the factor is global, the disutility index will place more relative importance on stops rather than delay or fuel consumption. Also note that the stop penalty specified on the Edit > Optional > Global > Run Controls screen (or record type 1) can be used instead to automatically indicate that the disutility

index coefficients are designed to 1) minimize excess fuel consumption, or 2) consider minimizing delay only, or 3) place extra emphasis on minimizing stops.

If the weighting factor is coded as '0', then the link's stops calculation will not be included in the calculation of the network-wide disutility index. These links are also excluded from all network-wide totals in the output performance tables, and fuel consumption is not calculated for these links. Coding the weighting factor as '0' here is recommended primarily for pedestrian or non-vehicular links.

### **Platoon Dispersion Factors** (associated with record type 39)

The platoon dispersion factor affects the simulated flow profile, which in turn affects traffic flow and performance throughout the network. This screen allows the user to specify platoon dispersion factors on individual links that are different from the global default value. These values are replacement values as opposed to adjustment values. For example, to change the platoon dispersion factor to 0.4, code '40'. The acceptable range for the replacement factors is from 0 to 100. A value of '0' indicates no platoon dispersion. This differs from the global platoon dispersion factor, specified on the Edit > Optional > Global > Network Parameters screen (record type 10), which does not permit a coded value of '0' indicating no platoon dispersion.

## **Saturation Flow Rate Calculator (associated with record types 28 and 292)**

This input screen was implemented to assist the user in determining adjusted saturation flow rates. As explained later, some adjustment factors are computed and displayed in exact accordance with Highway Capacity Manual (HCM2000) [12] procedures, whereas other adjustment factors are simply approximated. Detailed guidelines for analytical estimation of saturation flow rates are summarized in HCM chapter 16, page 11. If the user already knows the adjusted saturation flow rates, they can be coded directly on the Edit > Traffic screen, and this screen can be skipped.

Saturation flow rates are used by the macroscopic simulation model to affect the departure of queues from a link. Along with the volume, travel time between nodes, number of lanes, and timing plan, saturation flow rate is one of the most important factors in determining traffic network performance. Because of this, it is important that reasonable estimates of the saturation flow rate are provided by the user.

The Satflow Flow Rate Calculator (Edit > Optional > Capacities) is illustrated below. This screen can be sent to the printer, or to an image file, by pressing Ctrl-P. Also note that a summary text report of saturation flow rate adjustment factors, called "Node Satflow Summary", can be accessed from the Output Quick Jump. The Node Satflow Summary report only contains information if adjustment factors have been saved into the TRANSYT-7F input (\*.TIN) file (record type 292) by pressing the Apply button. The Satflow View screen, which is described earlier in this chapter, can be accessed by selecting Edit > Satflow.

As shown at the top of the screen, the adjusted saturation flow rate is simply the product of the base saturation flow rate, the number of lanes, and eleven "adjustment factors". The Stored Value displayed on the right is the current value within the TRANSYT-7F input (\*.TIN) file. Clicking on the Apply button will cause the calculated value from this screen (in the white data field) to be saved into the TIN file, thus becoming the new Stored Value. The Headway is the average time spacing between vehicles in any given lane, based on the Stored Value. For thru movements, this is also the link-specific mean discharge headway that would be coded into a CORSIM input (\*.TRF) file.

The blue arrow (directly above the Apply button) was added to clarify the turning movement associated with the active link number. This arrow is only displayed for link numbers within the link numbering scheme.



**7F Saturation Flow Rate Calculator**

$$S = S_o * N * F_w * F_{hv} * F_g * F_p * F_{bb} * F_a * F_{lu} * F_{lt} * F_{rt} * F_{rt}$$

$$= 1900 * 3.00 * 0.97 * 0.95 * 1.01 * 1.00 * 1.00 * 1.00 * 0.95 * 0.95 * 1.00$$

Base Saturation Flow Rate (vplphg)	<input type="text" value="1900"/>	
Number of Lanes	<input type="text" value="3"/>	Stored Value (vplphg)
Lane Width (ft)	<input type="text" value="11"/>	Headway (sec)
Heavy Vehicles (%)	<input type="text" value="5"/>	
Grade (%)	<input type="text" value="-2"/>	Link Number
Parking Maneuvers Per Hour	<input type="text" value="0"/>	Node Number
	<input checked="" type="checkbox"/> Parking Prohibited	
Number of Buses Stopping	<input type="text" value="0"/>	
Area Type	<input type="text" value="Other"/>	
Lane Utilization	<input type="text" value="Slightly unequal"/>	

Techniques for field measurement of saturation flow rates are discussed on HCM chapter 16, page 158. Time permitting, it is always advisable to field-measure saturation flow rates at a few key locations in the network. Although the HCM recommends a "base" (unadjusted) saturation flow rate of 1900 vehicles per lane per hour of green (vplphg), this value is not likely to be realistic for all cities or city sections. If there is a consistent and predictable discrepancy between field-measured and HCM-estimated saturation flow rates, this might indicate that an adjustment to the base saturation flow rate is warranted. Once the user has calibrated the base saturation flow rate for a certain jurisdiction, the HCM analytical estimation procedures could then be used for adjusting the updated base saturation flow rate, at all other (non-field-measured) network locations.

In the Traffic Screen section of the documentation, rule-of-thumb (non-HCM) suggested values for adjusted saturation flow rates are available. The most efficient method of data entry might depend on the number of adjustments necessary. If conditions are fairly simple, adjusted saturation flow rates could be estimated using the general guidelines listed in the rule-of-thumb table. If numerous saturation flow adjustments are necessary for the protected movements, it might be easiest to use the Satflow View screen, or the HCS file import feature (which automatically imports the adjusted saturation flow rates).

This input screen does not estimate adjusted saturation flow rates for permitted-only links, or for secondary shared links, because any saturation flow rates coded for these links would be ignored by the model. For links representing permitted turning movements, the saturation flow rate is significantly lower, since these vehicles must wait for gaps in traffic on the opposing links. Saturation flow rate determination is accomplished automatically by the internal permitted movement model. The "maximum flow rate" (MFR) for periods when the permitted movement is unopposed may be adjusted (from the internal default value) on the Edit > Optional > Permitted screen. For protected-permitted turns, the saturation flow rate to be used during only the protected phase can be estimated on this screen. If multiple links are specified

as being shared (Edit > Optional > Global > Shared Lanes screen, or record type 7), the screen only provides a value for the primary link, and this saturation flow rate applies collectively to all shared-lane links.

As stated above, The Saturation Flow Rate Calculator does not estimate adjusted saturation flow rates for permitted-only links, or for secondary shared links, because any user-coded saturation flow rates for these links would be ignored by the model. Because of this, the Link Number combo box will only be populated with protected, primary links.

The following adjustment factor calculations are taken directly from the HCM:

$$f_w = 1 + \frac{(W - 12)}{30}, \text{ where } W \text{ is the lane width in feet}$$

$$f_{HV} = \frac{100}{100 + \%HV(E_T - 1)}, \text{ where } E_T \text{ equals 2.0 passenger cars per heavy vehicle}$$

$$f_g = 1 - \frac{\%G}{200}, \text{ where negative is downhill}$$

$$f_p = \frac{N - 0.1 - \frac{18N_m}{3600}}{N}, \text{ where } f_p \geq 0.05, \text{ and } f_p = 1.00 \text{ for no parking}$$

$$f_{bb} = \frac{N - \frac{14.4N_B}{3600}}{N}, \text{ where } f_{bb} \geq 0.05$$

$$f_a = 0.90 \text{ in CBD, } 1.00 \text{ in all other areas}$$

$$f_{LT} = 0.95 \text{ for exclusive left turn lanes, if right hand drive}$$

$$f_{RT} = 0.85 \text{ for exclusive right turn lanes, if right hand drive}$$

When implementing these adjustments, left-hand drive is handled differently than right-hand drive. For example, in the case of an exclusive right-turn lane under left-hand drive, the adjustment factor becomes 0.95, and is listed as the right-turn adjustment factor. Metric units are also handled differently for the lane width adjustment factor. Lane widths are accepted in meters, with decimal points allowed. The adjustment factor should be the same for equivalent English and metric lane widths.

The lane utilization adjustment factor is handled differently by the Saturation Flow Rate Calculator, relative to the HCM. The combo box allows for selection of "equal", "slightly unequal", "moderately unequal", and "highly unequal" vehicle utilization of each lane. These selections result in adjustment factor values of 1.00, 0.95, 0.90, and 0.85, respectively. If a value of '1' is present in the Number of Lanes field, the lane utilization combo box is automatically disabled, and the lane utilization adjustment factor is assumed to be 1.00. If a value greater than '1' is present in the Number of Lanes field, and the user has not yet coded a lane utilization, the default lane utilization adjustment factor is assumed to be 0.95 ("slightly unequal").

The lane utilization adjustment factor computation from the HCM is as follows:

$$f_{LU} = \frac{v_g}{v_{g1}N}$$

where,

vg = unadjusted demand flow rate for the lane group, vph

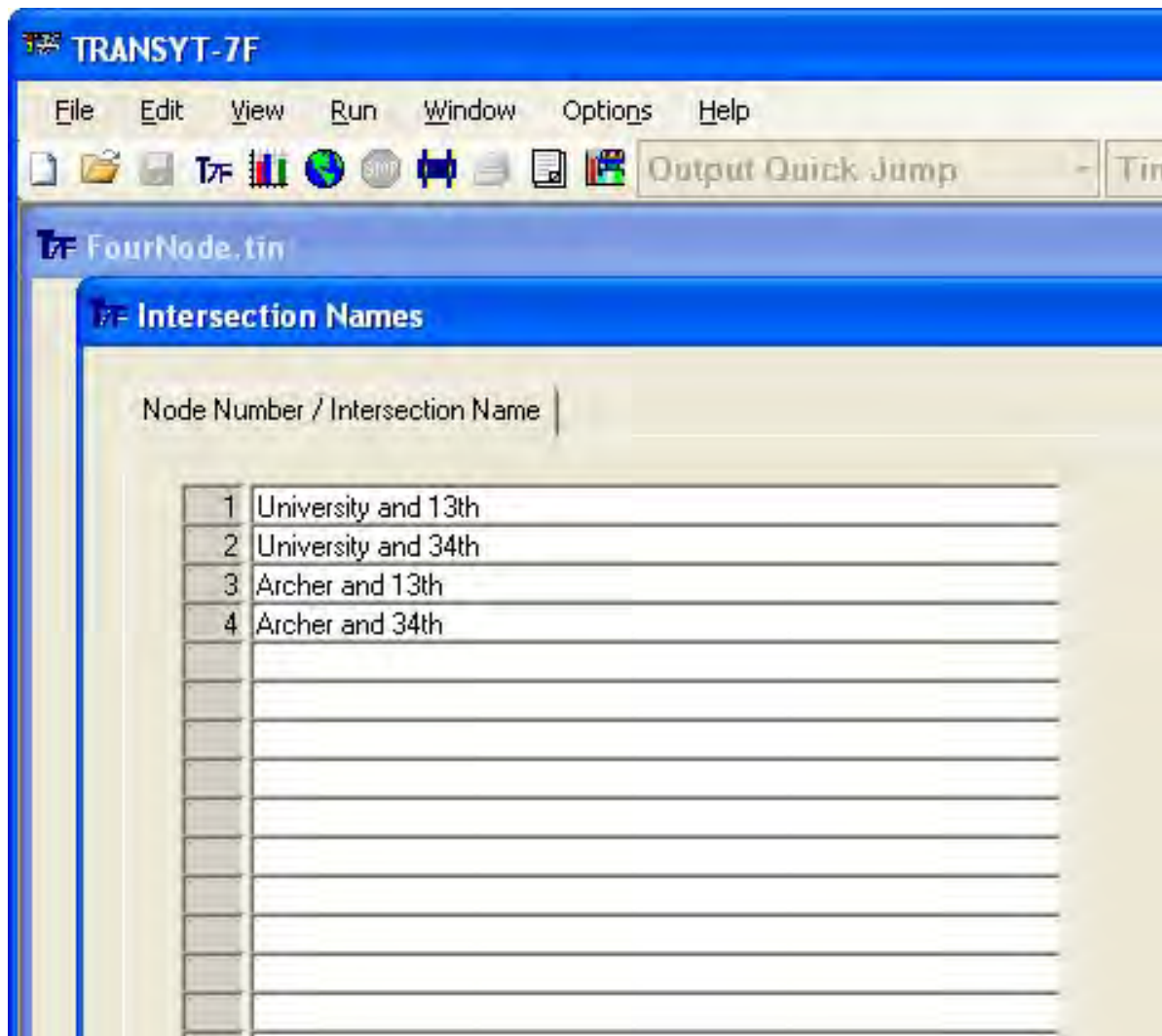
vg1 = unadjusted demand flow rate on the single lane in the lane group with the highest volume

N = number of lanes in the lane group

The pedestrian-bicycle blockage adjustment factors for left-turns and right-turns are also handled differently by the Saturation Flow Rate Calculator, relative to the HCM. The combo boxes allow for selection of "none", "light", "medium", and "heavy" pedestrian-bicycle blockage of the subject link. These selections result in adjustment factor values of 1.00, 0.95, 0.90, and 0.85, respectively. Through links should only be affected by pedestrian-bicycle blockage if they are primary links within a shared lane group, due to impedance of the turning vehicles. The HCM computation of the pedestrian-bicycle blockage adjustment factor is fairly detailed, and readers are referred to the HCM to review these details.

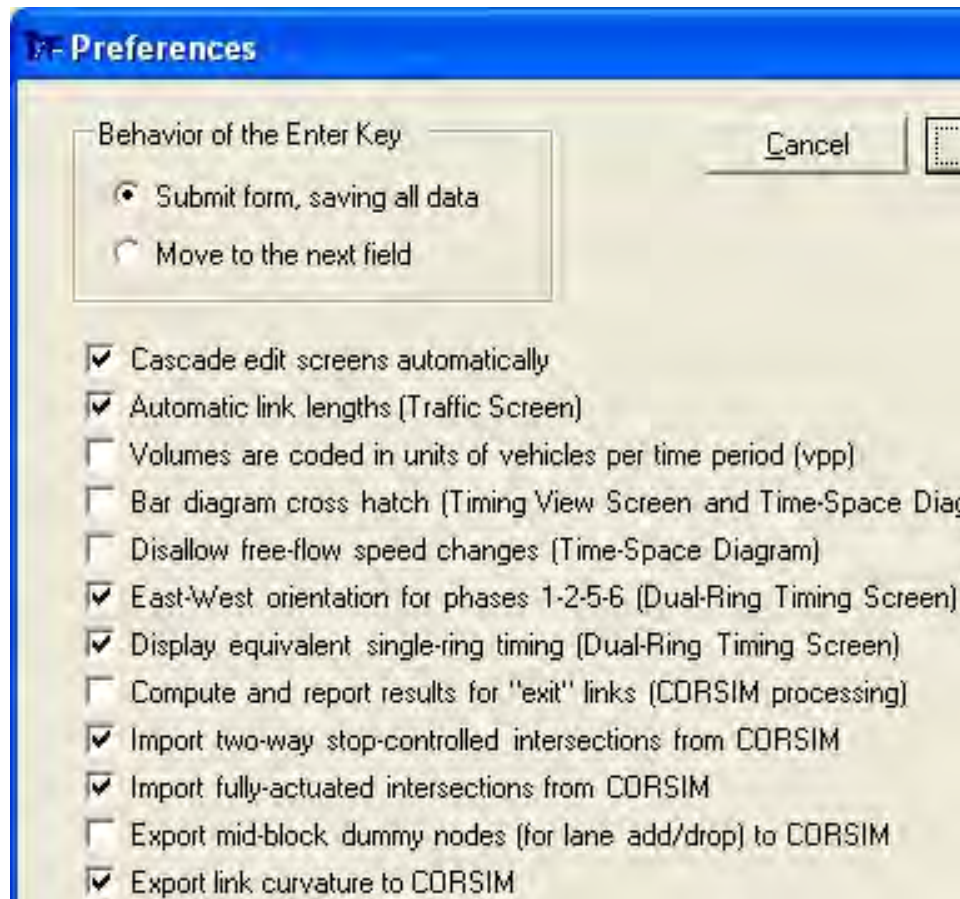
## Intersection Names Screen

Intersection names may optionally be coded by the user on the Edit > Optional > Intersection > Names screen. Subsequently, these names will be displayed at the top of the TRANSYT-7F node-specific edit screens, as well as certain output reports. Intersection names may be exported into the link performance tables of existing legacy reports, which have the \*.TOF extension, provided that the user clicks on the "Insert into TOF" button. Intersection names can be displayed in the Map View by resting the mouse cursor on top of any particular node. A sample Intersection Names screen is illustrated below.



## Preferences Screen

The Options > Preferences screen from TRANSYT-7F is illustrated below.



#### Behavior of the Enter Key (applicable to most screens within the TRANSYT-7F Edit menu)

The standard behavior for the Enter key is to "submit" the form or window, while saving all data. However, some users may prefer to switch between numerical edit fields by pressing the Enter key.

#### Cascade edit screens automatically

By default, secondary input screens on the Edit menu are "cascaded" for better organization. This has the same effect of selecting Window > Cascade. However, with certain screen resolutions, it is sometimes preferable to be able to view multiple Edit screens simultaneously. Cascading the Edit screens can make this more difficult.

Primary Edit screens (Lanes, Traffic, Timing, Satflow, Feeders) are designed to be accessed from the Map View. When right-clicking on a node in the Map View, these screens pop up as dialog boxes above the Map. Therefore, these screens are independent from the secondary Edit screens, and cannot be cascaded automatically.

#### Automatic link lengths (applicable to the Traffic screen)

In TRANSYT, all link lengths on the same approach should have equal length, in order to maintain the correct simulation of platoon dispersion. Short turn bays or turn pockets are optionally simulated by coding the appropriate queuing capacity. Link numbers normally associated with the same approach could occasionally have unequal lengths if they are being used to model movements on different approaches, or if the stopline-to-stopline distance of individual turning movements were significantly different.

Because all link lengths on the same approach should usually have equal length, using the "automatic link length" feature is recommended for speeding up the data entry process. When the user codes a link length, the program will automatically code the same value for the other existing movements on that approach. If the user creates a new link by changing a volume from 0 to something positive, the link length for that new movement will automatically be coded with the same link length as a previously existing movement on the same approach.

#### Volumes are coded in units of vehicles per time period (vpp)

By default, traffic volumes in TRANSYT-7F are coded in units of vehicles per hour (vph). If this check box is activated, traffic volumes will instead be interpreted in units of vehicles per time period (vpp). This feature is possibly useful if the known volumes are not available units of vehicles per hour. However, selection of the analysis period duration (on the Edit > Analysis screen, or record type 1) has an important effect on modeling, and should not necessarily be governed by data collection techniques. The default analysis period duration (for a single-period simulation) is 15 minutes.

If this check box is activated, labels on the Lanes, Traffic and Feeders screens will reflect "vpp" instead of "vph". Any currently loaded data file will be flagged (with a negative sign on record type 1, field 12) to indicate vehicle per time period coding. Simulation model results will also be affected, if the analysis period duration is not equal to 60 minutes. The legacy output report, optimization reports, and summary text reports will display certain results "per time period" instead of "per hour". If this check box has been activated recently, the "New File" dialog screen will also reflect and generate volumes in units of vehicles per time period. If the user wants to change their preference prior to generating a new file, this check box may also be modified while no data file is currently loaded.

#### Bar diagram cross hatch (applicable to the Timing View and Time-Space Diagram screens)

By default, bar diagrams are displayed using only the colors green, yellow, and red, to distinguish between the different signal intervals. The "cross hatch" option provides another mechanism besides color for distinguishing between the different signal intervals. When the cross hatch option is activated, yellow intervals are drawn using a single diagonal line as the fill pattern, in addition to the yellow coloring. All-red intervals are drawn using double diagonal lines as the fill pattern, in addition to the red coloring. This setting also affects both bar diagrams at the bottom of the Dual-Ring Timing screen.

#### Disallow free-flow speed changes (applicable to the Time-Space Diagram screen)

In cases where the cruise speed does not remain constant along the progression route, this option exists to use the exact speeds that were coded in the data file. When "Disallow free-flow speed changes" is selected, speeds cannot be changed directly on the Time-Space Diagram screen, and only "cumulative" bandwidths can be displayed (as opposed to "constant" bands).

#### East-West orientation for phases 1-2-5-6 (Dual-Ring Timing Screen)

#### Display equivalent single-ring timing (Dual-Ring Timing Screen)

For a discussion of these functions, refer to the Dual-Ring Timing screen section of the documentation.

#### Compute and report results for "exit" links (applicable to CORSIM processing)

#### Import two-way stop-controlled intersections from CORSIM

#### Import fully-actuated intersections from CORSIM

#### Export mid-block dummy nodes (for lane add/drop) to CORSIM

#### Export link curvature to CORSIM

#### Export dual-ring timing (for pre-timed signals) to CORSIM

#### Export right-turns on red (RTOR) to CORSIM

#### Export heavy vehicles to CORSIM (%)

For a discussion of these functions, refer to the CORSIM Processing Methodology and One-Touch CORSIM Animation sections of the documentation.

---

## Record Type Format

### Record Type Format

The TRANSYT-7F input file format is based on 80-column "record types". In the current version of TRANSYT-7F, the user is not required or expected to edit the record types directly. Instead, the user can code data into the more intuitive Windows edit screens, and the software will update the record types in the background automatically. However, to view or edit record types directly, the record type editor can be used.

Different record types are used to describe specific categories of data. For example, record type 1 specifies the Run Controls, whereas record type 28 specifies Link Data. Some record types are required within the data file, whereas some record types are optional. For example, record type 10 (Network Parameters) is required, but record type 36 (Volume Multiplier) is optional. In the spreadsheet-style record type documentation that follows, clarification is provided regarding whether individual record types are required or optional. Similarly, the TRANSYT-7F Edit menu contains an extra "Optional" menu, to assist the user in determining which data are required and which are not.

Some record types are only allowed to appear once within the data file, whereas other record types may be repeated several times. For example, record type 1 (Run Controls) can appear only once, whereas record type 28 (Link Data) must appear repeatedly based on the number of links in the network. The program will issue a fatal error message if record types are duplicated inappropriately, or are listed in the wrong sequence. These record type "duplicity" and "sequence" formatting requirements are no longer fully documented, but can be determined by using the TRANSYT-7F edit screens to generate various data files.

Most record types containing a non-numeric character in column 1 (on the left-hand side) are considered "comment records", and are ignored by the software. The exceptions to this rule of thumb are the Run Title, Route Title, Intersection Names, and Link Names, which are indeed used by the program. In fact, the Run Title Record is required to be the first record type within the data file. Any record type containing a blank space in column 1 is considered numeric modeling data to be processed by the program.

When adding or deleting information using TRANSYT-7F, any existing comment records are subject to being shifted within the data file. However, the graphical user interface (GUI) attempts to preserve the logical location of the following comment records:

```
--- Run Controls ---
--- Node List ---
--- Link Numbering Scheme ---
--- Shared Lanes ---
--- Model Coefficients ---
--- Network Parameters ---
*** Intersection 1 ***
--- Signal Timing ---
--- Link Data ---
--- Arterial Route Specifications ---
--- Run Specifications ---
```

```

--- Route Parameters ---
--- Node Coordinates ---
--- Termination ---

```

Record types containing valid data are typically subdivided into 16 data fields, each one of which is 5 characters wide. Field 1 (i.e., columns 1 through 5) typically indicates the record type number. Numerical values are right-justified within each data field. For example, record type 36 would appear as "36" within columns 1 through 5. Record types containing valid data are not allowed to contain letter characters or decimal points; only numerical values, and sometimes negative signs, are accepted. Non-integer values are typically coded as multiples of 10, to avoid using the decimal point. For example, a value such as "2.75" would often be coded as "275".

If a zero or a blank is coded in any given field, this sometimes causes the program to use a default value, but other times causes the program to use a value of zero. In the spreadsheet-style record type documentation that follows, the allowable data ranges and default values for each field are listed. If a numerical coded value is out of range for any particular field, the program will likely issue a fatal error message. The TRANSYT-7F edit screens are helpful, because they often do not allow the user to type out-of-range values. However, it is sometimes possible for the program to issue fatal error messages even when all of the individual values are within range. These errors are called "processing errors", and they are used to indicate data values that do not agree with one another. For example, if the user codes interval durations that conflict with the minimum phase times or the cycle length, a processing error may occur.

The sample file "FourNode.tin" from the TRANSYT-7F Example folder is illustrated below, in record type format.

#### Four Node Sample File

```

--- Run Controls ---
 1 100 100 10 0 1 2 2 -1 1 0 15 0 0 0 0
--- Node List ---
 2 1 2 3 4 0 0 0 0 0 0 0 0 0 0 0
--- Link Numbering Scheme ---
 3 2 1 9 4 3 10 6 5 11 8 7 12
--- Shared Lanes ---
 7 101 109 0 0 0 103 110 0 0 0 0 0 0 0 0
 7 105 111 0 0 0 107 112 0 0 0 0 0 0 0 0
 7 201 209 0 0 0 203 210 0 0 0 0 0 0 0 0
 7 205 211 0 0 0 207 212 0 0 0 0 0 0 0 0
 7 301 309 0 0 0 303 310 0 0 0 0 0 0 0 0
 7 305 311 0 0 0 307 312 0 0 0 0 0 0 0 0
 7 401 409 0 0 0 403 410 0 0 0 0 0 0 0 0
 7 405 411 0 0 0 407 412 0 0 0 0 0 0 0 0
--- Network Parameters ---
10 1 4 1 1900 35 35 0 0 0 0 0 0 0 0 0
*** Intersection 1 ***
--- Signal Timing ---
14 1 0 5 10 4 1 25 4 1 10 4 1 35 410000
18 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
21 1 1 1 2 3 10 0 0 0 0 0 0 0 0 1
21 1 0 0 0 0 0 106 0 0 108 0 0 0 0 2
22 1 4 4 5 6 10 0 0 0 0 0 0 0 0 0
22 1 0 0 0 0 0 0 105 111 0 107 112 0 0 2
23 1 7 7 8 9 10 102 0 0 104 0 0 0 0 1
23 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2
24 1 10 10 11 12 10 0 101 109 0 103 110 0 0 1
24 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2
--- Link Data ---
28 101 1320 1900 400 0 0 0 0 0 0 0 0 0 0 0
29 101 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```



```

291 101 102 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 102 1320 1805 50 0 0 0 0 0 0 0 0 0 0 12
29 102 0 0 0 10 0 0 0 103 100 0 0 0 0 0
291 102 0 0 101 0 1 1 1 1 1 1 0 0 3009003010000
28 103 1320 1900 400 0 303 300 35 308 50 35 311 50 35 0
29 103 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 103 104 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 104 1320 1805 50 0 303 50 35 0 0 0 0 0 0 12
29 104 0 0 0 10 0 0 0 101 100 0 0 0 0 0
291 104 0 0 103 0 1 1 1 1 1 1 0 0 3009003010000
28 105 1320 1900 400 0 0 0 0 0 0 0 0 0 0 0
29 105 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 105 106 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 106 1320 1805 50 0 0 0 0 0 0 0 0 0 0 12
29 106 0 0 0 10 0 0 0 107 100 0 0 0 0 0
291 106 0 0 105 0 1 1 1 1 1 1 0 0 3009003010000
28 107 1320 1900 400 0 207 300 35 202 50 35 210 50 35 0
29 107 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 107 108 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 108 1320 1805 50 0 207 50 35 0 0 0 0 0 0 12
29 108 0 0 0 10 0 0 0 105 100 0 0 0 0 0
291 108 0 0 107 0 1 1 1 1 1 1 0 0 3009003010000
28 109 1320 0 50 0 0 0 0 0 0 0 0 0 0 0
29 109 0 0 0 0 0 0 0 0 105 100 0 0 0 0 18
291 109 0 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 110 1320 0 50 0 303 50 35 0 0 0 0 0 0 0
29 110 0 0 0 0 0 0 0 0 107 100 0 0 0 0 18
291 110 0 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 111 1320 0 50 0 0 0 0 0 0 0 0 0 0 0
29 111 0 0 0 0 0 0 0 0 103 100 0 0 0 0 18
291 111 0 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 112 1320 0 50 0 207 50 35 0 0 0 0 0 0 0
29 112 0 0 0 0 0 0 0 0 101 100 0 0 0 0 18
291 112 0 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000

```

\*\*\* Intersection 2 \*\*\*

--- Signal Timing ---

```

14 2 0 5 10 4 1 25 4 1 10 4 1 35 4 0
18 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0
21 2 1 1 2 3 10 0 0 0 0 0 0 0 0 1
21 2 0 0 0 0 0 0 206 0 0 208 0 0 0 2
22 2 4 4 5 6 10 0 0 0 0 0 0 0 0 0
22 2 0 0 0 0 0 0 205 211 0 207 212 0 0 2
23 2 7 7 8 9 10 202 0 0 204 0 0 0 0 1
23 2 0 0 0 0 0 0 0 0 0 0 0 0 0 2
24 2 10 10 11 12 10 0 201 209 0 203 210 0 0 1
24 2 0 0 0 0 0 0 0 0 0 0 0 0 0 2

```

--- Link Data ---

```

28 201 1320 1900 400 0 0 0 0 0 0 0 0 0 0 0
29 201 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 201 202 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 202 1320 1805 50 0 0 0 0 0 0 0 0 0 0 12
29 202 0 0 0 10 0 0 0 203 100 0 0 0 0 0
291 202 0 0 201 0 1 1 1 1 1 1 0 0 3009003010000
28 203 1320 1900 400 0 403 300 35 408 50 35 411 50 35 0
29 203 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 203 204 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 204 1320 1805 50 0 403 50 35 0 0 0 0 0 0 12
29 204 0 0 0 10 0 0 0 201 100 0 0 0 0 0
291 204 0 0 203 0 1 1 1 1 1 1 0 0 3009003010000
28 205 1320 1900 400 0 105 300 35 104 50 35 109 50 35 0
29 205 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 205 206 99 0 0 0 0 0 0 0 0 0 0 3009003010000

```

```

28 206 1320 1805 50 0 105 50 35 0 0 0 0 0 0 12
29 206 0 0 0 10 0 0 0 207 100 0 0 0 0 0
291 206 0 0 205 0 1 1 1 1 1 0 0 3009003010000
28 207 1320 1900 400 0 0 0 0 0 0 0 0 0 0
29 207 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 207 208 99 0 0 0 0 0 0 0 0 0 3009003010000
28 208 1320 1805 50 0 0 0 0 0 0 0 0 0 12
29 208 0 0 0 10 0 0 0 205 100 0 0 0 0 0
291 208 0 0 207 0 1 1 1 1 1 0 0 3009003010000
28 209 1320 0 50 0 0 0 0 0 0 0 0 0 0
29 209 0 0 0 0 0 0 0 205 100 0 0 0 0 18
291 209 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 210 1320 0 50 0 403 50 35 0 0 0 0 0 0
29 210 0 0 0 0 0 0 0 207 100 0 0 0 0 18
291 210 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 211 1320 0 50 0 105 50 35 0 0 0 0 0 0
29 211 0 0 0 0 0 0 0 203 100 0 0 0 0 18
291 211 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 212 1320 0 50 0 0 0 0 0 0 0 0 0 0
29 212 0 0 0 0 0 0 0 201 100 0 0 0 0 18
291 212 0 0 0 0 0 0 0 0 0 0 0 3009003000000
*** Intersection 3 ***
--- Signal Timing ---
14 3 0 5 10 4 1 25 4 1 10 4 1 35 4 0
18 3 1 0 0 0 0 0 0 0 0 0 0 0 0 0
21 3 1 1 2 3 10 0 0 0 0 0 0 0 0 1
21 3 0 0 0 0 0 306 0 0 308 0 0 0 0 2
22 3 4 4 5 6 10 0 0 0 0 0 0 0 0 0
22 3 0 0 0 0 0 0 305 311 0 307 312 0 0 2
23 3 7 7 8 9 10 302 0 0 304 0 0 0 0 1
23 3 0 0 0 0 0 0 0 0 0 0 0 0 0 2
24 3 10 10 11 12 10 0 301 309 0 303 310 0 0 1
24 3 0 0 0 0 0 0 0 0 0 0 0 0 0 2
--- Link Data ---
28 301 1320 1900 400 0 101 300 35 106 50 35 112 50 35 0
29 301 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 301 302 99 0 0 0 0 0 0 0 0 0 3009003010000
28 302 1320 1805 50 0 101 50 35 0 0 0 0 0 0 12
29 302 0 0 0 10 0 0 0 303 100 0 0 0 0 0
291 302 0 0 301 0 1 1 1 1 1 0 0 3009003010000
28 303 1320 1900 400 0 0 0 0 0 0 0 0 0 0
29 303 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 303 304 99 0 0 0 0 0 0 0 0 0 3009003010000
28 304 1320 1805 50 0 0 0 0 0 0 0 0 0 0 12
29 304 0 0 0 10 0 0 0 301 100 0 0 0 0 0
291 304 0 0 303 0 1 1 1 1 1 0 0 3009003010000
28 305 1320 1900 400 0 0 0 0 0 0 0 0 0 0
29 305 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 305 306 99 0 0 0 0 0 0 0 0 0 3009003010000
28 306 1320 1805 50 0 0 0 0 0 0 0 0 0 0 12
29 306 0 0 0 10 0 0 0 307 100 0 0 0 0 0
291 306 0 0 305 0 1 1 1 1 1 0 0 3009003010000
28 307 1320 1900 400 0 407 300 35 402 50 35 410 50 35 0
29 307 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 307 308 99 0 0 0 0 0 0 0 0 0 3009003010000
28 308 1320 1805 50 0 407 50 35 0 0 0 0 0 0 12
29 308 0 0 0 10 0 0 0 305 100 0 0 0 0 0
291 308 0 0 307 0 1 1 1 1 1 0 0 3009003010000
28 309 1320 0 50 0 101 50 35 0 0 0 0 0 0
29 309 0 0 0 0 0 0 0 305 100 0 0 0 0 18
291 309 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 310 1320 0 50 0 0 0 0 0 0 0 0 0 0

```

```

29 310 0 0 0 0 0 0 0 0 307 100 0 0 0 0 18
291 310 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 311 1320 0 50 0 0 0 0 0 0 0 0 0 0 0
29 311 0 0 0 0 0 0 0 0 303 100 0 0 0 0 18
291 311 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 312 1320 0 50 0 407 50 35 0 0 0 0 0 0 0
29 312 0 0 0 0 0 0 0 0 301 100 0 0 0 0 18
291 312 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
*** Intersection 4 ***
--- Signal Timing ---
14 4 0 5 10 4 1 25 4 1 10 4 1 35 4 0
18 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0
21 4 1 1 2 3 10 0 0 0 0 0 0 0 0 1
21 4 0 0 0 0 0 0 406 0 0 408 0 0 0 0 2
22 4 4 4 5 6 10 0 0 0 0 0 0 0 0 0
22 4 0 0 0 0 0 0 0 405 411 0 407 412 0 0 2
23 4 7 7 8 9 10 402 0 0 404 0 0 0 0 0 1
23 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2
24 4 10 10 11 12 10 0 401 409 0 403 410 0 0 1
24 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2
--- Link Data ---
28 401 1320 1900 400 0 201 300 35 206 50 35 212 50 35 0
29 401 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 401 402 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 402 1320 1805 50 0 201 50 35 0 0 0 0 0 0 12
29 402 0 0 0 10 0 0 0 0 403 100 0 0 0 0 0
291 402 0 0 401 0 1 1 1 1 1 1 0 0 3009003010000
28 403 1320 1900 400 0 0 0 0 0 0 0 0 0 0 0
29 403 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 403 404 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 404 1320 1805 50 0 0 0 0 0 0 0 0 0 0 12
29 404 0 0 0 10 0 0 0 401 100 0 0 0 0 0
291 404 0 0 403 0 1 1 1 1 1 1 0 0 3009003010000
28 405 1320 1900 400 0 305 300 35 304 50 35 309 50 35 0
29 405 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 405 406 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 406 1320 1805 50 0 305 50 35 0 0 0 0 0 0 12
29 406 0 0 0 10 0 0 0 407 100 0 0 0 0 0
291 406 0 0 405 0 1 1 1 1 1 1 0 0 3009003010000
28 407 1320 1900 400 0 0 0 0 0 0 0 0 0 0 0
29 407 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
291 407 408 99 0 0 0 0 0 0 0 0 0 0 3009003010000
28 408 1320 1805 50 0 0 0 0 0 0 0 0 0 0 12
29 408 0 0 0 10 0 0 0 405 100 0 0 0 0 0
291 408 0 0 407 0 1 1 1 1 1 1 0 0 3009003010000
28 409 1320 0 50 0 201 50 35 0 0 0 0 0 0 0
29 409 0 0 0 0 0 0 0 0 405 100 0 0 0 0 18
291 409 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 410 1320 0 50 0 0 0 0 0 0 0 0 0 0 0
29 410 0 0 0 0 0 0 0 0 407 100 0 0 0 0 18
291 410 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 411 1320 0 50 0 305 50 35 0 0 0 0 0 0 0
29 411 0 0 0 0 0 0 0 0 403 100 0 0 0 0 18
291 411 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
28 412 1320 0 50 0 0 0 0 0 0 0 0 0 0 0
29 412 0 0 0 0 0 0 0 0 401 100 0 0 0 0 18
291 412 0 0 0 0 0 0 0 0 0 0 0 0 3009003000000
--- Arterial Route Specifications ---
42 1 105 107 205 207 0 0 0 0 0 0 0 0 0
42 2 305 307 405 407 0 0 0 0 0 0 0 0 0
--- Run Specifications ---
-51 0 0 0 100 0 000000 10 30 10 10 20 7781 100 1

```

```

--- Route Parameters ---
60 1 0 3 67 1 1 0 0 1 0 0 0 0 0 3
Route #1 Title
61 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
60 1 0 3 67 1 1 0 0 1 0 0 0 0 0 3
Route #2 Title
61 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
--- Node Coordinates ---
70 1 1320 1320 2 2640 1320 3 1320 2640 4 2640 2640 0 0 0
--- Termination ---
90 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Intersection 1: University and 13th
Intersection 2: University and 34th
Intersection 3: Archer and 13th
Intersection 4: Archer and 34th
Linkname 1, 2: University
Linkname 3, 4: Archer
Linkname 1, 3: 13th
Linkname 2, 4: 34th

```

## Run Title Record

[illegible]

## Record Type 0

Name: Record Type 0: Special Options		Option: Optional, must follow Run Title		
Purpose: Establishes certain advanced user options				
Field	Description	Units	Range	Default
1	'0'	#	0	REQ
2	output format flag (0 = wide 132-column output, 1 = narrow 80-column output)	#	0-1	0
3	TRANSYT "debug" flag (0 = no debug output, 1 = overall debug arrays, 2 = intermediate arrays)	#	0-2	0
4	TRANSYT "debug" flag (0 = no debug output, 1 = interval link list, 2 = stopline flow profile values)	#	0-2	0
5	logical unit for <u>scratch</u> file for updated data set ("punch") routine	#	1-99	4
6	logical unit to "punch" the new data set	#	1-99	7
7	logical unit for the graphics data file (GDF) outputs	#	1-99	8
8	flags for requesting older models for "equivalent lanes" and random delay (d2)	#	0-31	0
9	number of printed lines per page in the legacy output file	lines/ page	20-1000	56
10	flag for printing internal TRANSYT model coefficients	#	0-1	0
11	time-space diagram symbols flag	#	0-1	0
12	maximum number of fatal error messages to process before aborting execution	#	1-1000	100
13	not used			
14	not used			
15	not used			
16	not used			

## Record Type 1

Name: Record Type 1: Run Controls		Option: Required		
Purpose: Establishes system-wide run parameters				
Field	Description	Units	Range	Default
1	'1' for wide (132-column) output, '-1' for narrow (80-column) output	#	-1 or 1	REQ
2	minimum network cycle length (if cycle optimization) network background cycle length (if simulation only)	sec	30-300	REQ
3	maximum network cycle length (if cycle optimization)	sec	30-300	0
4	cycle length search increment (if cycle optimization)	sec	1-270	5
5	left-hand drive flag (1 = left-hand drive)	#	0 or 1	0
6	simulation step size resolution (seconds per step)	sec	1-5	1
7	network-wide default start-up lost time	sec sec*10	0-4 10-40	0
8	network-wide default extension of effective green time	sec sec*10	0-4 10-40	0
9	network-wide stop penalty (for optimization) to use a stop penalty of zero, code '-2'	#	-2-10000	-1
10	output level flag	#	0-3 10-13	0
11	initial timing flag	#	0-2	0
12	duration of analysis (negative value: volumes coded in veh per time period)	minutes	1-1200	15
13	initial timing units flag (0 = seconds, 1 = percent of cycle)	#	0-1	0
14	speed or travel time units flag (0 = cruise speed, 1 = travel time)	#	0-1	0
15	English or metric units flag (0 = English, 1 = metric)	#	0-1	0
16	analysis type (0,1,2 = traditional, 3 = macroscopic w/ linkage to CORSIM, 4 = direct CORSIM)	#	0-4	0

## Record Type 2

Name: Record Type 2: Optimization Node List		Option: Required for optimization runs		
Purpose: Specifies which nodes will be optimized				
Field	Description	Units	Range	Default
1	'2'	#	2	REQ
2	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
3	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
4	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
5	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
6	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
7	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
8	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
9	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
10	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
11	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
12	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
13	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
14	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
15	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0
16	node number to be optimized, or node number for which initial timing will be provided by TRANSYT	#	1-99	0



## Record Type 3

Name: Record Type 3: Movement Numbering Scheme		Option: Optional		
Purpose: Establishes a customized link numbering scheme				
Field	Description	Units	Range	Default
1	'3'	#	3	REQ
2	last two digits of the northbound left-turn movement	#	1-12	REQ
3	last two digits of the northbound through movement	#	1-12	REQ
4	last two digits of the northbound right-turn movement	#	1-12	REQ
5	last two digits of the southbound left-turn movement	#	1-12	REQ
6	last two digits of the southbound through movement	#	1-12	REQ
7	last two digits of the southbound right-turn movement	#	1-12	REQ
8	last two digits of the eastbound left-turn movement	#	1-12	REQ
9	last two digits of the eastbound through movement	#	1-12	REQ
10	last two digits of the eastbound right-turn movement	#	1-12	REQ
11	last two digits of the westbound left-turn movement	#	1-12	REQ
12	last two digits of the westbound through movement	#	1-12	REQ
13	last two digits of the westbound right-turn movement	#	1-12	REQ
14	not used			
15	not used			
16	not used			

## Record Type 4

Name: Record Type 4: Hill-Climb Optimization Step Sizes		Option: Optional		
Purpose: Establishes user-specified optimization step size list				
Field	Description	Units	Range	Default
1	'4'	#	4	REQ
2	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
3	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
4	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
5	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
6	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
7	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
8	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
9	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
10	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
11	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
12	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
13	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
14	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
15	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	
16	step size for hill-climb optimization iteration (positive = offset setpsize, negative = split stepsize)	%	(-)3-(+)50	

## Record Type 5

Name: Record Type 5: Associated Links		Option: Optional		
Purpose: Sets complex associations for phasing optimization				
Field	Description	Units	Range	Default
1	'5'	#	5	REQ
2	primary link number	#	101-9999	REQ
3	secondary link number	#	0, or 101-9999	0
4	secondary link number	#	0, or 101-9999	0
5	secondary link number	#	0, or 101-9999	0
6	primary link number	#	0, or 101-9999	0
7	secondary link number	#	0, or 101-9999	0
8	secondary link number	#	0, or 101-9999	0
9	secondary link number	#	0, or 101-9999	0
10	primary link number	#	0, or 101-9999	0
11	secondary link number	#	0, or 101-9999	0
12	secondary link number	#	0, or 101-9999	0
13	secondary link number	#	0, or 101-9999	0
14	not used			
15	not used			
16	not used			

## Record Type 6

Name: Record Type 6: Hill-Climb Sensitivity Parameters		Option: Optional		
Purpose: Establishes user-specified sensitivity thresholds				
Field	Description	Units	Range	Default
1	'6'	#	6	REQ
2	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
3	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
4	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
5	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
6	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
7	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
8	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
9	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
10	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
11	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
12	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
13	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
14	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
15	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1
16	sensitivity parameter used to stop downstream flow profile calculation	%*100	1-200	1

## Record Type 7

Name: Record Type 7: Shared Lanes		Option: Optional		
Purpose: Defines shared lane links				
Field	Description	Units	Range	Default
1	'7'	#	7	REQ
2	primary link number	#	101-9999	REQ
3	secondary link number	#	0, or 101-9999	0
4	secondary link number	#	0, or 101-9999	0
5	secondary link number	#	0, or 101-9999	0
6	secondary link number	#	0, or 101-9999	0
7	primary link number	#	0, or 101-9999	0
8	secondary link number	#	0, or 101-9999	0
9	secondary link number	#	0, or 101-9999	0
10	secondary link number	#	0, or 101-9999	0
11	secondary link number	#	0, or 101-9999	0
12	primary link number	#	0, or 101-9999	0
13	secondary link number	#	0, or 101-9999	0
14	secondary link number	#	0, or 101-9999	0
15	secondary link number	#	0, or 101-9999	0
16	secondary link number	#	0, or 101-9999	0

## Record Type 8

Name: Record Type 8: Model Coefficients		Option: Optional		
Purpose: Modification or calibration of internal models				
Field	Description	Units	Range	Default
1	'8'	#	8	REQ
2	model number to be modified or calibrated	#	1-150	REQ
3	model coefficient	#		0
4	model coefficient	#		0
5	model coefficient	#		0
6	model coefficient	#		0
7	model coefficient	#		0
8	model coefficient	#		0
9	model coefficient	#		0
10	model coefficient	#		0
11	model coefficient	#		0
12	model coefficient	#		0
13	model coefficient	#		0
14	model coefficient	#		0
15	model coefficient	#		0
16	model coefficient	#		0

## Record Type 10

Name: Record Type 10: Network Parameters		Option: Required		
Purpose: Establishes network-wide default parameter values				
Field	Description	Units	Range	Default
1	'10'	#	10	REQ
2	node number of the master controller (offsets are not optimized for this node)	#	1-99	0
3	a network-wide yellow change interval length (applies if initial timing flag on and node values empty)	sec	0-9	4
4	a network-wide all-red clearance interval length (applies if initial timing flag on and node values empty)	sec	0-9	1
5	a network-wide saturation flow rate	vplphg	1000-2000	1700
6	a network-wide approach speed for external links	mph kmph	5-60 8-97	30 48
7	a network-wide platoon dispersion factor (PDF)	#*100	1-100	35
8	fuel consumption multiplier	#*100	50-200	100
9	an average vehicle spacing used to calculate queuing capacity (jam spatial headway)	feet (m*10)-50	15-50 0-99	25 26
10	orientation flag	#	0-14	0
11	target degree of saturation for actuated controllers (if coded negative, network is modeled as pre-timed)	%	50-100	85
12	threshold degree of saturation to recommend double cycling	%	1-100	25
13	queue spillback penalty	#*100	1-9999	40
14	inflation rate to update cost model estimates	#*100	1-9999	0
15	cost of fuel per gallon (must be coded in U.S. dollars per gallon)	\$*100 /gal	1-9999	125
16	average vehicle occupancy	#*100	1-99999	120

## Record Type 1X

Name: Record Type 1X: Controller Timing		Option: Required, one per node		
Purpose: Establishes controller timing				
Field	Description	Units	Range	Default
1	'1X', where 'X' is the number of signal phases negative sign means do not allow split optimization	#	11-17 (-)12-(-)17	REQ
2	node number (if negative, MOEs will not be included in the network summary outputs)	#	1-99 (-)1-(-)99	REQ
3	offset or yield point value for this signal negative value means uncoordinated signal	sec %	0-(-)300 0-100	0 0
4	reference interval number (to which the offset or yield point is referenced)	#	1-25	1
5	duration of interval 1	sec %	0-300 0-100	0 0
6	duration of interval 2	sec %	0-300 0-100	0 0
7	duration of interval 3	sec %	0-300 0-100	0 0
8	duration of interval 4	sec %	0-300 0-100	0 0
9	duration of interval 5	sec %	0-300 0-100	0 0
10	duration of interval 6	sec %	0-300 0-100	0 0
11	duration of interval 7	sec %	0-300 0-100	0 0
12	duration of interval 8	sec %	0-300 0-100	0 0
13	duration of interval 9	sec %	0-300 0-100	0 0
14	duration of interval 10	sec %	0-300 0-100	0 0
15	duration of interval 11	sec %	0-300 0-100	0 0
16	offset optimization flag (col 76), phasing optimization flags (col 77-80), double cycle flag (col 80)	#	00000- 11199	0 0



## Record Type 18

Name: Record Type 18: Controller Timing Continuation		Option:		
Purpose: Continues controller timing data		Required for 12+ intervals		
Field	Description	Units	Range	Default
1	'18'	#	18	REQ
2	node number (if negative, MOEs will not be included in the network summary outputs)	#	1-99 (-)1-(-)99	REQ
3	duration of interval 12	sec %	0-300 0-100	0 0
4	duration of interval 13	sec %	0-300 0-100	0 0
5	duration of interval 14	sec %	0-300 0-100	0 0
6	duration of interval 15	sec %	0-300 0-100	0 0
7	duration of interval 16	sec %	0-300 0-100	0 0
8	duration of interval 17	sec %	0-300 0-100	0 0
9	duration of interval 18	sec %	0-300 0-100	0 0
10	duration of interval 19	sec %	0-300 0-100	0 0
11	duration of interval 20	sec %	0-300 0-100	0 0
12	duration of interval 21	sec %	0-300 0-100	0 0
13	duration of interval 22	sec %	0-300 0-100	0 0
14	duration of interval 23	sec %	0-300 0-100	0 0
15	duration of interval 24	sec %	0-300 0-100	0 0
16	duration of interval 25	sec %	0-300 0-100	0 0

## Record Type 2Y

Name: Record Type 2Y: Phasing Data		Option:		
Purpose: Establishes the interval and phase sequence		Required, one per phase		
Field	Description	Units	Range	Default
1	'2Y', where 'Y' is the phase number RT 2Y's must be in ascending order	#	21-27	REQ
2	node number	#	1-99	REQ
3	interval number that starts the green for this phase	#	1-25	REQ
4	variable interval number for this phase	#	1-25	field 3
5	yellow interval number for this phase	#	1-25	REQ
6	all-red interval number for this phase	#	1-25	0
7	minimum phase duration (includes yellow and all-red)	sec	2-300	REQ
8	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
9	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
10	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
11	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
12	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
13	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
14	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
15	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
16	actuated phase flag (0 = non-actuated, 1 = actuated) 50-100 = override the global target degree of saturation	# %	0-1 50-100	0

## Record Type 2Z

Name: Record Type 2Z: Phasing Data Continuation		Option: Required for 9+ moving links		
Purpose: Establishes phase sequence				
Field	Description	Units	Range	Default
1	'2Z', where 'Z' is the same phase number as the preceding RT 2Y	#	21-27	REQ
2	node number	#	1-99	REQ
3	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
4	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
5	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
6	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
7	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
8	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
9	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
10	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
11	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
12	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
13	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
14	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
15	number of a link that receives green in this phase negative link numbers are permitted (opposed) links	#	0, or 101-9999	0
16	continuation flag for RT 2Z	#	2	REQ

## Record Type 28

Name: Record Type 28: Link Data		Option: Required, one per link		
Purpose: Establishes the basic link data				
Field	Description	Units	Range	Default
1	'28' (or '-28' for bottleneck or dummy links)	#	-28 or 28	REQ
2	link number of the current link	#	101-9999	REQ
3	link length	feet meters	50-16404 15-5000	0
4	saturation flow rate	vphg	50-20000	REQ
5	total flow or total volume	vph vpp	10-15000	REQ
6	mid-block source flow	vph vpp	0-15000	0
7	link number of the first upstream feeding link	#	0, or 101-9999	0
8	feeding flow from the first upstream feeding link	vph vpp	10-15000	0
9	average cruise speed for the first upstream feeding link or average cruise speed for this external link	mph kmph	5-60 8-97	0
10	link number of the second upstream feeding link	#	0, or 101-9999	0
11	feeding flow from the second upstream feeding link	vph vpp	10-15000	0
12	average cruise speed for the second upstream feeding link	mph kmph	5-60 8-97	0
13	link number of the third upstream feeding link	#	0, or 101-9999	0
14	feeding flow from the third upstream feeding link	vph vpp	10-15000	0
15	average cruise speed for the third upstream feeding link	mph kmph	5-60 8-97	0
16	desired queuing capacity (i.e., maximum back of queue capacity)	veh	0-200	0

## Record Type 29

Name: Record Type 29: Link Data Continuation		Option: Optional		
Purpose: Provides additional link data				
Field	Description	Units	Range	Default
1	'29' (or '-29' for bottleneck or dummy links)	#	-29 or 29	REQ
2	link number of the current link (must match the preceding record type 28)	#	101-9999	REQ
3	addition to or reduction of the start-up lost time on this link (also see RT 1, field 7)	sec*10	-40-cycle	0
4	addition to or reduction of the extension of effective green time on this link (also see RT 1, field 8)	sec*10	-40-cycle	0
5	permitted movement maximum flow rate	vphg	50-2000	0
6	sneakers per cycle	veh*10	0-50	0
7	link number of the fourth upstream feeding link	#	0, or 101-9999	0
8	feeding flow from the fourth upstream feeding link	vph vpp	10-15000	0
9	average cruise speed for the fourth upstream feeding link	mph kmph	5-60 8-97	0
10	link number of the first opposing link	#	0, or 101-9999	0
11	percentage of opposing flow for the first opposing link	%	1-100	100
12	link number of the second opposing link	#	0, or 101-9999	0
13	percentage of opposing flow for the second opposing link	%	1-100	100
14	link number of the third opposing link	#	0, or 101-9999	0
15	percentage of opposing flow for the third opposing link	%	1-100	100
16	permitted movement model override	#	-3-50	0

## Record Type 291

Name: Record Type 291: Link Data Further Continuation		Option: Optional		
Purpose: Provides additional link data				
Field	Description	Units	Range	Default
1	'291'	#	291	REQ
2	link number of the current link (must match the preceding record type 29)	#	101-9999	REQ
3	link number (generally a left-turn link with turn bay) affected by congestion on the link coded in field 2	#	0, or 101-9999	0
4	percent reduction in saturation flow rate due to queue spillover from the link coded in field 2	%	0-100	0
5	link number (generally a right-turn link with turn bay) affected by congestion on the link coded in field 2	#	0, or 101-9999	0
6	percent reduction in saturation flow rate due to queue spillover from the link coded in field 2	%	0-100	0
7	percent reduction in saturation flow rate of the feeding link coded in RT 28 field 7 due to queue spillback	%	0-100	100
8	percent reduction in saturation flow rate of the feeding link coded in RT 28 field 10 due to queue spillback	%	0-100	100
9	percent reduction in saturation flow rate of the feeding link coded in RT 28 field 13 due to queue spillback	%	0-100	100
10	percent reduction in saturation flow rate of the feeding link coded in RT 29 field 7 due to queue spillback	%	0-100	100
11	set to 1 if this is a turn bay	#	0-1	0
12	jam spatial headway for this link	feet m*10	15-50 45-150	field 9 RT 10
13	optimal spatial headway for this link	feet m*10	45-150 135-450	field 9 RT 10
14	gap setting or unit extension for actuated controllers	sec*100	0-500	300
15	detector length for actuated controllers (columns 71-72 = peak hour factor * 100)	feet meters	20-100 6-31	30 9
16	miscellaneous flags (columns 76-78 = lane use flags, columns 79-80 = actuated estimation flags)	#		0

The lane configuration inputs below, which are only used by the Lane Configuration (Edit > Lanes) screen, are not mandatory. In the absence of these inputs, or when using an older input file, lane configuration is usually estimated correctly by the Lanes screen for most conditions. The TRANSYT model is ultimately only responsive to the adjusted volumes, adjusted saturation flow rates, and link definitions within the Traffic screen.

Field 16, column 76, indicates the number of lanes for link numbers 1 through 12 within the link numbering scheme. If blank, the number of lanes is estimated via the saturation flow rate.

Field 16, column 76, indicates the type of turning movement for link numbers 13 and higher (not within the link numbering scheme), where 1=left, 2=thru, and 3=right.

Field 16, column 77, indicates the lane number for link numbers 13 and higher (not within the link numbering scheme), where lane #1 is the leftmost lane.

Field 16, column 78, indicates the approach direction for link numbers 13 and higher (not within the link numbering scheme), where 1=NB, 2=SB, 3=EB, and 4=WB.

## Record Type 292

Name: Record Type 292: Satflow Adjustment Factors		Option: Optional		
Purpose: Provides assistance in computing satflow rates				
Field	Description	Units	Range	Default
1	'292'	#	292	REQ
2	link number of the current link (must match the preceding record type 29 or 291)	#	101-9999	REQ
3	base saturation flow rate	vphg	1300-2300	1900
4	number of lanes	#	1-9	0
5	lane width	feet meters	8-16 2.4-4.8	12 3.6
6	heavy vehicles	%	0-25	0
7	grade	%	-6-10	0
8	parking prohibited	yes or no	yes or no	yes
9	parking maneuvers per hour	#	0-180	0
10	number of buses stopping	#	0-250	0
11	area type	other CBD	other CBD	other
12	lane utilization (equal, slightly unequal, moderately unequal, highly unequal)			slightly unequal
13	type of turning movement (left, thru, right)			
14	ped-bike influence on left-turns (none, light, medium, heavy)			none
15	ped-bike influence on right-turns (none, light, medium, heavy)			none
16	calculated saturation flow rate	vphg	0-99999	1900

## Record Type 30

Name: Record Type 30: Performance Index (PI) Weighting		Option: Optional		
Purpose: Establishes link-specific performance index weighting				
Field	Description	Units	Range	Default
1	'30'	#	30	REQ
2	the weight that, when divided by 100, multiplies the PI calculation on this link	%	0-9999	0
3	a link for which the value in field 2 applies if no link list coded, the modification applies to all links	#	0, or 101-9999	0
4	a link for which the value in field 2 applies	#	0, or 101-9999	0
5	a link for which the value in field 2 applies	#	0, or 101-9999	0
6	a link for which the value in field 2 applies	#	0, or 101-9999	0
7	a link for which the value in field 2 applies	#	0, or 101-9999	0
8	a link for which the value in field 2 applies	#	0, or 101-9999	0
9	a link for which the value in field 2 applies	#	0, or 101-9999	0
10	a link for which the value in field 2 applies	#	0, or 101-9999	0
11	a link for which the value in field 2 applies	#	0, or 101-9999	0
12	a link for which the value in field 2 applies	#	0, or 101-9999	0
13	a link for which the value in field 2 applies	#	0, or 101-9999	0
14	a link for which the value in field 2 applies	#	0, or 101-9999	0
15	a link for which the value in field 2 applies	#	0, or 101-9999	0
16	a link for which the value in field 2 applies	#	0, or 101-9999	0



## Record Type 31

Name: Record Type 31: Link-to-Link Flow Weighting		Option: Optional		
Purpose: Establishes link-to-link progression weighting				
Field	Description	Units	Range	Default
1	'31'	#	31	REQ
2	the input (feeder) number of the upstream link feeding the links listed on the remainder of this record type	#	1-4	REQ
3	a link for which the feeder number in field 2 applies if no link list coded, the modification applies to all links	#	0, or 101-9999	0
4	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
5	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
6	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
7	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
8	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
9	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
10	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
11	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
12	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
13	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
14	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
15	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0
16	a link for which the feeder number in field 2 applies	#	0, or 101-9999	0

## Record Type 32

Name: Record Type 32: Queuing Ratio Weighting		Option: Optional		
Purpose: Establishes link-specific queuing ratio weighting				
Field	Description	Units	Range	Default
1	'32'	#	32	REQ
2	the weight that, when divided by 100, is the power of the queuing ratio applied to the links in fields 3-16	#*100	1-9999	100
3	an internal link for which the value in field 2 applies if no link list coded, value applies to all internal links	#	0, or 101-9999	0
4	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
5	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
6	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
7	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
8	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
9	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
10	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
11	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
12	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
13	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
14	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
15	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
16	an internal link for which the value in field 2 applies	#	0, or 101-9999	0

## Record Type 33

Name: Record Type 33: Throughput Weighting		Option: Optional		
Purpose: Establishes link-specific throughput weighting				
Field	Description	Units	Range	Default
1	'33'	#	33	REQ
2	the weight that, when divided by 100, is the power of the throughput applied to links in fields 3-16	#*100	1-9999	100
3	an internal link for which the value in field 2 applies if no link list coded, value applies to all internal links	#	0, or 101-9999	0
4	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
5	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
6	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
7	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
8	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
9	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
10	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
11	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
12	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
13	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
14	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
15	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
16	an internal link for which the value in field 2 applies	#	0, or 101-9999	0

## Record Type 34

Name: Record Type 34: Queuing Capacity Modifier		Option: Optional		
Purpose: Modifies link-specific queuing capacity				
Field	Description	Units	Range	Default
1	'34'	#	34	REQ
2	percentage of the queuing capacity to be used for the desired storage zone	%	20-100	100
3	an internal link for which the value in field 2 applies if no link list coded, value applies to all internal links	#	0, or 101-9999	0
4	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
5	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
6	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
7	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
8	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
9	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
10	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
11	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
12	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
13	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
14	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
15	an internal link for which the value in field 2 applies	#	0, or 101-9999	0
16	an internal link for which the value in field 2 applies	#	0, or 101-9999	0

## Record Type 35

Name: Record Type 35: Cruise Speed Multiplier		Option: Optional		
Purpose: Modifies link-specific cruise speeds				
Field	Description	Units	Range	Default
1	'35'	#	35	REQ
2	the factor which, when divided by 100, multiplies the cruise speeds for the links in fields 3-16 of this record	%	50-200	100
3	a link number for which the multiplier in field 2 applies if no link list coded, the multiplier applies to all links	#	0, or 101-9999	0
4	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
5	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
6	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
7	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
8	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
9	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
10	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
11	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
12	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
13	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
14	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
15	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
16	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0

## Record Type 36

Name: Record Type 36: Volume Multiplier		Option: Optional		
Purpose: Modifies link-specific traffic volumes				
Field	Description	Units	Range	Default
1	'36'	#	36	REQ
2	the factor which, when divided by 100, multiplies the total flow rates for the links in fields 3-16 of this record	%	5-2000	100
3	a link number for which the multiplier in field 2 applies if no link list coded, the multiplier applies to all links	#	0, or 101-9999	0
4	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
5	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
6	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
7	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
8	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
9	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
10	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
11	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
12	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
13	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
14	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
15	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
16	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0

## Record Type 37

Name: Record Type 37: Delay Weight Multiplier		Option: Optional		
Purpose: Establishes link-specific delay weights in the DI				
Field	Description	Units	Range	Default
1	'37'	#	37	REQ
2	the factor which, when divided by 100, multiplies the delay calculation in the DI for the links in fields 3-16	%	0-9999	0
3	a link number for which the multiplier in field 2 applies if no link list coded, the multiplier applies to all links	#	0, or 101-9999	0
4	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
5	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
6	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
7	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
8	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
9	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
10	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
11	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
12	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
13	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
14	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
15	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
16	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0

## Record Type 38

Name: Record Type 38: Stop Weight Multiplier		Option: Optional		
Purpose: Establishes link-specific stop weights in the DI				
Field	Description	Units	Range	Default
1	'38'	#	38	REQ
2	the factor which, when divided by 100, multiplies the stops calculation in the DI for the links in fields 3-16	%	0-9999	0
3	a link number for which the multiplier in field 2 applies a global multiplier can be coded on RT 1, field 9	#	0, or 101-9999	0
4	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
5	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
6	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
7	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
8	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
9	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
10	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
11	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
12	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
13	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
14	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
15	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0
16	a link number for which the multiplier in field 2 applies	#	0, or 101-9999	0



## Record Type 39

Name: Record Type 39: Platoon Dispersion Factors		Option: Optional		
Purpose: Establishes link-specific platoon dispersion factors				
Field	Description	Units	Range	Default
1	'39'	#	39	REQ
2	the number which, when divided by 100, represents the platoon dispersion factor (PDF) for links in fields 3-16	#*100	0-100	0
3	a link number for which the PDF in field 2 applies if no link list coded, the PDF applies to all links	#	0, or 101-9999	0
4	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
5	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
6	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
7	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
8	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
9	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
10	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
11	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
12	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
13	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
14	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
15	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0
16	a link number for which the PDF in field 2 applies	#	0, or 101-9999	0

## Record Type 40

Name: Record Type 40: Flow Profile Diagrams		Option: Optional		
Purpose: Requests flow profile diagrams in the output file				
Field	Description	Units	Range	Default
1	'40'	#	40	REQ
2	link number for plot on left-hand side of page	#	0, or 101-9999	0
3	link number for plot on right-hand side of page (or left-hand side, if narrow output format selected)	#	0, or 101-9999	0
4	link number for plot on left-hand side of page	#	0, or 101-9999	0
5	link number for plot on right-hand side of page (or left-hand side, if narrow output format selected)	#	0, or 101-9999	0
6	link number for plot on left-hand side of page	#	0, or 101-9999	0
7	link number for plot on right-hand side of page (or left-hand side, if narrow output format selected)	#	0, or 101-9999	0
8	link number for plot on left-hand side of page	#	0, or 101-9999	0
9	link number for plot on right-hand side of page (or left-hand side, if narrow output format selected)	#	0, or 101-9999	0
10	link number for plot on left-hand side of page	#	0, or 101-9999	0
11	link number for plot on right-hand side of page (or left-hand side, if narrow output format selected)	#	0, or 101-9999	0
12	link number for plot on left-hand side of page	#	0, or 101-9999	0
13	link number for plot on right-hand side of page (or left-hand side, if narrow output format selected)	#	0, or 101-9999	0
14	link number for plot on left-hand side of page	#	0, or 101-9999	0
15	link number for plot on right-hand side of page (or left-hand side, if narrow output format selected)	#	0, or 101-9999	0
16	not used			

## Record Type 41

Name: Record Type 41: Bandwidth Constraints		Option: Optional		
Purpose: Establishes bandwidth constraints for the thru bands				
Field	Description	Units	Range	Default
1	'41'	#	41	REQ
2	link number applicable to the band defined in fields 3 and 4	#	101-9999	REQ
3	the start time of the current thru band	sec	0-300	REQ
4	the end time of the current thru band	sec	0-300	REQ
5	link number applicable to the band defined in fields 6 and 7	#	101-9999	0
6	the start time of the current thru band	sec	0-300	0
7	the end time of the current thru band	sec	0-300	0
8	link number applicable to the band defined in fields 9 and 10	#	101-9999	0
9	the start time of the current thru band	sec	0-300	0
10	the end time of the current thru band	sec	0-300	0
11	link number applicable to the band defined in fields 12 and 13	#	101-9999	0
12	the start time of the current thru band	sec	0-300	0
13	the end time of the current thru band	sec	0-300	0
14	link number applicable to the band defined in fields 15 and 16	#	101-9999	0
15	the start time of the current thru band	sec	0-300	0
16	the end time of the current thru band	sec	0-300	0

## Record Type 42

Name: Record Type 42: Arterial Routes		Option: Optional		
Purpose: Defines routes for PROS & other route analyses				
Field	Description	Units	Range	Default
1	'42'	#	42	REQ
2	arterial reference number applicable to the route defined by link numbers in fields 3 through 16	#	1-9999	REQ
3	the first (entry) link of the arterial route this link indicates the "forward" direction of the route	#	101-9999	REQ
4	link opposing the entry link in field 3 at the same node this link indicates the "reverse" direction of the route	#	101-9999	REQ
5	another forward direction link within the arterial route	#	101-9999	REQ
6	another reverse direction link within the arterial route	#	101-9999	REQ
7	another forward direction link within the arterial route	#	101-9999	0
8	another reverse direction link within the arterial route	#	101-9999	0
9	another forward direction link within the arterial route	#	101-9999	0
10	another reverse direction link within the arterial route	#	101-9999	0
11	another forward direction link within the arterial route	#	101-9999	0
12	another reverse direction link within the arterial route	#	101-9999	0
13	another forward direction link within the arterial route	#	101-9999	0
14	another reverse direction link within the arterial route	#	101-9999	0
15	another forward direction link within the arterial route	#	101-9999	0
16	another reverse direction link within the arterial route	#	101-9999	0

## Record Type 43

Name: Record Type 43: Arterial Weighting		Option: Optional		
Purpose: Defines weighting for PROS & other route analyses				
Field	Description	Units	Range	Default
1	'43'	#	43	REQ
2	arterial reference number applicable to the factors coded in fields 3 and 4	#	1-9999	REQ
3	the factor that multiplies PROS and CPROS calculated for the "forward" direction of the artery	#	1-9999	REQ
4	the factor that multiplies PROS and CPROS calculated for the "reverse" direction of the artery	#	1-9999	REQ
5	arterial reference number applicable to the factors coded in fields 6 and 7	#	1-9999	0
6	the factor that multiplies PROS and CPROS calculated for the "forward" direction of the artery	#	1-9999	0
7	the factor that multiplies PROS and CPROS calculated for the "reverse" direction of the artery	#	1-9999	0
8	arterial reference number applicable to the factors coded in fields 9 and 10	#	1-9999	0
9	the factor that multiplies PROS and CPROS calculated for the "forward" direction of the artery	#	1-9999	0
10	the factor that multiplies PROS and CPROS calculated for the "reverse" direction of the artery	#	1-9999	0
11	arterial reference number applicable to the factors coded in fields 12 and 13	#	1-9999	0
12	the factor that multiplies PROS and CPROS calculated for the "forward" direction of the artery	#	1-9999	0
13	the factor that multiplies PROS and CPROS calculated for the "reverse" direction of the artery	#	1-9999	0
14	arterial reference number applicable to the factors coded in fields 15 and 16	#	1-9999	0
15	the factor that multiplies PROS and CPROS calculated for the "forward" direction of the artery	#	1-9999	0
16	the factor that multiplies PROS and CPROS calculated for the "reverse" direction of the artery	#	1-9999	0

## Record Type 5X

Name: Record Type 5X: Run Instructions		Option: Required, one per time period		
Purpose: Defines the run-type and objective function				
Field	Description	Units	Range	Default
1	'5X', where this number indicates the type of simulation or optimization run to be performed	#	50-59 (-)50-(-)59	REQ
2	definition of the disutility index (DI)	#	0-3	0
3	definition of the performance index (PI)	#	-1, or 0-7	0
4	a factor that assigns the relative weight of PROS only applicable if '2' has been coded in field 3	%	0-1000	100
5	length of the simulation initialization period	sec	30-3600	cycle length
6	maximum v/c ratio above which a penalty is applied only applicable if '7' has been coded in field 3	#*100	0-300	100
7	penalty applied if the maximum v/c ratio is exceeded only applicable if '7' has been coded in field 3	#	0-9999	0
8	signal settings to be optimized, and optimization algorithm	#	00000-11113	00000
9	population size only applicable when genetic optimization is selected	#	6-99	10
10	crossover probability only applicable when genetic optimization is selected	%	1-100	30 (30%)
11	mutation probability (a '99' coded implies 0.0%) only applicable when genetic optimization is selected	%*10	99, or 1-1000	10 (1%)
12	convergence threshold only applicable when genetic optimization is selected	%*100	1-10000	10 (0.01%)
13	maximum number of generations only applicable when genetic optimization is selected	#	1-9999	20
14	random number seed only applicable when genetic optimization is selected	#	1-9999	7781
15	original cycle length only applicable when genetic optimization is selected	sec	30-300	0
16	graphics data file output request flag for TRANSYT, or random number seed variations for CORSIM	#	0-1	0
		#	1-9999	1

## Record Type 60

Name: Record Type 60: Route Parameters		Option: Optional		
Purpose: Requests route-specific plots and outputs				
Field	Description	Units	Range	Default
1	'60' each new diagram must begin with this record type	#	60	REQ
2	time-space diagram (TSD) flag (0 = do not generate TSD, 1 = generate TSD)	#	0-1	1
3	time-space diagram time units flag (0 = same as RT 1, 1 = seconds, 2 = percent)	#	0-2	1
4	time-space diagram time axis scale factor (number of seconds per character)	sec/ char	1-10	3
5	time-space diagram distance axis scale factor (number of feet or meters per line)	feet meters	25-300 8-90	67 20
6	flow profile diagram (FPD) flag (0 = do not generate FPD, 1 = generate FPD)	#	0-1	1
7	route summary report (RSR) flag (0 = do not generate RSR, 1 = generate RSR)	#	0-1	1
8	platoon progression diagram (PPD) flag (0 = do not generate PPD, 1 = generate PPD)	#	0-1	1
9	single ring timing (SRT) flag (0 = do not generate SRT, 1 = generate SRT)	#	0-1	1
10	time location diagram (TLD) flag (0 = do not generate TLD, 1-3 = generate TLD)	#	0-(-)3	0
11	not used			
12	not used			
13	not used			
14	not used			
15	not used			
16	route orientation (1 = NB, 2 = SB, 3 = EB, 4 = WB) affects time-space diagrams (TSD)	#	0-4	RT 3 RT 10

## Route Title Record

[illegible]



## Record Type 61

Name: Record Type 61: Route Link		Option:		
Purpose: Establishes the sequence of links in a route		Required after any Route Title		
Field	Description	Units	Range	Default
1	'61' this record type must follow the Route Title	#	61	REQ
2	route number from record type 42, or a forward direction link on this route	#	1-9999 101-9999	REQ
3	blank, if a route number was coded in field 2, or a reverse direction link on this route	#	101-9999	
4	blank, if a route number was coded in field 2, or a forward direction link on this route	#	101-9999	
5	blank, if a route number was coded in field 2, or a reverse direction link on this route	#	101-9999	
6	blank, if a route number was coded in field 2, or a forward direction link on this route	#	101-9999	
7	blank, if a route number was coded in field 2, or a reverse direction link on this route	#	101-9999	
8	blank, if a route number was coded in field 2, or a forward direction link on this route	#	101-9999	
9	blank, if a route number was coded in field 2, or a reverse direction link on this route	#	101-9999	
10	blank, if a route number was coded in field 2, or a forward direction link on this route	#	101-9999	
11	blank, if a route number was coded in field 2, or a reverse direction link on this route	#	101-9999	
12	blank, if a route number was coded in field 2, or a forward direction link on this route	#	101-9999	
13	blank, if a route number was coded in field 2, or a reverse direction link on this route	#	101-9999	
14	blank, if a route number was coded in field 2, or a forward direction link on this route	#	101-9999	
15	blank, if a route number was coded in field 2, or a reverse direction link on this route	#	101-9999	
16	not used			

## Record Type 70

Name: Record Type 70: Node Coordinates		Option: Optional		
Purpose: Establishes node locations for the optional Map View				
Field	Description	Units	Range	Default
1	'70'	#	70	REQ
2	node number	#	1-99	REQ
3	X (horizontal) coordinate, applicable to the node number coded in field 2	twips	-9999 99999	0
4	Y (vertical) coordinate, applicable to the node number coded in field 2	twips	-9999 99999	0
5	node number	#	1-99	0
6	X (horizontal) coordinate, applicable to the node number coded in field 5	twips	-9999 99999	0
7	Y (vertical) coordinate, applicable to the node number coded in field 5	twips	-9999 99999	0
8	node number	#	1-99	0
9	X (horizontal) coordinate, applicable to the node number coded in field 8	twips	-9999 99999	0
10	Y (vertical) coordinate, applicable to the node number coded in field 8	twips	-9999 99999	0
11	node number	#	1-99	0
12	X (horizontal) coordinate, applicable to the node number coded in field 11	twips	-9999 99999	0
13	Y (vertical) coordinate, applicable to the node number coded in field 11	twips	-9999 99999	0
14	node number	#	1-99	0
15	X (horizontal) coordinate, applicable to the node number coded in field 14	twips	-9999 99999	0
16	Y (vertical) coordinate, applicable to the node number coded in field 14	twips	-9999 99999	0

## Record Type 71

Name: Record Type 71: Link Curvature		Option: Optional		
Purpose: Establishes optional link curvature for the Map View				
Field	Description	Units	Range	Default
1	'71'	#	71	REQ
2	upstream node number	#	1-99	REQ
3	downstream node number	#	1-99	REQ
4	X (horizontal) coordinate, applicable to the first curvature control point	twips	-9999 99999	REQ
5	Y (vertical) coordinate, applicable to the first curvature control point	twips	-9999 99999	REQ
6	X (horizontal) coordinate, applicable to the second curvature control point	twips	-9999 99999	REQ
7	Y (vertical) coordinate, applicable to the second curvature control point	twips	-9999 99999	REQ
8	not used			
9	not used			
10	not used			
11	not used			
12	not used			
13	not used			
14	not used			
15	not used			
16	not used			

## Record Type 9X

Name: Record Type 9X: Termination Instructions		Option:		
Purpose: Establishes run termination status		Required, one per time period		
Field	Description	Units	Range	Default
1	'9X', where 'X' may be (0 = end of job, 1 = another data set, 2 = another file name, 3 = another time period)	#	90-93	REQ
2	not used			
3	not used			
4	not used			
5	not used			
6	not used			
7	not used			
8	not used			
9	not used			
10	not used			
11	not used			
12	not used			
13	not used			
14	not used			
15	not used			
16	not used			

## Intersection Name Record

[illegible]

## Link Name Record

Name: Link Name Record		Option: Optional		
Purpose: Provides street names for each link				
Field	Description	Units	Range	Default
	The first 9 characters must say "Linkname "			
	Columns 10 and 11 must contain the upstream node number, and this node number must be right-justified			
	Column 12 must contain a comma ","			
	Columns 13 and 14 must contain the downstream node number, right-justified			
	Column 15 must contain a colon ":"			
	Columns 17 and beyond contain the link name			
	Example: Linkname 1, 2: University Avenue			

## Header Information Record

Name: Header Information Record		Option: Optional		
Purpose: Provides agency information for printed reports				
Field	Description	Units	Range	Default
	hdr_Analyst: analyst performing the analysis			
	hdr_Agency: agency performing the analysis			
	hdr_Analysis: analysis time period			
	hdr_Date: date that the analysis was performed			
	hdr_Juris: jurisdiction represented by the analysis			
	Example: hdr_Analyst: DKH			
	hdr_Agency: McTrans Center hdr_Analysis: 5-6 PM			
	hdr_Date: 8/2/2006 hdr_Juris: Alachua			



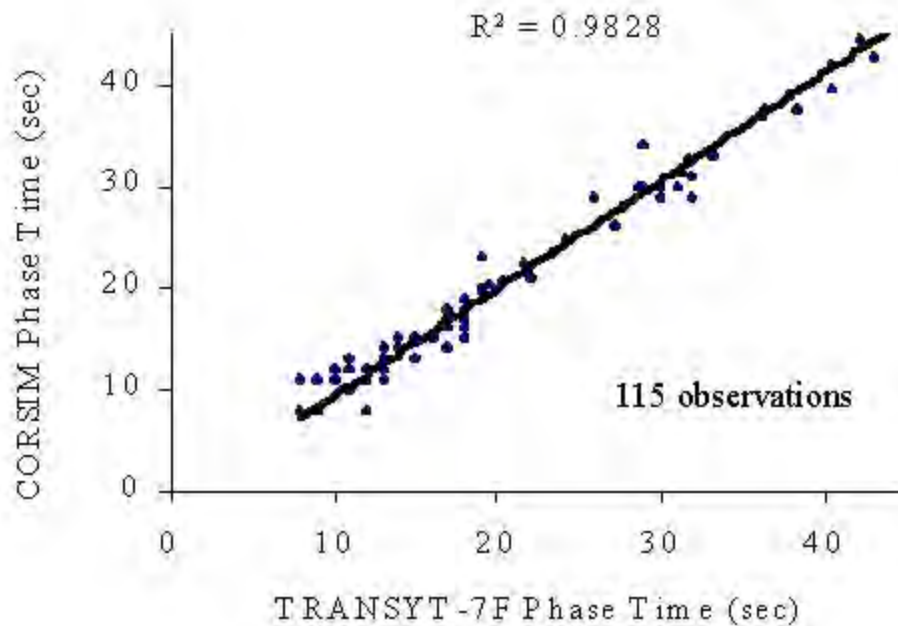


# Chapter 4: TRANSYT-7F Model

## Actuated Estimation Methodology

Refer to the Traffic-Actuated Control section of the documentation for more details about practical model application, and refer to the Actuated Estimation Examples section for tutorial examples.

The actuated estimation model is designed to calculate average phase times as a function of user-coded maximum greens and force-offs. The estimation model is capable of computing traffic-actuated phase times with precision (e.g., R squared = 0.98 with CORSIM, illustrated below). The new methodology uses TRANSYT step-wise simulation to compute queue service times that are sensitive to progression effects, protected-permitted left-turn effects, and queue spillback effects. The new methodology also uses applied probability analysis of the TRANSYT flow profile to determine when gap-out takes place, or whether force-off/max-out occurs. Also included are an advanced green time reallocation algorithm, advanced search for critical links, and sensitivity to stochastic (multilane or competing link) effects. The end result is that performance estimates generated by TRANSYT, after computing actuated phase times under the new methodology, are more realistic. Technical aspects of the new methodology are summarized in Appendix A [37], and detailed in the supplemental research report "Modeling Traffic-Actuated Control with TRANSYT-7F" [36].



Actuated estimation runs are requested by selecting "Estimation" on the Edit > Analysis screen. Following the estimation run, a "punch" input data file will be created containing the calculated phase times. The punch file can optionally be used for subsequent simulation or optimization runs.

As stated in the Chapter 2 section on Traffic-Actuated Control, traffic networks tend to be optimized when actuated phases operate in the 80%-90% degree of saturation range. This range allows random delay to remain reasonably low for the actuated phases, but without borrowing too much green time from the major street (if semi-actuated), or without forcing vehicles to wait through too long of a cycle length (if fully-actuated). Under these ideal conditions, the estimation model tends to demonstrate that the force-off or maximum green can be set equal to the average or optimum phase time, such that the queue could be served anytime during the last few seconds of the phase. When conditions are oversaturated, the estimation model tends to demonstrate that the force-off or maximum green is again equal to the average or optimum phase time, but that the queue is often not fully served. When actuated phases operate in the inefficient 0%-79% range, the estimation model tends to demonstrate that the force-off or maximum green is several seconds higher than the average or optimum phase time. This tends to severely inflate either major street delay (if semi-actuated), or uniform delay (if fully-actuated).

The estimation model employs single-cycle, step-wise simulation. Multi-cycle and multi-period simulation are not supported in conjunction with actuated estimation, but in most cases, this should not compromise the accuracy of results. Multi-cycle and multi-period simulation are primarily needed to measure the exact extent of queue spillback, but this is unnecessary when estimating actuated phase durations, because the duration of an oversaturated actuated phase will naturally be equal to the maximum green time. Only one cycle of simulation is needed to determine this. In the scenario where several cycles would elapse before queue spillback would begin to affect (i.e., increase) actuated phase durations at the upstream signal, engineering judgment would be required, because the estimation model would not simulate this explicitly.

The original target degree of saturation (initial timing) model, employed by TRANSYT for many years, is effective for optimization runs. This model facilitates location of the optimal actuated phase times, i.e., phase times that would result in 80%-90% degrees of saturation. To ensure that the optimal actuated phase times materialize in the field, it is necessary to set the force-offs and maximum greens appropriately. The estimation model can subsequently be used to confirm whether or not the force-offs

and maximum greens have been set appropriately.

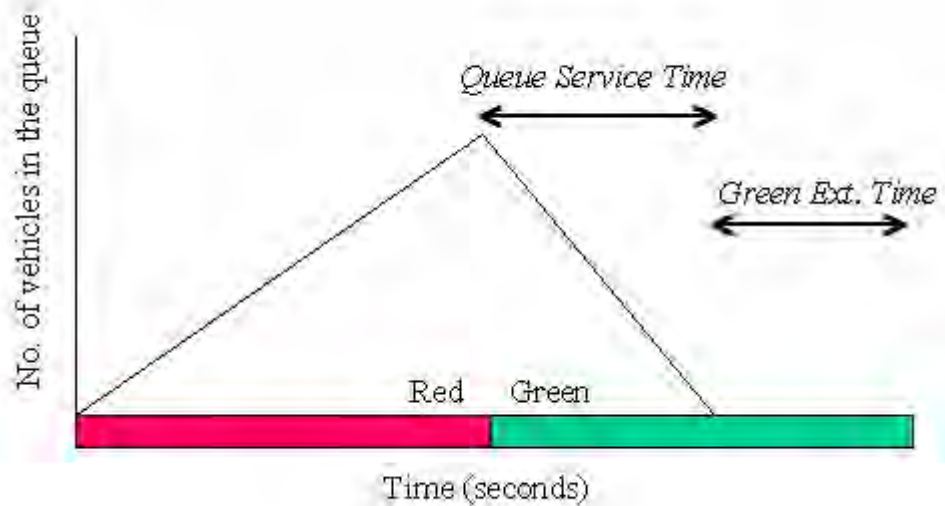
While the estimation model is running, the familiar TRANSYT-7F banner screen will intermittently pop up in an adjacent "console" window that will be minimized and without focus. The user can switch focus to that window, if desired, to monitor the progress of individual simulation runs. If the GUI indicates that TRANSYT-7F "is running", but no adjacent console window is present, this indicates that the estimation model is actively re-calculating actuated phase times as a result of the latest TRANSYT simulation run.

### Input Data

The estimation model cannot process node numbers higher than 99 in the TRANSYT-7F input (\*.TIN) file. In addition, link numbers 1 through 9 must contain a leading zero. For example, link number 1 at node number 54 must be coded as '5401' instead of '541'. A code of '541' would be interpreted as link number 41 at node number 5. The software issues an error message if any node numbers higher than 99 are detected within the TRANSYT-7F input file.

Although several required input parameters may be optionally specified inside the Edit > Optional > Actuated screen, other input parameters must be specified correctly within the standard TRANSYT-7F (\*.TIN) input file. Specifically, initial timing plans from the Edit > Timing screen (or record types 1X and 18) are interpreted by the estimation model as maximum greens or force-offs. The initial set of force-offs or maximum greens can potentially have a significant impact on overall results. If these signal settings are not specified in the input file, the estimation model must perform a preliminary TRANSYT simulation run to obtain an initial timing plan. In addition, actuated phase flags must be specified as usual on the Edit > Timing screen (or record type 2Y field 16). The program issues a warning message in the event that certain intersections within the network do not have actuated flag definitions. The estimation model uses input data from the standard input file in conjunction with TRANSYT simulation results in order to compute the actuated phase times; however, the following additional input data (related to the actuated estimation methodology) can be specified within the Edit > Optional > Actuated screen.

The queue service time model calculates the amount of time required to serve the initial queue, after the signal has turned green. The green extension time model calculates the amount of time until a suitable gap has been located at the detector, following the queue service time. The default models shown in the combo boxes are the ones that displayed the best performance during extensive testing, as described in the supplemental documentation [36]. The queue service time and green extension time are illustrated on the queue accumulation polygon diagram below.



The "T7F" selection (from the Edit > Optional > Actuated screen) indicates that queue service times will be obtained directly from TRANSYT step-wise simulation. The other models available for selection are existing models from the literature. These other models are only suitable for isolated intersections without overlapping phasing or movements. The "T7F" model has been implemented as the default queue service time model, based on its comprehensive modeling structure, and its experimental correlation with CORSIM. Queue service times are taken directly from the simulated uniform queue profile, which can be plotted using Spyglass and the Profile View screen. During this process, an additional adjustment is made to account for stochastic effects. Numerically computed queue service times are reported in the estimation output file "t7fact.out".

Green extension time models whose names begin with the letters "GP" were developed exclusively for use with TRANSYT. Models whose names begin with the letters "GL" are existing models from the literature. The "GP1" model has been implemented as the default green extension time model, based on its modeling structure and its experimental correlation with CORSIM. Green extension times are computed using applied probability analysis of the simulated input flow rate profile, which can be plotted using Spyglass and the Profile View screen. During this process, flow profile adjustments are made to account for stochastic effects. Numerically computed green extension times are reported in the estimation output file "t7fact.out".

The Critical Link Search Flag check box is toggled on by default to indicate that the updated actuated control methodology will be applied to each candidate link. When complex conditions and overlapping actuated phases prevail, the critical link search is necessary to obtain correct results. When simple conditions prevail, removing the search flag will not affect results, and will allow faster running times on the computer.

The Global Detector Length field is used to specify the default length of presence detectors throughout the network. The detector length value potentially affects the green extension time. If desired, link-specific detector lengths (that are not equal to the global default value) can be specified on the Edit > Optional > Actuated screen. In general, higher detector lengths result in longer actuated phase times and higher delays.

The Global Gap Setting field is used to specify the default gap setting (a.k.a. "unit extension") throughout the network. If desired, link-specific gap settings (that are not equal to the global default value) can be specified on the Edit > Optional > Actuated screen. In general, higher gap settings result in longer actuated phase times and higher delays. However, higher gap settings also reduce the probability of premature phase termination due to inattentive drivers. TRANSYT does not model driver

inattentiveness, and also does not model premature phase termination. According to conventional wisdom, a gap setting of 2 or 3 seconds is best for preventing premature phase termination without significant increases in delay.

The Convergence Override Flag check box is toggled off by default to indicate that convergence will be achieved when the signal timing plan appears to be stable. If this box is checked on, then the user must specify the desired number of actuated model iterations to be computed, using the Number of Iterations Override field. The only reason to manually code the number of iterations would be to reduce program running time, but this might compromise accuracy of the results.

(Note: The actuated estimation model requires TRANSYT step-wise simulation. If the estimation model is directed to process a file that specifies link-wise simulation, the file will automatically be converted to request stepwise simulation.)

### Estimation Report

Whereas legacy TRANSYT simulation output files typically have the "\*.TOF" extension, the estimation output filename has a \*.OUT extension. The output file report is a text file that contains the following information:

- run title, input filename, and file path
- user-selected queue service time and green-extension time models
- intersection or node numbers, and phase numbers
- critical link labels or numbers
- queue service time (*gs*)\*\* and green extension time (*ge*)
- minimum and maximum green time; minimum and maximum phase time
- model-estimated phase time
- offset value and offset reference phase
- queue calibration factor (*f<sub>q</sub>*)
- intermediate and final signal timing plans
- intermediate and final average vehicle delay reported by TRANSYT

\*\*It is sometimes not possible to obtain queue service times when oversaturated conditions prevail. If the queue is not served during the available green time, this is known as a "phase failure", and the queue service time will be reported as "999" in these cases. An actuated phase will be driven to its maximum green or force-off point in these cases.

A sample output report is listed below.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                      Analysis Time Period:  
Jurisdiction:  
Run Title:    Four Node Sample File  
File Name:    FourNode.tin

Queue Service Time Model: T7F  
Green Extension Time Model: GP1  
Global Gap Setting (sec): 3  
Global Detector Length (ft): 30

Iter	Node	Phase	Durations	Control Delay
1	1	15 30 15 40 0 0 0	54.3	
	2	15 30 15 40 0 0 0		
	3	15 30 15 40 0 0 0		
	4	15 30 15 40 0 0 0		

2	1	14	36	14	36	0	0	0	46.5
	2	14	36	14	36	0	0	0	
	3	14	36	14	36	0	0	0	
	4	14	36	14	36	0	0	0	
3	1	13	39	13	35	0	0	0	44.5
	2	13	39	13	35	0	0	0	
	3	13	39	13	35	0	0	0	
	4	13	39	13	35	0	0	0	
4	1	13	39	13	35	0	0	0	44.4
	2	13	39	13	35	0	0	0	
	3	13	38	13	36	0	0	0	
	4	13	39	13	35	0	0	0	
5	1	13	39	13	35	0	0	0	44.5
	2	13	39	13	35	0	0	0	
	3	13	39	13	35	0	0	0	
	4	13	39	13	35	0	0	0	
6	1	13	39	13	35	0	0	0	44.4
	2	13	39	13	35	0	0	0	
	3	13	38	13	36	0	0	0	
	4	13	39	13	35	0	0	0	

Convergence has been achieved

To obtain detailed output, process the punch file:

FourNode.pun

Node Split #1 Split #2 Split #3 Split #4

	Init	Final	Init	Final	Init	Final	Init	Final
1	15	13	30	39	15	13	40	35
2	15	13	30	39	15	13	40	35
3	15	13	30	38	15	13	40	36
4	15	13	30	39	15	13	40	35

Node 1 (Iteration 1) University and 13th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	60	5	1.00	4.56	3.06	5	14	15	10	
Non-Act	75	5	0	0	0	5	36	70	30	
NB Left	5	5	1.00	4.56	3.06	5	14	15	10	0
NB Thru	20	5	0.98	22.2	5.25	5	36	40	10	

Node 2 (Iteration 1) University and 34th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	60	5	1.00	4.81	3.00	5	14	15	10	
Non-Act	75	5	0	0	0	5	36	70	30	
NB Left	5	5	1.00	4.56	3.06	5	14	15	10	0
NB Thru	20	5	0.98	22.2	5.24	5	36	40	10	

Node 3 (Iteration 1) Archer and 13th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	60	5	1.00	4.56	3.06	5	14	15	10	
Non-Act	75	5	0	0	0	5	36	70	30	
NB Left	5	5	1.00	4.67	3.00	5	14	15	10	0
SB Thru	20	5	0.98	22.2	4.28	5	36	40	10	

Node 4 (Iteration 1) Archer and 34th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	60	5	1.00	4.81	3.00	5	14	15	10	
Non-Act	75	5	0	0	0	5	36	70	30	
NB Left	5	5	1.00	4.67	3.00	5	14	15	10	0
SB Thru	20	5	0.98	22.2	4.28	5	36	40	10	

{...intermediate outputs for iterations 2 through 5 are omitted here...}

#### Node 1 (Iteration 6) University and 13th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	53	5	1.00	4.62	3.11	5	13	15	10	
Non-Act	66	5	0	0	0	5	39	70	30	
NB Left	5	5	1.00	4.62	3.06	5	13	15	10	0
NB Thru	18	5	1.01	24.5	5.85	5	35	40	10	

#### Node 2 (Iteration 6) University and 34th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	53	5	1.00	4.81	3.00	5	13	15	10	
Non-Act	66	5	0	0	0	5	39	70	30	
NB Left	5	5	1.00	4.62	3.06	5	13	15	10	0
NB Thru	18	5	1.01	24.5	5.83	5	35	40	10	

#### Node 3 (Iteration 6) Archer and 13th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	54	5	1.00	4.62	3.11	5	13	15	10	
Non-Act	67	5	0	0	0	5	39	70	30	
NB Left	5	5	1.00	4.65	3.00	5	13	15	10	0
SB Thru	18	5	1.00	24.0	6.15	5	35	40	10	

#### Node 4 (Iteration 6) Archer and 34th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	53	5	1.00	4.81	3.00	5	13	15	10	
Non-Act	66	5	0	0	0	5	39	70	30	
NB Left	5	5	1.00	4.65	3.00	5	13	15	10	0
SB Thru	18	5	1.01	24.5	5.85	5	35	40	10	

Critical link labels (or numbers) indicate the links that tend to extend the actuated phase, and thus govern the length of the phase. More thorough descriptions of the output parameters above are summarized in Appendix A [37], and detailed in the supplemental research report "Modeling Traffic-Actuated Control with TRANSYT-7F" [36].

In the case of a project involving offset optimization, results within the estimation report may not be the final results. Once the actuated phase times have been computed by the estimation model, these phase times may be held constant during a subsequent offsets-only optimization run. A punch file (e.g., "filename.pun") is automatically generated by the estimation model, and this punch file contains the final actuated phase times.

Optionally, additional estimation runs and optimization runs can be performed after completing the first optimization. Since the queues and flow profiles generated by step-wise simulation are affected by the offsets, any changes in the offsets could affect the actuated phase times.

### Determining Maximum Green Settings

As stated in the Traffic-Actuated Control section, the actuated estimation model can be used to determine maximum green times for implementation in the field. However, alternative guidelines developed by the Institute of Transportation Studies (ITS) in Berkeley, California on using TRANSYT outputs to develop timing plan parameters for actuated controllers, are given below [26]. In this discussion, the term "phase" is used to indicate a phase combination in the pre-timed or single-ring actuated sense, not as an individual traffic movement or NEMA phase (model). Using these guidelines, the maximum green times for each actuated phase can be determined as follows:

1. Determine the critical link, the one with the highest degree of saturation that moves within an actuated phase.
2. The TRANSYT-optimized (or estimated) phase time can be set as the maximum for an actuated phase if the degree of saturation for the critical movement is less than 85%.
3. If the degree of saturation is 85% or higher, the maximum green is computed as follows:

$$G_{\max} = G + \frac{3600X^2}{4(1-X)s}$$

where  $G_{\max}$  = maximum green time (sec)  
 $G$  = TRANSYT-optimized green time (sec)  
 $X$  = degree of saturation (or v/c ratio)  
 $s$  = saturation flow rate of the critical link (vphg)

4. Repeat steps 1-3 for each actuated phase at the intersection.

The non-actuated minimum green time ( $G_{\min}$ ) can be determined as:

$$G_{\min}^n = C - G_{\max_i}$$

The equation in step 3 for determining maximum green times applies for  $X < 1$ . If the predicted degree of saturation is greater than or equal to 1, then the TRANSYT timings should be adjusted to avoid saturated movements (assuming that the basic input data have been already thoroughly checked). Also, pre-timed or fully-actuated operation at the intersection may be warranted.

Finally, check to verify that the computed maximum green times are greater than or equal to the minimum phase lengths required for pedestrians and vehicles. If not, usually at low-volume approaches, they should be set to the minimum phase lengths.

This method has been tested on limited data sets with good results [38]. It prevents the premature termination of green on actuated phases with heavy volumes, and provides reasonable maximum green times for uncoordinated operation. For more details on these procedures, as well as complete guidelines on using TRANSYT developed timing plans for actuated controllers, see reference [26].

## Arrival Type

This output parameter is based on the green (g/C) ratio, and the percentage of vehicles arriving on green simulated by TRANSYT. In other software packages, arrival type is sometimes an input parameter instead. Arrival type is reported for every single link in the network. The value ranges between 1 and 6. Higher values of the arrival type indicate better progression, whereas lower values indicate poor progression. By definition, the arrival type is 3 on external links (having no nearby upstream signal).

The HCM2000 [12] defines the following six arrival types:

- 1 = poor progression
- 2 = unfavorable progression
- 3 = random arrivals or minimal progression
- 4 = favorable progression
- 5 = very good progression
- 6 = exceptional progression

The percentage of vehicles arriving on green (PVG) output is a more precise measurement, and is available on the printed flow profile diagrams. In addition, the quality of progression can be viewed graphically by plotting the input flow rate profile, using the Profile View screen.



To quantify system-wide progression, "efficiency" and "attainability" are very helpful measurements. Efficiency and attainability are available in the TRANSYT legacy (\*.TOF) report, and are also displayed on the Time-Space Diagram screen.

## Artplan Import Methodology

The number of thru lanes and exclusive left-turn lanes is imported for each approach. If exclusive right-turn bays exist at a given intersection, they are assumed to be 200 feet long in both major-street directions. The side-street number of thru lanes is set to 2 if the number of major-street thru lanes exceeds 1, or set to 1 if the number of major-street thru lanes equals 1. The side-street number of exclusive left-turn lanes is set to 1, and the side-street number of exclusive right-turn lanes is set to zero. Left-turn pocket lengths for both major-street directions are imported at each intersection. However, if the left-turn g/C ratio equals zero, all left-turn pocket lengths and/or number of exclusive left-turn lanes will be set to zero.

Saturation flow rate is imported for each intersection, and then multiplied by number of lanes for each turn movement to obtain adjusted saturation flow rates for each movement. Left-turn saturation flow rates are further multiplied by 0.95.

Thru movement volume is first imported for each approach. Left-turn and right-turn movement volumes are then obtained by multiplying thru movement volumes by the user-coded percentage of left-turns and percentage of right-turns. Side-street volume is then reset to 20% of the major-street volume for each turn movement. Off-peak direction volumes for all three turn movements are then reset based on the following formula:  $V_{\text{offpeak}} = (V_{\text{peak}} / DF_{\text{factor}}) * (1 - DF_{\text{factor}})$ .

The cycle length is imported at each intersection. The cycle length imported from intersection #2 is used as the background cycle length; other intersections with unequal cycle lengths are flagged as uncoordinated. Yellow times are set to 4 seconds for each phase. All-red times are set to 1 second for each phase. If the major-street left-turn g/C ratio is greater than zero, a timing plan is synthesized to include exclusive left-turn phases. Major-street left-turn green time is set equal to (major-street left-turn g/C ratio) \* (arterial cycle length). Major-street thru green time is set equal to (major-street thru g/C ratio) \* (arterial cycle length). Side-street left-turn green time is set equal to one-third of the green time that remains in the cycle, after having assigned the major-street green time. Finally, the side-street thru movements receive all remaining green time in the cycle. If the major-street left-turn g/C ratio is zero, a timing plan is synthesized to include one "green ball" phase on the major street, and one green ball phase on the side street. Major-street thru green time is set equal to (major-street thru g/C ratio) \* (arterial cycle length). The side-street movements receive all remaining green time in the cycle. After the timing plans have been synthesized, phase actuations and offset reference intervals are set for consistency with the imported "control mode".

Internal link free-flow speeds are imported at each roadway segment, in both major-street directions. External link free-flow speeds are imported from the arterial "posted speed".

The following data are also imported: link lengths, peak hour factor, percent heavy vehicles, arterial forward direction, "agency" information (analyst, organization, date, jurisdiction, project ID), cross street names, control mode.

## Attainability

According to Courage and Wallace [42], attainability is "a measure of how much of the maximum available green, as determined by the critical green splits in each direction, is actually used in progression". According to Chang and Messer [41], attainability is "the average fraction of the minimum

through movement greens used for progression". Attainability is calculated as follows:

$$\text{Attainability} = \frac{B_a + B_b}{G_a + G_b}$$

where  $B_a, B_b$  = bandwidth (sec.) in directions a and b, respectively, and  
 $G_a, G_b$  = shortest thru green (sec.) in directions a and b, respectively

Thus, attainability is defined as bandwidth divided by the shortest major street green. Attainability can be computed for the forward and reverse arterial directions simultaneously (see above formula), or computed for one arterial direction at a time (e.g.,  $B_a/G_a$ ). If attainability is 100%, this implies that the bandwidth cannot be increased without increasing the shortest green time available along the major street. For attainability, the PASSER II software originally recommended [41] the arterial progression evaluation criteria listed below.

1.00 – 0.99	"increase min thru phase"
0.99 – 0.70	"fine-tuning needed"
0.69 – 0.00	"major changes needed"

Attainability can only be computed when one or more progression routes have been defined in the input file. Progression routes are either coded graphically (using the Map View), or numerically (using the Edit > Optional > Route screen).

Attainability is displayed on the Time-Space Diagram screen, and is reported in the legacy TRANSYT output (\*.TOF) file. The computation of attainability in these two locations is slightly different, due to their definitions of effective green time, which affect both  $B_{a,b}$  and  $G_{a,b}$  alike. In the Time-Space Diagram screen, effective green time is calculated as (actual green – startup lost time + extension of effective green). The legacy report uses an older calculation where effective green time is calculated as (actual green + yellow change + all red clearance). Apart from efficiency and attainability, all other results from the legacy report are based on the more modern effective green time calculation. Also note that the on-screen attainability reflects user-coded green times, whereas the legacy report attainability sometimes reflects program-computed green times, due to the initial timing flag and/or optimization.

## Automatic Flow Balancing

The Edit > Feeders screen features "automatic flow balancing". Simply type in the upstream node number in the Feeding Link #1 field, and the software will calculate the required flows (based on the link-numbering scheme) and fill in the remaining link connection information fields, including all feeding links, flows, and cruise speeds. The automatic flow balancing feature applies the original Arterial Analysis Package (AAP) logic, where upstream turns on to the major street are assumed to continue through at the subject intersection, and turns off the major street at the subject intersection are assumed to have come from the upstream through movement. Whenever link connections are drawn graphically in the Map View, the Feeders screen is called so that automatic flow balancing can be applied.

The automatic flow balancing logic produces the most common type of traffic flow pattern. It assumes that vehicles do not turn onto the side street (a.k.a. minor street, cross street) at the downstream intersection if they originated on the side street at the upstream intersection. If a significant number of vehicles turn onto the side street at the downstream intersection after originating on the upstream side street, feeding flows should be coded manually on the Feeders screen.

It is important to note that the automatic flow balancing is simply an input data coding feature, and does not necessarily represent the exact traffic flow patterns occurring in any given network. Under many traffic flow patterns and geometric configurations, these default values would be appropriately revised by

the analyst. Values coded by the user will not be overridden by the automatic flow balancing feature. Instead, link connection information will only be suggested when it is clear that none has been coded yet.

The connection between upstream through movement and subject link through movement is a special case. If the user types in the upstream node number for the subject intersection's through movement, automatic flow balancing will recommend link connection information for up to 3 upstream feeding links. The following example (illustrated in the screen image below) assumes the TRANSYT-7F Standard link-numbering scheme. If the user specifies that eastbound link 105 feeds eastbound link 205, on the Feeders screen at node #2, then the software may fill in remaining data for link 105, 104 (upstream southbound left), and 109 (upstream northbound right). The Feeding Link #2 field will indicate the upstream left, whereas the Feeding Link #3 field will indicate the upstream right. Note: on this screen is necessary to press the "Next Data" button to access the data for feeding links #3 and #4.

**TRANSYT-7F**

File Edit View Run Window Options Help

Output Quick Jump

FourNode.tin

**Link Connections at University and 34th**

Eastbound			Westbound			Northbound			Southbound	
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru
206	205	211	208	207	212	202	201	209	204	203

Upstream Node Number, or Feeding Link #1

105	105	105	0	0	0	0	0	0	0	0
-----	-----	-----	---	---	---	---	---	---	---	---

Feed Volume (vph)

89	422	264	0	0	0	0	0	0	0	0
----	-----	-----	---	---	---	---	---	---	---	---

Cruise Speed (mph)

45	45	45	0	0	0	0	0	0	0	0
----	----	----	---	---	---	---	---	---	---	---

Feeding Link #2

0	104	0	0	0	0	0	0	0	0	0
---	-----	---	---	---	---	---	---	---	---	---

Feed Volume (vph)

0	50	0	0	0	0	0	0	0	0	0
---	----	---	---	---	---	---	---	---	---	---

Cruise Speed (mph)

Apart from the special case (of an upstream through movement connecting to the subject link through movement), when the upstream link is coded, automatic flow balancing will simply calculate and fill in flows for the single upstream link (Feeding Link #1) only. Default cruise speeds, from the GUI menu selection Options > Cruise Speed, are automatically filled in as well.

There are situations where the upstream feeding volume may remain unchanged from 0, even after specifying the upstream link or node number:

1. The links do not appear to connect to one another, based on the user-specified link-numbering scheme.
2. The upstream feeding link represents multiple movements.
3. The upstream feeding volumes are significantly unequal to the downstream approach volume.

In TRANSYT, shared lanes are sometimes represented by only a single link number. In this case, the analyst could manually specify the amount of feeding traffic from an upstream shared lane, because the automatic flow balancing logic only recognizes upstream turning movements within exclusive lanes or links.

If "non-standard" traffic flow patterns prevail, the user is encouraged to override the automatic flow balancing information. An example of non-standard traffic flow patterns is when cross street vehicles turn on to the major street, and immediately turn off from the major street at the next downstream intersection. In addition, if non-standard geometric configurations prevail, the feeding link information may need to be customized by the user. Examples of non-standard geometric configurations include a shared lane adjacent to an exclusive lane, or a northbound through movement that feeds an eastbound through movement due to roadway curvature.

Note that feeding link flows may vary between time periods when performing multi-period simulation. The automatic flow balancing logic supports multiple time periods. Within the GUI, there is always a certain "active" time period that is currently selected. The active time period can be changed by using the combo box at the top of the GUI, or by going to Options > Time Period > Select. The Feeders screen will always display the feeding link volumes for the active time period, and will always apply automatic flow balancing for the active time period. When coding link connections in the Map View, this invokes automatic flow balancing for all time periods of a multi-period input file.

### Methodology

The following is a list of computational steps applied by the GUI to implement automatic flow balancing. In this list, feeding volumes are computed for the "subject" intersection (i.e., "downstream" intersection), as a function of turning movement volumes at the subject and "upstream" intersections. This list will be followed by numerical and graphical examples.

1. Determine the three turning movement volumes at the upstream intersection.
2. Determine the three turning movement volumes at the downstream (subject) intersection.
3. Adjust the downstream turning movement volumes as a function of mid-block source volumes. The summation of mid-block source volumes and feeding volumes should always be equal to the downstream volumes.
4. It is assumed that only the upstream through movement can send volume into turning movements at the subject intersection; no other movements contribute any volume.
5. For left-turns at the subject intersection, compare the left-turn volume here with the quantity ((upstream through volume) - (right-turn volume here)); whichever quantity is lower becomes the feeding volume into this movement.
6. For right-turns at the subject intersection, compare the right-turn volume here with the quantity ((upstream through volume) - (left-turn volume here)); whichever quantity is lower becomes the feeding volume into this movement.
7. For through movements at the subject intersection, compare the through volume here with the upstream left-turn volume; whichever quantity is lower becomes the feeding volume into this movement from the upstream left-turn movement.
8. For through movements at the subject intersection, compare the through volume here with the upstream right-turn volume; whichever quantity is lower becomes the feeding volume into this movement from the upstream right-turn movement.
9. For through movements at the subject intersection, determine the "remaining through volume" after

subtracting the amounts computed from steps 7 and 8. Proportionally distribute the upstream through volume as feeding volume among the remaining through volume, the downstream left-turn volume, and the downstream right-turn volume. The major street through volume should be appropriately distributed among all downstream turning movements.

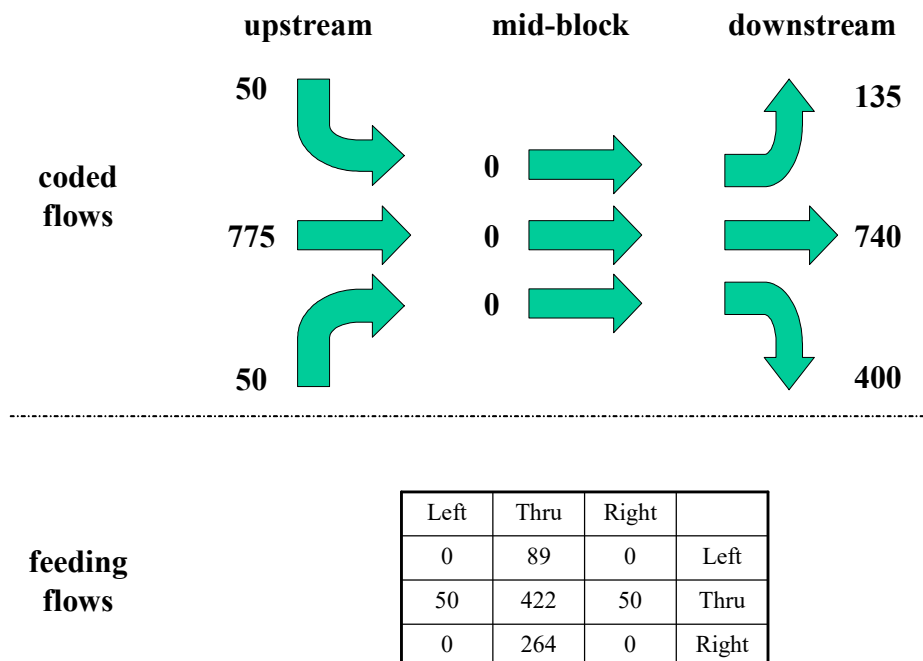
In some situations, the summation of upstream left-turn and right-turn volumes may exceed the downstream thru volume. For example, if every movement in the network were coded with 100 vph, the summation of upstream turning volumes (200 vph) would exceed the downstream thru volume (100 vph). In this situation, the nine-step automatic flow balancing methodology described above would not be used. Instead, the program would revert to oversimplified formulas for obtaining feeding flows:

$$\begin{aligned} & (\text{ThisLeftVolume} / \text{ThisApproachVolume}) * \text{UpstreamLeftVolume} \\ & (\text{ThisRightVolume} / \text{ThisApproachVolume}) * \text{UpstreamRightVolume} \\ & (\text{ThisThruVolume} / \text{ThisApproachVolume}) * \text{UpstreamThruVolume} \end{aligned}$$

Note that these steps are not applied by the traffic flow model within the simulation engine (T7FWin2.exe). The TRANSYT macroscopic simulation model balances flows in a different manner. External link traffic flows are always assumed to form a uniform flow profile. On internal links, if the summation of the feeding flows is not equal to the summation of the downstream turning movement volumes (minus the mid-block source volumes), the feeding flows will be proportionally factored up or down to match the downstream turning movement volumes. It is typical for the upstream and downstream flows to be not exactly balanced. The software issues a warning message when flows are significantly unbalanced, and may issue a fatal error message when flows are severely unbalanced.

#### Example #1

Example #1 illustrates automatic flow balancing between a pair of intersections with unbalanced flows. The "unbalanced flow" example is helpful, because both releases would generate the same feeding flow values if the upstream and downstream flows were balanced. In this case (illustrated below), the upstream intersection sends 875 vehicles per hour (vph) downstream. However, approach volumes at the downstream intersection have been measured at 1275 vph. In addition, there are no mid-block source volumes.

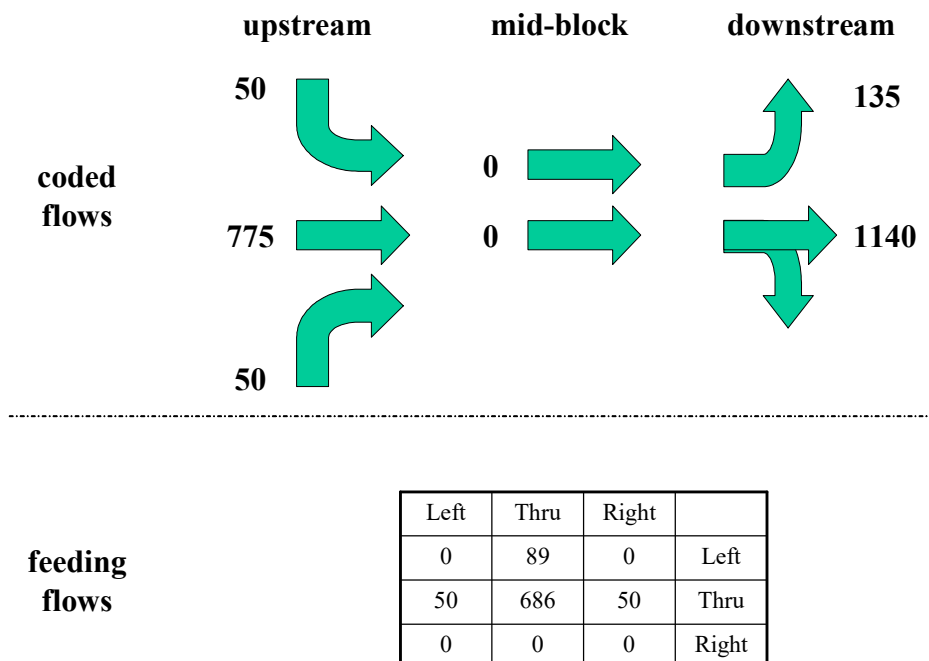


Automatic flow balancing computes the "remaining" downstream through traffic after deducting the upstream turning movements ( $740 - 50 - 50 = 640$  vph), and then finishes by proportionally distributing the upstream through traffic amongst the three downstream movements as follows:

remaining thru volume =  $740 - 50 - 50 = 640$  vph  
 downstream approach volume =  $640 + 135 + 400 = 1175$  vph  
 left feeding volume =  $(135 / 1175) * 775 = 89$  vph  
 thru feeding volume =  $(640 / 1175) * 775 = 422$  vph  
 right feeding volume =  $(400 / 1175) * 775 = 264$  vph

#### Example #2

Example #2 (illustrated below) is mostly similar to example #1 except that the downstream through and right-turn operate within a shared lane group.

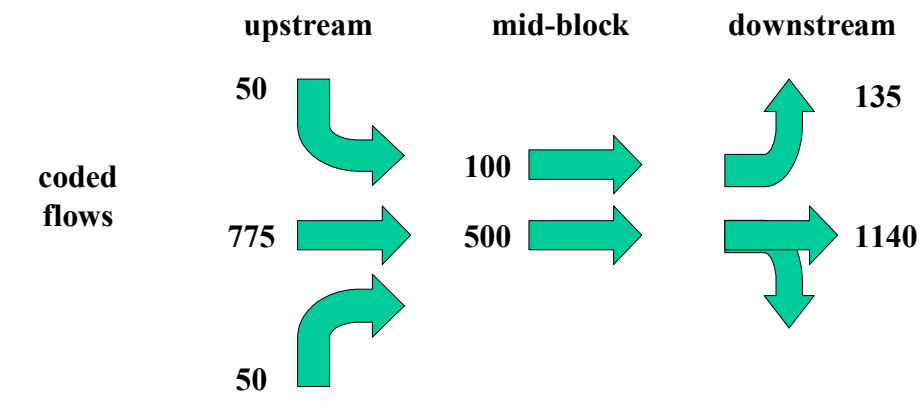


Automatic flow balancing computes the "remaining" downstream through traffic after deducting the upstream turning movements ( $1140 - 50 - 50 = 1040$  vph), and then finishes by proportionally distributing the upstream through traffic amongst the three downstream movements as follows:

remaining thru volume =  $1140 - 50 - 50 = 1040$  vph  
 downstream approach volume =  $1040 + 135 = 1175$  vph  
 left feeding volume =  $(135 / 1175) * 775 = 89$  vph  
 thru feeding volume =  $(1040 / 1175) * 775 = 686$  vph

### Example #3

Example #3 (illustrated below) is mostly similar to example #2 except that mid-block source volumes of 100 vph and 500 vph have been coded for the downstream left-turn and through movements, respectively. The automatic flow balancing logic is able to account for mid-block source volumes.



**feeding  
flows**

Left	Thru	Right	
0	47	0	Left
50	728	50	Thru
0	0	0	Right

Automatic flow balancing begins by adjusting the downstream turning movement volumes, based on the mid-block source volumes (e.g.,  $1140 - 500 = 640$  vph for the thru movement). It then proceeds to compute the "remaining" downstream through traffic after deducting the upstream turning movements ( $640 - 50 - 50 = 540$  vph), and then finishes by proportionally distributing the upstream through traffic amongst the three downstream movements as follows:

downstream left progressed volume =  $135 - 100 = 35$  vph  
 downstream thru progressed volume =  $1140 - 500 = 640$  vph  
 remaining thru volume =  $640 - 50 - 50 = 540$  vph  
 downstream approach volume =  $540 + 35 = 575$  vph  
 left feeding volume =  $(35 / 575) * 775 = 47$  vph  
 thru feeding volume =  $(640 / 575) * 775 = 728$  vph

Mid-block source volumes are thus considered along with the upstream progressed volumes, when determining the feeding link flows. The net effect of adding the mid-block flows was to shift a higher proportion of progressed flow over to the downstream through movement. During simulation, the traffic flow model will factor down the feeding flows to match the downstream flows (minus the mid-block flows), as described earlier.

#### Link Balancing Bias

The TRANSYT traffic flow model adjusts the upstream source flows to sum to the total volume coded (on the Edit > Traffic screen, or record type 28 field 5) for any given link, subtracting out any coded mid-block source flows. This is to account for variations in data collection, as well as mid-block sinks. Ordinarily, this feature is a time saver, because it avoids the need to balance flows manually.

In some cases, however, this can lead to anomalies, as documented by Hadi [25]. For example, consider the case where the coded input flow (on the Edit > Feeders screen, or record type 28) from an upstream link is significantly larger than the total flow coded for that link (on the Edit > Traffic screen, or record type 28 field 5). The adjusted input flow could make the arrival pattern higher than the downstream saturation flow rate, thus erroneously adding false delay and stops. If such a differential is



really due to mid-block sources, it is important that these be coded on the Edit > Traffic screen (or record type 28 field 6).

## Average System Speed

The average speed is an indication of the overall quality of flow in the network. It is simply the ratio of total travel (TT) and total travel time (TTT). It is expressed in units of vehicle miles (vehicle kilometers) divided by vehicle hours, resulting in miles per hour (kilometers per hour). This calculation excludes any link coded with a zero length (e.g., external links) or zero stop and delay weights (e.g., non-vehicular links).

$$\text{Avg. System Speed} = \frac{\text{total travel}}{\text{total travel time}}$$

## Average Travel Time

This is the total time spent on the link by the average vehicle. It is calculated by adding the average delay to the free flow travel time, where free flow travel time is based on the cruise speed on internal links (Edit > Feeders screen) and the cruise speed on external links (Edit > Optional > Global > Network Parameters screen).

## Back of Queue Algorithm

/\* the back of queue is calculated jointly for all links in the shared group by summing their IN and GO patterns \*/

IF this is the first step in the simulation initialization period THEN

```
INITIALIZE max_wave = 0
INITIALIZE min_wave = 0
INITIALIZE current_wave = 0
INITIALIZE gap_available = 0
INITIALIZE front_queue_status = stationary
INITIALIZE back_queue_status = stationary
SET optimal_distance_headway = 5280 / optimal_density
/* optimal density defaults to 70 veh/mi/lane */
SET jam_distance_headway = 5280/jam_density
/* jam density defaults to 211 veh/mi/lane */
INITIALIZE back_of_queue (ft) = initial queue (veh) ! jam-distance-headway/No. of lanes
```

END IF

IF the back of queue in the previous step = 0 THEN

IF GO pattern = 0 OR IN pattern < GO pattern THEN

```
front_queue_status = stationary
back_queue_status = stationary
max_wave = 0
min_wave = 0
current_wave = 0
/* note that at any point in time there might be several shock waves moving backward. "current
wave" refers to a wave moving backward with the movement that initiates that shock wave
```

continues from the stopline. max\_wave and min\_wave are the number of the first and last waves currently moving backwards toward the BOQ \*/  
total\_gap = 0

IF (GO pattern = 0) THEN

back of queue = IN pattern ! jam distance headway / No. of lanes

END IF

ELSE

front\_queue\_status = moving  
back\_queue\_status = moving  
max\_wave = 1  
min\_wave = 0  
current\_wave = 1  
wave\_status for max\_wave = not reached back of queue

END IF

END IF

IF front\_queue\_status = stationary and Go pattern > 0 THEN

INCREMENT max\_wave by 1  
SET front\_queue\_status = moving  
SET move\_time for max\_wave = 0  
SET wave\_position for max\_wave = 0  
SET current\_wave = max\_wave  
SET wave\_status for max\_wave = not\_reached\_back\_queue

END IF

IF front\_queue\_status = moving AND GO pattern = 0 then

front\_queue\_status = stationary  
SET current\_wave = 0

END IF

For wave\_i = min\_wave to max\_wave

IF wave\_status for wave\_i = reached\_back\_queue AND wave\_i = current\_wave THEN

COMPUTE gap\_available (ft) = gap\_available + GO pattern ! optimal\_distance\_headway/No. of lanes

ELSE IF wave\_status for wave\_i = not\_reached\_back\_queue THEN

IF wave\_i = current\_wave THEN

COMPUTE vehicles for wave\_i = vehicles for wave\_i + GO pattern

END IF

INCREMENT move\_time for wave\_i by 1

IF current link is permitted only THEN

SET saturation\_flow per lane = upper limit of maximum flow for the link

```

ELSE
    SET saturation_flow per lane = saturation flow coded by the user / No. of lanes
END IF

COMPUTE wave_speed (ft/hr) = 5280 * saturation_flow per lane / jam_density - optimal_density
COMPUTE wave_position for wave_i (ft) = move_time for wave_i ! wave_speed

IF wave_position for wave_i >= back_of_queue THEN

    SET wave_status for wave_i = reached_back_queue
    COMPUTE gap_available (ft/lane) = gap_available + vehicles for wave i !
    optimal_distance_headway/No. of lanes

    IF (wave_i) <> current_wave THEN

        INCREMENT min_wave by 1

    END IF

    IF back_queue_status = stationary THEN

        SET back_queue_status = moving

    END IF

END IF

ELSE IF the current_wave = 0 in this step AND it had already reached the BOQ

    INCREMENT min_wave by 1

END IF

NEXT wave_i

IF back_queue_status = moving THEN

    IF link is full THEN

        SET link is not full

    END IF

    COMPUTE reduction in back of queue = saturation flow per lane / optimal density
    /* optimal density = 1 / optimal distance headway */

    IF (back of queue < reduction in back of queue + (IN pattern ! optimal distance headway / No. of lanes)
    THEN

        back of queue = 0
        gap available = 0

    ELSE IF gap_available < saturation_flow ! optimal_distance_headway / No. of lanes THEN

        COMPUTE back_of_queue = back_of_queue - gap_available + IN pattern ! jam-distance-headway
        / No. of lanes
        SET gap_available = 0

        IF wave_status of current_wave = reached_back_queue

```

```

        SET back_of_queue = stationary
    END IF

    ELSE

        SET back_of_queue = back_of_queue - reduction in back_of_queue + IN pattern ! optimal distance
        headway / No. of lanes
        COMPUTE gap_available = gap_available - reduction in back of queue + IN_pattern !
        ((optimal_distance_headway - jam_distance_headway) / No. of lanes)

    END IF

END IF

IF back_queue_status = stationary THEN

    ADD (IN pattern ! jam_distance_headway / No. of lanes) to back_of_queue

END IF

```

## CORSIM Processing Methodology

CORSIM is sold separately as part of the TSIS package. TRANSYT-7F should be capable of locating CORSIM anywhere on the computer. If the software issues an error message saying that CORSIM cannot be located, TSIS should be launched at least once in standalone mode, so it can register itself on the computer.

If the user has a pre-existing CORSIM input file, this file can be imported by TRANSYT-7F for subsequent processing or optimization. CORSIM "processing" is a general term for post-processing, in which TRANSYT-7F can automatically perform one or more CORSIM runs and report the average results. These results could be presented in the form of a lengthy text file, a one-page intersection summary ("formatted report"), or by superimposing intersection delay and LOS on the Map View. In addition, the user could view a TRANSYT-7F time-space diagram of the CORSIM signal timings, without performing any CORSIM runs.

TRANSYT-7F can also generate or pre-process input data files for CORSIM. By selecting "File > CORSIM > Generate TRF file for this network", a CORSIM (NETSIM) input file can be generated from scratch, and copied to the user's chosen subdirectory on the computer. In the Map View, it is also possible to right-click on an intersection and select "CORSIM > Generate TRF file for this node". To instantly view CORSIM (TRAFVU) animation for the entire network, either click on the "CORSIM Animation" icon on the TRANSYT-7F toolbar, or select "File > CORSIM > View animation for this network". In the Map View, it is also possible to right-click on an intersection and select "CORSIM > View animation for this node". For more information, refer to the One-Touch CORSIM Animation section of the documentation.

TRANSYT-7F can also perform "direct" CORSIM optimization of cycle length, splits, and offsets. This is accomplished by managing the genetic algorithm from TRANSYT-7F, and running the CORSIM simulation engine in the "batch mode". As the optimization logic within TRANSYT-7F determines the new signal settings, new CORSIM input (\*.TRF) files are generated containing these settings. For more information, refer to the Direct CORSIM Optimization section of the documentation.

"Indirect" CORSIM optimization could be accomplished if the user optimized timing with traditional TRANSYT, and then exported the results back to CORSIM for subsequent simulation. Timing plans are exported in TRANSYT-7F by selecting "File > CORSIM > Export TRANSYT timing to CORSIM for this network", or by clicking the Export button for a given node in the Timing View screen. The primary

advantages of indirect CORSIM optimization might be faster program running times, the initial timing model, and phasing sequence optimization, none of which are available through direct CORSIM optimization. When performing direct CORSIM optimization, there is no need to export the timing back to CORSIM, because the punch file will already be in TRF format.

While performing a CORSIM processing or optimization run, the familiar CORSIM ticker screen will intermittently pop up in an adjacent "console" window that will be minimized and without focus. The user can switch focus to that window, if desired, to monitor the progress of individual simulation runs.

### CORSIM File Import Process

If a TRF file is opened, then a new TRANSYT-7F input (\*.TIN) file is automatically generated, in order to store the processing and optimization settings. If a CORSIM file is opened, but the corresponding TIN file already exists, the TIN file will be loaded automatically.

CORSIM and TRANSYT both contain somewhat similar data for signalized intersections within their input files. When a TIN file is automatically generated by opening a TRF file, only the data associated with coordinated, signalized intersections (nodes) are imported. Even though the number of imported nodes does not reflect unsignalized intersections or uncoordinated signals, their vehicle delays and trips can still be tabulated by the processor, reported in the output file, and may affect the optimization process whenever optimization is requested. This is explained later in the Output Report and Output Data section.

The number of nodes in the new TIN file will not include entry/exit nodes, dummy nodes, interface nodes, or freeway nodes specified within the TRF file. TRANSYT is dimensioned to handle up to 99 nodes, so the first 99 signalized and two-way stop (TWSC) intersections encountered in the TRF file can be imported. If desired, the user can choose (under Options > Preferences) not to import TWSC intersections or fully-actuated intersections. When performing coordination or optimization, better timing plans can be expected when limiting the data file size to separate groups of 20 nodes or fewer. The CORSIM "internal" node numbers, which can vary from 1 through 6999, are stored within the TIN file as "Intersection Name" comment records.

After the CORSIM file import process has been completed, most TRANSYT-7F Edit screens will be enabled for viewing the imported data. For example, it might help to verify that the CORSIM data are being displayed properly in the graphical Lane Configuration and Timing View screens. It might also help to confirm that directional orientation (north, south, east, west) is consistent between the TIN and TRF files. Because some links in TRANSYT must have a directional orientation, these directions are estimated as a function of the CORSIM node coordinates. Within the Map View, it might help to determine whether node positions were imported properly. Certain other features, such as the Profile View and Time-Space Diagram screens, are only available if the user "converts to macroscopic analysis" as explained later.

Certain CORSIM input data cannot be imported precisely into TRANSYT format. These issues usually have no effect on CORSIM processing or optimization results, but if the user wishes to perform a companion macroscopic analysis, any discrepancies should be corrected in the TRANSYT-7F Edit screens.

1. If the same turning movement can be made from both a shared lane and an exclusive lane, only the exclusive lane will be imported correctly. The shared lane will be imported as an exclusive thru lane.
2. If a CORSIM link has turn pockets with no thru lane, this will not be imported properly.
3. Left-turns that yield to opposing right-turns will be imported as protected-only links.
4. Gap settings for actuated controllers are automatically set to 3.0 seconds in the TIN file, and are not imported from the TRF file.
5. Hourly turning movement volumes are imported directly from TRF record type 21, regardless of whether these were originally coded as "turn volumes" or "turn fractions".
6. CORSIM input files that request traffic assignment sometimes do not contain the turning movement

counts required for link data to be fully imported. In this case, it may be preferable to instead import the "punch" TRF file, which contains turning movement counts generated by the traffic assignment process.

7. If two intersections are governed by one controller in CORSIM, this might not be imported properly.

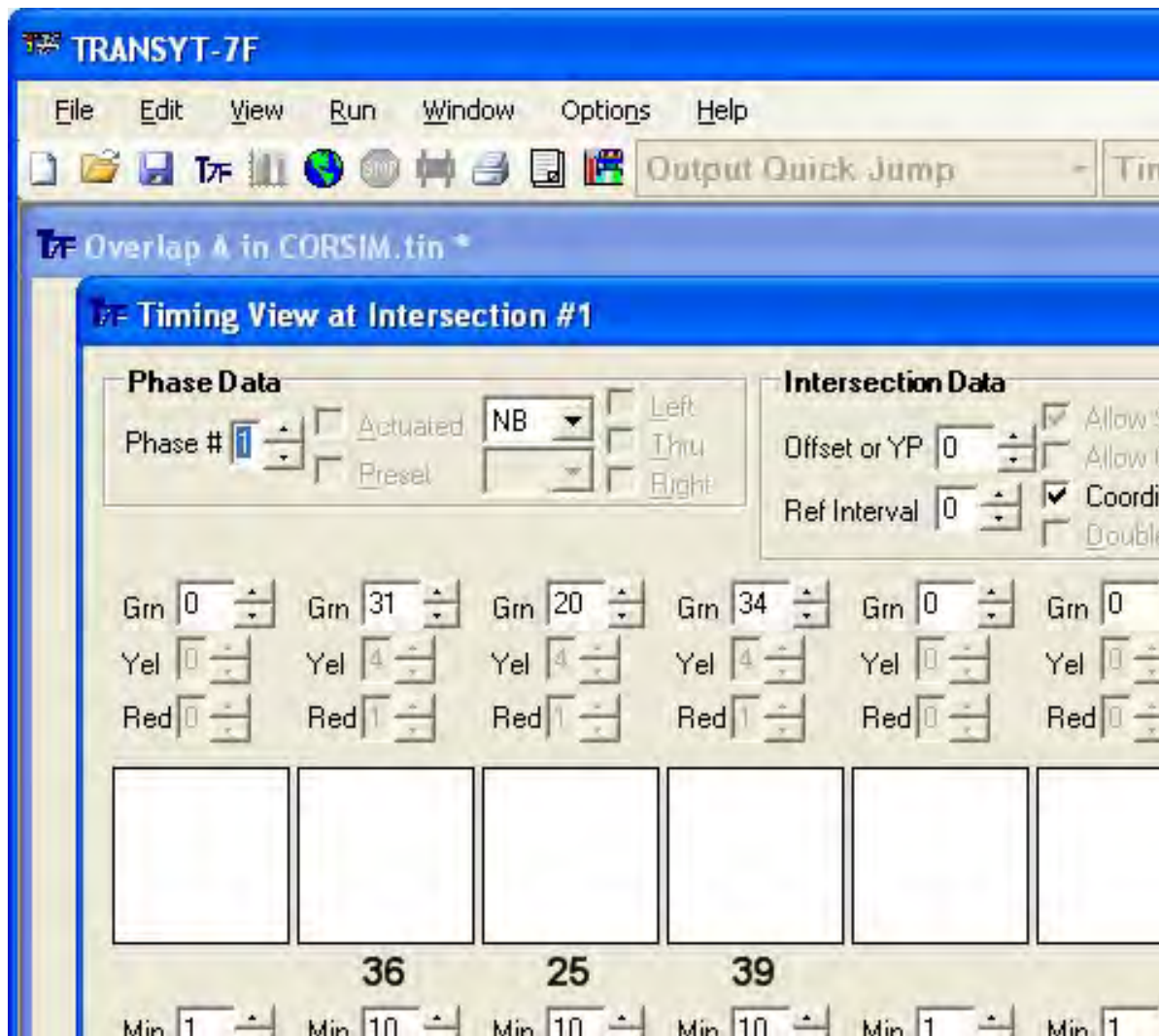
8. If the TRF file contains a coordinated actuated controller, the timing plan might not be imported properly if phases 1&5 or 3&7 (typically the left-turn phases) were coded as "lag phases".

9. If a CORSIM timing plan contains any yellow times coded as zero seconds, or an all-red clearance times coded as 6 seconds or higher, this will not be imported properly.

10. Signal timing plans with only one phase will not be imported properly.

11. At two-way stop-controlled intersections, shared lanes may be imported as exclusive lanes.

At fully-actuated intersections, the phasing sequence is imported in CORSIM's native dual-ring format. This differs from pre-timed and semi-actuated intersections, for which phasing sequences can be imported in TRANSYT's native single-ring format. Importing fully-actuated phasing sequences in dual-ring format has the disadvantage of preventing the user from "converting to macroscopic analysis", because the TRANSYT model requires single-ring timings. However, importing fully-actuated phasing sequences in dual-ring format has the advantage of allowing the user to specify minimum phase times (for up to 8 dual-ring phases) for direct CORSIM optimization. For direct CORSIM optimization applications, the Edit > Timing > Single-Ring screen should be used to specify minimum phase times. The Offset and Reference Interval are used to indicate maximum and minimum phase times for dual-ring phase #8, but only if phase #8 exists within the timing plan. The Coordinated Signal checkbox is turned on for the sole purpose of enabling the Offset and Reference Interval input fields. Movements for fully-actuated intersections are not displayed on the Edit > Timing > Single-Ring screen, which might look weird and unexpected. A more intuitive timing display can be obtained from the Edit > Timing > Dual-Ring screen. The screen image below shows an example of what the Edit > Timing > Single-Ring screen would look like with a fully-actuated intersection that was imported from CORSIM.



### Input Data

The primary purposes of the TIN file are to store the settings for direct CORSIM optimization (e.g., minimum phase times, objective function, etc.), and to facilitate a companion macroscopic (TRANSYT-based) analysis. Although the TRANSYT-7F Edit screens are enabled for viewing the imported data, changing this data will generally not affect the results of CORSIM processing. The exceptions to this are when modified timing plans are explicitly exported to the TRF file, or when the Edit > Analysis screen is used to adjust some basic settings. When timing is exported to CORSIM, this modifies the user's original TRF file, although a backup copy is generated. When the Edit > Analysis screen is used to adjust some basic settings, this updates the temporary TRF file used for CORSIM processing, but does not modify the user's original TRF file.

When using the TRANSYT-7F Edit screens during a direct microscopic analysis, certain input fields are automatically disabled to prevent possible problems when exporting timing to CORSIM, or when converting back and forth between microscopic and macroscopic analysis. For example, in the Lane Configuration screen, the number of lanes and lane usage cannot be changed. Several parameters (reference interval, phasing sequence, actuation, Y+AR clearance times) cannot be changed within the Timing View screen. The Edit > Optional > Intersection > Names screen is disabled, so that the CORSIM node numbers are not accidentally erased. However, the user can probably append Intersection Name

information beyond the CORSIM node number by using the record type editor. All of these input fields are enabled if the user converts to macroscopic analysis, as explained later.

The Edit > Analysis screen, which contains the relevant settings for CORSIM processing, is illustrated below. These Edit > Analysis settings are all relevant to direct CORSIM optimization as well, but several settings (e.g., optimization node list, allow offset optimization, etc.) on the other Edit screens have no effect without optimization. If desired, the user can modify the settings below to obtain different results from CORSIM processing. The Simulation Period and the Initialization Period are both imported directly from the TRF file.



The Random # Seed Variations field is used to specify the number of CORSIM runs to be performed during processing. When a TRF file is imported for the very first time, the Random # Seed Variations field is set to '1' by default. The processor automatically updates random number seeds within the CORSIM input file, and automatically tabulates average vehicle delays and vehicle trips from the multiple runs. An increased number of random number seed variations may increase the accuracy of the results, but will also increase program running times on the computer.

For any processing or optimization run, the first simulation run (for each candidate timing plan, if



optimization) will reflect the user-coded random number seeds in the TRF file. If the Random # Seed Variations on the Edit > Analysis screen is coded as 2 or higher, the second simulation run uses random number seeds of '1', '3', and '7'. The third simulation run uses random number seeds of '11', '13', and '17', the fourth simulation run uses random number seeds of '21', '23', and '27', and so forth. Therefore, any optimization result can always be easily reproduced by loading the punch file ("Genetic.trf") and performing a processing-only run, with the same number of Random # Seed Variations.

The Volume Multiplier field is used to automatically adjust traffic volumes defined within the CORSIM input file (\*.TRF, record type 50). When a TRF file is imported for the very first time, the Volume Multiplier field is set to '100' by default. This option may be useful when attempting to gauge the sensitivity of results to potential variations in traffic volume, or for analysis of future conditions.

The Lower Cycle Length setting has no effect on CORSIM processing results without optimization. However, the Lower Cycle Length on this screen and the Coordinated Signal checkbox on the Edit > Timing screen both affect the cycle length that is displayed on the formatted report. Without cycle length optimization, the Cycle Increment simply affects the Lower Cycle Length "spinner" control increment.

### Conversion Between Microscopic and Macroscopic Analysis

If the user selects File > Convert to Macroscopic Analysis, minor modifications are made to the TRANSYT-7F interface and the TIN file, to support a subsequent macroscopic analysis. Regarding the interface, all Edit screens and input fields are fully enabled, as mentioned earlier. The Profile View and Time-Space Diagram screens are also enabled, as is the ability to run the TRANSYT model. Regarding the TIN file, the objective function is set to Disutility Index, and the Initialization Period is set to the value of one local cycle length. The Analysis Type is set to '4', to indicate a linkage to CORSIM. To eliminate any linkage to the CORSIM file, one could use the record type editor to set the Analysis Type to '0', and use File > Save As to generate a unique file name.

At this stage, the ability to export signal timing plans from TRANSYT (TIN) to CORSIM (TRF) is disabled. The user could elect to perform optimization, load the punch file, use File > Save As to overwrite the original TIN file, and then select File > Convert to Microscopic Analysis. This series of steps would enable exporting of the optimized TRANSYT timings back to CORSIM. In order to convert between microscopic and macroscopic analysis, and to export timing to CORSIM, the TRF and TIN files must have the same basic filename (e.g., "MyFile.tin" and "MyFile.trf") and reside in the same computer folder.

If the user selects File > Convert to Microscopic Analysis, minor modifications are made to the TRANSYT-7F interface and the TIN file, to support a subsequent microscopic analysis. Regarding the interface, some Edit screens and input fields are disabled, as mentioned earlier. The Profile View and Time-Space Diagram screens are also disabled, as is the ability to run the TRANSYT model. Regarding the TIN file, the objective function is set to NETSIM Control Delay, and the Initialization Period is set to the value in the TRF file. The Analysis Type is set to '3', to indicate direct CORSIM processing or optimization.

### Output Report and Output Data

Genetic algorithm optimization punch files for CORSIM are called "Genetic.trf". Whereas standard CORSIM output files typically have the "\*.OUT" extension, the CORSIM processing output filename has a "\*.TXT" extension.

The CORSIM processing output file reports network-wide results as well as link-specific results. Network-wide results will automatically reflect all links and nodes, including unsignalized nodes, uncoordinated nodes, exit links, and the entire FRESIM subnetwork. By default, link-specific results are provided for the approach links going into coordinated NETSIM signals. Alternatively, link-specific results can be provided for all NETSIM links whose Control Delay exceeds 0.1 seconds per vehicle, using a selection in the Options > Preferences screen. These options are in place to prevent the simulation

results on unimportant links and nodes from diluting the analysis.

In CORSIM, queue delay is accumulated for each second when a vehicle has an acceleration of less than 2 ft/sec\*\*2, and a speed of less than 3 ft/sec [40]. Queue delay is accumulated every other second when a vehicle has an acceleration of less than 2 ft/sec\*\*2, and a speed between 3 ft/sec and 9 ft/sec. Total delay time is determined as the difference between actual travel time and the travel time if constantly moving at the free flow speed. Stop delay is accrued when a vehicle's speed is less than 3 feet per second on the link.

Control delay is measured from the simulation in accordance with Highway Capacity Manual [12] specifications. Intersection-wide level of service (LOS) is provided in the CORSIM processing output file for all coordinated signals. LOS is determined by taking a volume-weighted average of the control delays on relevant approach links, and then applying HCM thresholds to the intersection-wide average control delay. When comparing CORSIM LOS to the LOS estimates of other programs or models, it is helpful to keep the following issues in mind:

1. Start-up lost times for CORSIM should be set at 2.0 seconds for internal links, and 4.0 seconds for external links. Informal research has shown that setting the start-up lost times in this manner allows control delays from CORSIM to match control delays from the HCM/HCS more closely.
2. If during a given cycle a movement is oversaturated in CORSIM, vehicles on that movement will automatically use every second of yellow time as green time. In TRANSYT or HCS, this would be equivalent to setting the extension of effective green (EEG) time equal to the yellow time. Accepting the default EEG times (in TRANSYT or HCS) could cause their LOS estimates to be overly pessimistic.
3. If a movement is oversaturated at the end of simulation in CORSIM, vehicles remaining on the link will not have their delay estimated. This would naturally result in a more optimistic LOS estimate when compared to other programs. However, when conditions are clearly oversaturated in TRAFVU animation, or when the CORSIM legacy report shows a high number of phase failures for a given movement, it is usually preferable to assume LOS F.
4. When traffic-actuated control is in effect, it is crucial that the average phase times in CORSIM match those from the other program(s). Average phase times in CORSIM should be observed across several cycles of TRAFVU animation. If the average phase times are significantly different between programs, there is little chance of their results matching, and the input data from one or more programs may require calibration and/or correction.
5. The analysis period length should always match between programs, and the quality of platoon progression should always match between programs. Refer to the section on Verification and Validation for more discussion.

As stated in item #3 above, a vehicle trip is counted when a vehicle has completely traversed a link. If a vehicle does not completely traverse a link during simulation, delay is not counted for that vehicle. For this reason, vehicle trips (throughput) may be an effective optimization objective function for severely congested conditions.

A sample output listing is illustrated below.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                     Analysis Time Period:  
Jurisdiction:  
Run Title: Sample CORSIM processing output  
File Name: Actctrl.tin

CORSIM Version 6  
Random # Seed Variations: 1  
Volume Multiplier: 1  
Simulation Period (sec): 300  
Initialization Period (sec): 900

# Processing Results:

```

<-----NETSIM Approach Links-----> Network-Wide
Control Total Stop Queue Vehicle Percent Delay
Delay Delay Delay Delay Trips Stops Speed Time Speed
------(sec/veh)----- (%) (mph) (veh-hr) (mph)

```

```

8.3 10.4 7.1 7.4 2048 36.6 15.7 6.4 17.8

```

## Links

```

5.5 6.8 4.5 4.7 73 39.0 16.9 ( 375, 450)
12.5 12.9 11.5 11.6 26 73.0 13.3 ( 246, 450)
1.4 3.7 0.8 0.9 213 6.0 24.2 ( 375, 350)
0.5 1.4 0.6 0.6 79 2.0 20.3 ( 151, 350)
7.6 9.9 7.1 7.4 207 33.0 6.8 ( 350, 151)
27.2 28.5 23.8 24.7 126 80.0 7.8 ( 150, 151)
7.0 9.4 6.2 6.4 76 36.0 14.4 ( 152, 151)
3.6 6.4 2.7 3.0 290 25.0 16.5 ( 151, 152)
4.3 6.8 4.1 4.4 80 32.0 11.0 ( 255, 152)
24.9 26.4 22.0 22.7 62 87.0 9.0 ( 153, 152)
1.5 3.2 1.2 1.3 242 14.0 15.5 ( 152, 255)
7.0 8.8 6.1 6.2 29 82.0 16.2 ( 256, 255)
6.6 9.8 5.9 6.0 47 38.0 23.0 ( 190, 255)
33.0 34.1 31.4 31.7 18 83.0 6.8 ( 257, 255)
22.1 26.1 17.7 18.8 172 61.0 13.5 ( 255, 190)
9.6 10.0 8.9 9.2 11 63.0 18.7 ( 2, 190)
12.2 12.7 10.5 10.8 65 56.0 13.1 ( 1, 190)
4.4 5.2 3.2 3.6 201 36.0 24.1 ( 100, 450)
16.9 16.9 15.9 15.9 19 89.0 11.4 ( 245, 450)
15.7 16.6 14.8 15.0 12 100.0 11.8 ( 351, 350)

```

## Average Queue by Lane

```

Lane 1 Lane 2 Lane 3 Lane 4 Lane 5 Lane 6 Lane 7 Links

```

```

0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 375, 450)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 246, 450)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 375, 350)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 151, 350)
1.0 1.0 0.0 0.0 0.0 0.0 0.0 ( 350, 151)
2.0 4.0 0.0 0.0 0.0 0.0 0.0 ( 150, 151)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 152, 151)
1.0 0.0 0.0 0.0 0.0 0.0 1.0 ( 151, 152)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 255, 152)
2.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 153, 152)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 152, 255)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 256, 255)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 190, 255)
0.0 0.0 0.0 0.0 0.0 0.0 1.0 ( 257, 255)
4.0 3.0 0.0 0.0 0.0 0.0 0.0 ( 255, 190)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 2, 190)
0.0 1.0 0.0 0.0 0.0 0.0 0.0 ( 1, 190)
1.0 1.0 0.0 0.0 0.0 0.0 0.0 ( 100, 450)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 245, 450)
0.0 0.0 0.0 0.0 0.0 0.0 0.0 ( 351, 350)

```

## Maximum Queue by Lane

```

Lane 1 Lane 2 Lane 3 Lane 4 Lane 5 Lane 6 Lane 7 Links

```

```

3.0 3.0 0.0 0.0 0.0 2.0 3.0 ( 375, 450)
1.0 0.0 0.0 0.0 0.0 2.0 1.0 ( 246, 450)
3.0 3.0 0.0 0.0 0.0 0.0 0.0 ( 375, 350)
0.0 0.0 0.0 0.0 0.0 0.0 1.0 ( 151, 350)
6.0 6.0 0.0 0.0 0.0 0.0 2.0 ( 350, 151)
7.0 10.0 0.0 0.0 0.0 0.0 2.0 ( 150, 151)
2.0 4.0 0.0 0.0 0.0 0.0 3.0 ( 152, 151)
7.0 5.0 0.0 0.0 0.0 0.0 5.0 ( 151, 152)
2.0 2.0 3.0 0.0 0.0 0.0 0.0 ( 255, 152)
8.0 0.0 0.0 0.0 0.0 0.0 3.0 ( 153, 152)
4.0 3.0 0.0 0.0 0.0 1.0 2.0 ( 152, 255)

```

1.0	0.0	0.0	0.0	0.0	1.0	2.0	( 256, 255)
3.0	1.0	0.0	0.0	0.0	0.0	1.0	( 190, 255)
1.0	0.0	0.0	0.0	0.0	1.0	3.0	( 257, 255)
10.0	9.0	0.0	0.0	0.0	0.0	0.0	( 255, 190)
1.0	1.0	0.0	0.0	0.0	0.0	0.0	( 2, 190)
1.0	5.0	0.0	0.0	0.0	0.0	0.0	( 1, 190)
4.0	4.0	0.0	0.0	0.0	1.0	1.0	( 100, 450)
1.0	0.0	0.0	0.0	0.0	1.0	2.0	( 245, 450)
1.0	0.0	0.0	0.0	0.0	0.0	1.0	( 351, 350)

Average	Percent	Phase	Lane	
Occupancy	Storage	Failures	Changes	Links

2.2	5.2	0.0	8.0	( 375, 450)
1.7	6.2	0.0	0.0	( 246, 450)
7.1	10.9	0.0	50.0	( 375, 350)
0.8	4.3	0.0	0.0	( 151, 350)
4.8	32.7	0.0	8.0	( 350, 151)
9.1	14.3	0.0	15.0	( 150, 151)
2.5	6.3	0.0	12.0	( 152, 151)
7.0	17.6	0.0	72.0	( 151, 152)
1.8	6.9	0.0	56.0	( 255, 152)
4.7	16.8	0.0	0.0	( 153, 152)
3.1	13.3	0.0	13.0	( 152, 255)
1.6	4.9	0.0	0.0	( 256, 255)
2.6	2.0	0.0	12.0	( 190, 255)
2.0	6.2	0.0	0.0	( 257, 255)
15.1	15.1	0.0	32.0	( 255, 190)
1.0	1.4	0.0	0.0	( 2, 190)
3.1	7.0	0.0	4.0	( 1, 190)
9.7	9.3	0.0	27.0	( 100, 450)
1.6	5.4	0.0	0.0	( 245, 450)
1.1	4.3	0.0	0.0	( 351, 350)

Signalized Intersection Level of Service

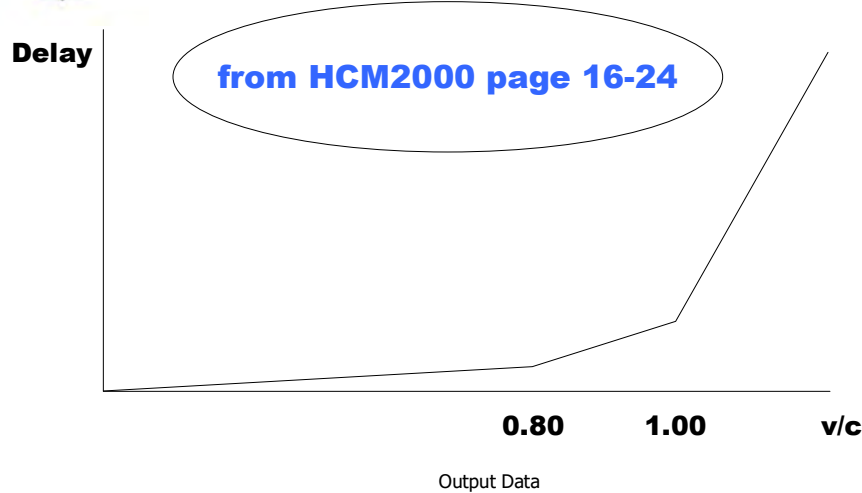
Node	Control Delay	LOS	CORSIM_ID
	(sec / veh)		

1	19.0	B	Intersection #190
2	1.7	A	Intersection #350
3	13.5	B	Intersection #151
4	6.8	A	Intersection #152
5	4.4	A	Intersection #255
6	6.1	A	Intersection #450

## Degree of Saturation

This is defined as the ratio of volume to capacity, where volume is the input flow rate demand specified by the user, and capacity is the amount of traffic that could be served under prevailing traffic network conditions. In addition to reporting the degree of saturation for each link, TRANSYT reports the total number of links with degree of saturation greater than 1.0. This measure of effectiveness provides an indication of the level of congestion experienced by the overall system or route. A positive value indicates the existence of congested conditions, and optimization objective functions suitable for these conditions could be applied. The Highway Capacity Manual [12] figure below illustrates the relationship between control delay and degree of saturation.

# Delay vs. Degree of Saturation



For protected (unopposed) movements, degree of saturation can be computed by hand, or using a calculator. Hand calculations can be used to verify results from the program if single-cycle simulation is used, or if conditions are undersaturated. If multi-cycle simulation is used to analyze oversaturated conditions, degrees of saturation may exceed the hand-calculated (or single-cycle simulation) values, due to residual queuing. Capacity is equal to saturation flow / cycle length, or  $c = s/(g/C)$ . Degree of saturation is equal to volume / capacity, or  $X = v/c$ . When the program reports degrees of saturation that exceed the hand-calculated value, this is often caused by queue spillback from downstream links.

## Delay and Level of Service

One of the most important measures of effectiveness in traffic studies is the delay to vehicles in the system. Delay represents indirect costs to the motorist in terms of lost time, discomfort and frustration, and a direct cost in terms of fuel consumption during idling. Excessive delay at signalized intersections reflects inefficiency in the signal timing.

In a field study, delay is usually measured by periodically counting the number of vehicles queued at a signal and integrating this series of counts over time. In TRANSYT, this histogram is known as a "queue profile". There are bound to be variations in the arrivals from cycle to cycle, so randomization must be taken into account, particularly near or at saturation, where the effect is far more severe. The user also can request multi-cycle analysis, which allows the queue profile to vary from cycle to cycle. When conditions are stable and undersaturated, the queue profile tends to remain constant between cycles. When conditions are unstable and oversaturated, the queue profile tends to reveal increasing queue lengths over time.

TRANSYT uses a delay estimate that combines these principles. Delay is the entire amount of time spent while not traveling at the prevailing cruise speed. Although TRANSYT does not explicitly trace the trajectories of individual vehicles to obtain these delay estimates, it can obtain macroscopic estimates of delay from the queue profiles. In addition, there may be differences in the number of stops for the

same value of delay, which is an important consideration in estimating fuel consumption.

Uniform delay is calculated by integrating the area under the uniform queue profile, which has the effect of averaging the queue length over the cycle. For example, in the Profile View screen below, the average uniform queue length value is 1.3 vehicles. This would indicate a uniform delay (d1) of 1.3 vehicle-hours. By multiplying by 3600 seconds and dividing by the hourly volume of 194 vph, uniform delay could be expressed as 24.1 seconds per vehicle.



TRANSYT exercises an algorithm to estimate uniform delay by examining the simulated queue length step-by-step. In effect, if  $m_t$  is averaged over the cycle, uniform delay is the average queue length times the cycle length, or:

$$D_u = \frac{\sum_{t=1}^N m_t}{N}$$

where,

$D_u$  = uniform delay in veh-hr/hr  
 $m_t$  = queue length in number of vehicles during step  $t$   
 $N$  = number of steps in the cycle

TRANSYT uses the Highway Capacity Manual 2000 delay model [12], but further uses macroscopic simulation results to allow this delay model to recognize complex traffic operations. Several delay model values (e.g., capacity, PVG,  $X_u$ ) are obtained from simulation, instead of user input. The model estimates average "control" delay, which includes initial deceleration delay, queue move-up time,

stopped delay, and final acceleration delay. The delay equation contains three terms and is illustrated below:

$$d = d_1 PF + d_2 + d_3$$

where,

- d1 = uniform delay (seconds per vehicle)
- PF = progression adjustment factor
- d2 = incremental delay (seconds per vehicle)
- d3 = residual demand delay (seconds per vehicle)

Since TRANSYT obtains uniform delay by integrating the simulated queue profile, it is not necessary for the program to estimate or apply the progression adjustment factor. In addition, because the TRANSYT step-wise simulation model explicitly carries the residual queue (if one exists) from one cycle to the next, the third term of the HCM delay model is also handled by the uniform delay estimation. Therefore, when TRANSYT reports uniform delay, this value automatically takes progression (PF) and residual demand (d3) effects into account.

By default, TRANSYT currently uses the Highway Capacity Manual 2000 second-term delay (d2) model, defined as follows:

$$d_2 = 900T \left\{ (X - 1) + \sqrt{(X - 1)^2 + \frac{8kIX}{cT}} \right\}$$

where,

- T = duration of analysis period in hours
- k = incremental delay factor
- I = upstream filtering and metering adjustment factor
- c = lane group capacity (vehicles per hour)
- X = lane group degree of saturation or v/c ratio

For pre-timed controllers, the k value is 0.50. For traffic-actuated controllers, the k value is computed as a function of the degree of saturation and kmin, using the following formula:

$$k = (1 - 2k_{\min})(X - 0.5) + k_{\min}$$

where X values lower than 0.5 are set equal to 0.5, and kmin is a function of the controller's unit extension, as given in the table below:

Unit Extension (seconds)	<= 2.0	2.5	3.0	3.5	4.0	4.5	5.0
kmin	0.04	0.08	0.11	0.13	0.15	0.19	0.23

The net effect of the k value is to reduce delay for actuated movements operating at low degrees of saturation. Low values of the unit extension allow for the quickest phase terminations upon "gap-out", and thus the lowest delays. However, beware that low unit extension values also increase the probability of premature phase termination (due to driver inattentiveness), which is not modeled by TRANSYT. Engineering judgment is required in selecting unit extensions that are high enough to prevent premature phase termination, but low enough to prevent wasted green times on the minor (actuated) movements. The default unit extension in TRANSYT is 3.0 seconds, but the user can change this value globally, or for individual links. Refer to the Actuated Screen documentation for more details.

As suggested in the Highway Capacity Manual, the I-factor value is 1.0 for isolated signalized intersections. For intersections with nearby upstream signals, the I-factor value is a function of the degree of saturation of the upstream feeding movements (Xu), defined as:

$$I = 1.0 - 0.91X_u^{2.68}$$

When the upstream degree of saturation is greater than 1.0, it is set equal to 1.0 for the purposes of this computation, which in turn constrains the I-factor to a minimum value of 0.09. TRANSYT examines the feeding volume from each feeding link, and computes a volume-weighted average degree of saturation. The net effect of the I-factor is to reduce delay for movements that have saturated upstream movements, resulting in minimal variation amongst the arriving traffic flow patterns. In other words, movements with unsaturated upstream movements are expected to experience a wider variety of traffic flow patterns, thus increasing the probabilities of phase failure at the downstream movement.

Although TRANSYT reports the total control delay in units of seconds per vehicle, incremental (a.k.a. random) delay  $d_2$  is expressed in units of veh-hr/hr after computation with the following formula:

$$d_2 = 900T \left\{ (X - 1) + \sqrt{(X - 1)^2 + \frac{8kIX}{cT}} \right\} \cdot \frac{v}{3600}$$

where  $v$  is the link volume.

Different second-term delay models have been used in previous versions of TRANSYT. These models are also available for selection if the user wants to see an alternative to the Highway Capacity Manual 2000 second-term delay model. They can be requested on record type 0, using the record type editor. In field 8, the user can code:

" BA"

where B and A represent the following numbers right-justified within the field:

A = equivalent lane option, code '0' for no or '1' for yes

B = second-term delay model, code '1' for the release 6 and prior model, '2' for the release 7 model, or '3' for the release 8.1 model

The release 6 (and earlier) model was:

$$d_2 = -\frac{B_n}{B_d} + \sqrt{\left(\frac{B_n}{B_d}\right)^2 + \frac{X^2}{B_d}}$$

where,

$B_n = 2(1 - X) + XZ$

$B_d = 4Z - Z^2$

$Z = (2X / v)(60 / T)$

$v$  = volume on the link

$X$  = degree of saturation, or  $v/c$

$c$  = capacity of lane group in vph, or  $c = sg/C$

$s$  = saturation flow rate in vphg

$C$  = cycle length in seconds

$g$  = effective green time in seconds

$T$  = period length in minutes

Release 7 used a functionally similar second-term delay originally from the Australian Road Research Board [16,33], defined as follows:



$$d_2 = 900TX^2 \left\{ (X-1) + \sqrt{(X-1)^2 + \frac{4X}{cT}} \right\} \bullet \frac{v}{3600}$$

Release 8.1 used a similar model adopted from the 1997 Highway Capacity Manual, defined as follows:

$$d_2 = 900TX^2 \left\{ (X-1) + \sqrt{(X-1)^2 + \frac{mX}{cT}} \right\} \bullet \frac{v}{3600}$$

where m is 2, 3, 4, 2, 2, or 1 for arrival types 1 through 6, respectively.

Control delay has the tendency to increase rapidly when a link is saturated, and this condition will strongly influence the measures of effectiveness, including that of fuel consumption. Such high delays should be viewed as gross estimates at best. Only a multi-period simulation could facilitate a defensible estimate of delays for significantly oversaturated conditions, and only if the start of the first period and the end of the last period were undersaturated.

TRANSYT simulates a constant, non-fluctuating signal timing plan within any given time period, and the measures of effectiveness should be considered as "average" for the period of analysis. Even traffic-actuated control is simulated with non-fluctuating phase times, but recent releases are more effective at estimating the average actuated phase times that will occur in the field (actuated estimation model), and are more effective at optimizing the actuated phase times (target degree of saturation model).

The second-term delay formula from the Highway Capacity Manual 2000, which is the default second-term delay applied by TRANSYT, includes a deterministic component and a random component. Beginning with release 8, the user can request multi-cycle step-wise simulation. In this case, a deterministic component (oversaturation delay) is considered explicitly when calculating the first-term delay. Thus, in multi-cycle step-wise simulation, a deterministic component ( $D_d$ ) defined below, is subtracted from the second-term delay:

$$D_d = \frac{(X-1)cT}{2}$$

Control delay is the sum of uniform, incremental, and residual delay components. TRANSYT only reports uniform and random delay, where uniform delay includes residual delay, and random delay represents incremental delay. Residual delay ( $d_3$ ) can only be simulated and computed when step-wise simulation has been selected. If the narrow (80-column) legacy output format has been selected on the Edit > Analysis screen, only the total control delay will be reported. If the wide (132-column) output format has been selected, the uniform and random delay components will be reported along with the total control delay.

The average delay per vehicle is the total control delay on the link divided by the total flow on the link. The control delay values represent the delays caused by control (signalized and unsignalized) only. Delay due to roadway topography or discretionary lane changing is not reported. The amount of delay affects fuel consumption (that is, idling consumes fuel non-productively), disutility index, and operating costs.

Uniform and residual delay can be computed based on the area under the uniform queue profile (i.e., queue.out from Spyglass). Incremental delay is computed by using the Highway Capacity Manual equation, where certain input parameters (e.g., capacity) are obtained directly from TRANSYT simulation, and other input parameters (e.g., duration of the analysis) are obtained directly from the input data file.

Level of service (LOS) is a qualitative measure used to describe the operational conditions within a traffic stream, and the perception of these operating conditions by motorists and/or passengers. Specifically, the LOS for signalized links and nodes is based on average control delay. To determine the LOS, TRANSYT first converts the computed delay into units of seconds per vehicle, and then uses the lookup table below from the Highway Capacity Manual 2000:

LOS	Operational Conditions	Control Delay (sec/veh)
A	Progression is extremely favorable. Most vehicles arrive during the green phase and do not stop.	<=10
B	Good progression with more vehicles stopped than with LOS A.	> 10 and <= 20
C	Fair progression with a significant number of vehicles stopping. Many vehicles still pass through the intersection without stopping. Individual cycle failures may appear.	> 20 and <= 35
D	Unfavorable progression with many vehicles stopped. Individual cycle failures are noticeable.	> 35 and <= 55
E	Poor progression with frequent individual cycle failures.	> 55 and <= 80
F	Poor progression with many individual cycle failures.	> 80

## Disutility Index

The disutility index (DI) is a measure of disadvantageous operation; that is, stops, delay, fuel consumption, etc. Unless the DI has specifically been defined as excess fuel consumption, its value has no intrinsic meaning, since it is simply a linear combination of delay and stops, whose units differ. An excess maximum back of queue penalty can optionally be included within the disutility index.

The "standard" delay and stops DI, which is one of the primary optimization objective functions within TRANSYT, is defined as follows:

$$DI = \sum_{i=1}^n \left\{ (w_{d_i} d_i + K w_{s_i} S_i) + U_i (w_{d_{i-1}} d_{i-1} + k w_{s_{i-1}} S_{i-1}) + QP \right\}$$

where,

DI = disutility index, analogous to the original TRANSYT performance index

di = delay on link i (of n links) and on an optional user-specified upstream input link i-1

K = user-coded stop penalty factor to express the importance of stops relative to delay (if coded as '-1', the coefficients are set internally so that the DI is equivalent to excess fuel consumption)

Si = stops on link i (and similarly for link i-1) per second

wxi = link-specific weighting factors for delay (d) and stops (s) on link i (and i-1)

Ui = binary variable that is '1' if link-to-link weighting has been established, zero otherwise (see the Weighting > Link-to-Link Weights screen)

QP = queuing penalty

The queuing penalty is used to minimize the possibility of spillback. If link-wise simulation is used to evaluate system performance after each optimization shift, this term is calculated as follows:

$$QP = QB_i W_q (q_i - qc_i)^2$$

where,

Q = a binary variable set by the user

'1' if the queue penalty is included in the DI  
 '0' otherwise  
 Bi = a binary variable  
 '1' if the maximum back of queue (qi) exceeds the queuing capacity  
 '0' otherwise  
 Wq = a network-wide penalty applied to the excess queue spillback  
 qi = computed maximum back of queue on link i  
 qci = queuing capacity for link i

For oversaturated conditions, step-wise simulation is recommended. Under step-wise simulation, the queuing penalty cannot be calculated as above because (qi - qci) is always zero. It is always zero because vehicles are not allowed to enter a full link. Thus, the queuing penalty for step-wise simulation is defined as follows:

$$QP = Q \bullet W_F \bullet F_i$$

where,

Fi = number of steps link i is full  
 WF = network-wide penalty applied to the number of steps the link is full

Regarding the queuing capacity (qci), this is either the default value calculated by TRANSYT, or a value coded by the user.

The default value for the disutility index (DI) is excess fuel consumption. Thus, the initial wxi in the DI formula are both internally set to values corresponding with the fuel consumption model. The wxi coefficients may also be modified by applying artery stop and delay factors.

Random stops can optionally be omitted from the DI. To request the omission of random stops from the DI, a value of '6' must be coded as the model number on the Edit > Optional > Global > Model Coefficients screen (or record type 8). However, note that random delay cannot be omitted from the DI.

## Double Cycling

It is common to double-cycle at minor intersections where traffic demand is light. This means to operate the local controller on a cycle length exactly half of the system cycle length. For example, if the system cycle length is 120 seconds, the local double cycle length is 60 seconds, and interval times coded on the Timing screen should sum to 60 seconds. Delays, and even possibly stops, will almost certainly be reduced. However, the phasing sequence cannot be optimized.

If progression is not needed, another option besides double-cycling would be to mark the intersection as "uncoordinated" on the Timing screen, and choose an appropriate isolated cycle length, possibly by Uncoordinated Optimization. If the data file contains only one intersection, this intersection should not be double-cycled. The master controller should not be double-cycled. If the Uncoordinated Network checkbox was turned on in the Edit > Analysis screen, none of the intersections should be double-cycled.

Coding double cycling is trivial:

1. Turn on the "Double Cycle" checkbox (Edit > Timing screen), or code a '1' in the Phasing Flag field (Edit > Optional > Timing screen), for the node to be double cycled. This input is associated with field 16 of record type 1X.
2. The user is limited to coding a maximum of three phases, because TRANSYT uses three more for the

second (half) cycle. Likewise, only code the minimum phase durations for one half cycle. To model a double-cycled node with four phases or more, simply code it as a single-cycled node, but with the phase durations doubled.

TRANSYT identifies nodes that are good candidates for double cycling in the legacy output report. If the degree of saturation is lower than a user-specified threshold (default 25%) a 'D' is printed after the maximum degree of saturation for the node. The double cycle threshold degree of saturation can be specified on the Edit > Optional > Global > Network Parameters screen (or record type 10).

If several such intersections are adjacent, the user should double cycle them to promote progression between them. A common use of double cycling is to tie-in a major artery, operating at a long cycle, with a lesser arterial that can operate more efficiently at the lower cycle. Progression between them is thus optimized. Double cycled nodes can be used with bandwidth constraints and progression opportunities (PROS), but only one progression band (the largest) will be identified in the Flow Profile Diagrams.

## Effective Green Time

Effective green time is an output parameter that is only provided by TRANSYT if the wide (132-column) legacy report format has been selected, and can be very useful in the calibration process. It is a function of the initial timing plan, as well as the start-up lost time (SLT), actual green time (G), and the extension of effective of green time (EEG). It is calculated as follows:

$$g = G - SLT + EEG$$

The Profile View screen can be used to observe effective green times graphically. By plotting the saturation flow rate profile, which is the GO-pattern from Spyglass (gopat.out), it is usually possible to determine when links are moving. Keep in mind that the saturation flow rate profile can be altered if the link is affected by queue spillback, or if the link is permitted and yielding to an opposing movement, or if the link is permitted and is discharging "sneakers".

## Efficiency

According to Chang and Messer [41], efficiency is "the average fraction of the cycle used for progression". Efficiency is calculated as follows:

$$\text{Efficiency} = \frac{B_a + B_b}{2 \times C}$$

where  $B_a, B_b$  = bandwidth (sec.) in directions a and b, respectively, and  
 $C$  = cycle length (sec.)

Thus, efficiency is defined as bandwidth divided by the cycle length. Efficiency can be computed for the forward and reverse arterial directions simultaneously (see above formula), or computed for one arterial direction at a time (e.g.,  $B_a/C$ ). For efficiency, the PASSER II software originally recommended [41] the arterial progression evaluation criteria listed below.

0.00 – 0.12	"poor progression"
0.13 – 0.24	"fair progression"
0.25 – 0.36	"good progression"
0.37 – 1.00	"great progression"

Efficiency can only be computed when one or more progression routes have been defined in the input file. Progression routes are either coded graphically (using the Map View), or numerically (using the Edit > Optional > Route screen).

Efficiency is displayed on the Time-Space Diagram screen, and is reported in the legacy output (\*.TOF) file. The computation of efficiency in these two locations is slightly different, due to their definitions of effective green time, which affect both  $B_a$  and  $B_b$  alike. In the Time-Space Diagram screen, effective green time is calculated as (actual green - startup lost time + extension of effective green). The legacy report uses an older calculation where effective green time is calculated as (actual green + yellow change + all red clearance). Apart from efficiency and attainability, all other results from the legacy report are based on the more modern effective green time calculation. Also note that the on-screen efficiency reflects user-coded green times, whereas the legacy report efficiency sometimes reflects program-computed green times, due to the initial timing flag and/or optimization.

## Excess Operating Cost

Although TRANSYT estimates the total operating cost, it can also explicitly optimize "excess" operating cost as follows:

$$CI = \sum_{i=1}^n \left\{ \frac{k_{i2}S_i + C_d d_i}{1000} + C_f f_i + OC_i \left( \frac{TT_i}{V_i} + d_i \right) \bullet I + QP \right\}$$

where CI is the cost index, or excess operating cost in dollars, plus an optional queuing penalty. The other variables are defined in the documentation section on total operating cost. Notice that all but the last term are the same as from the total operating cost formula, except that the contribution of cruise on the links is omitted.

## Flow

This output parameter indicates the number of vehicles discharged during TRANSYT simulation. It can be obtained from the sum of the OUT-pattern (output.out from Spyglass). Flow will tend to be equal to the input volume when undersaturated conditions prevail, and flows are not affected by queue spillback. Flow will tend to be less than input volume when oversaturated conditions prevail, and flows are indeed affected by queue spillback. Thus, this is an appropriate measure of effectiveness to examine when oversaturated conditions prevail. In addition, the "throughput" performance index within TRANSYT, which is analogous to Flow, is an appropriate optimization objective function to use when severely oversaturated conditions prevail.

Optimization takes into account the reduced traffic flows, unless the chosen performance index (PI) is PROS-only. Reduced traffic flows are measured within the context of step-wise simulation. All other PI's (besides PROS-only) consider delay, or throughput, or both. When maximizing throughput, the timing plan that maximizes the (reduced) traffic flows is recommended as optimal. When minimizing delay, the timing plan that maximizes the (reduced) capacity of each movement is recommended as optimal, because lower capacities produce higher delay estimates.

## Force-Off and Maximum Green

The force-off denotes the point, within a coordinated-actuated cycle length, at which the actuated

phase must be terminated. The maximum green denotes the duration of time, within a fully-actuated cycle length, at which the actuated phase must be terminated. Force-off points and maximum green settings can be either optimized or evaluated by TRANSYT. Refer to the Traffic-Actuated Control and Actuated Estimation Methodology sections of the documentation, for more details on these optimization and evaluation procedures.

### Design and Optimization

When direct CORSIM optimization is used, force-offs can be optimized directly. Otherwise, optimization of force-offs and/or maximum greens in TRANSYT is a two-step process. The first step is always to optimize splits (i.e., phase times). Following this, multiple techniques are available for determining the appropriate force-off and/or maximum green values.

Operationally, it is best to set the force-offs and maximum greens equal to the optimum splits, which usually allows actuated phases to operate in the efficient 80%-90% degree of saturation range. This can be demonstrated by the computationally intensive "actuated estimation" method, which is designed to relate force-off and/or maximum green values to their corresponding average phase times. An exception to this rule-of-thumb occurs when volumes are relatively high, and the resulting actuated phase times are 40 seconds or higher. Under these conditions, force-offs and maximum greens can be a few seconds higher than the optimum splits, while still maintaining an efficient 80%-90% degree of saturation. These conditions are not extremely common, because actuated phases often serve minor movements with relatively low volumes.

Politically, it is sometimes best for actuated phases to operate somewhat less efficiently, at a degree of saturation somewhat lower than 80%. Such phases have a tendency to always terminate via "gap-out", but never by "max-out", and the initial queue of vehicles is always served. This form of control is less efficient than it may appear in the field, and leads to higher average delays. A simple analytical technique [26] (developed by the Institute of Transportation Studies in Berkeley, California) lends itself to such politically correct timings by factoring up the force-offs and/or maximum greens for movements whose degree of saturation exceeds 85%. Since the resulting timings may then differ from the optimized timings, this should be considered a design technique instead of an optimization technique.

### Evaluation

To evaluate force-offs and/or maximum greens in TRANSYT, the actuated estimation procedure must be used. The first step is to code force-offs and/or maximum greens on the Edit > Timing screen, and turn off the "Coordinated Signal" checkbox for any fully-actuated nodes. The second step is to confirm that all of the traffic-actuated signal settings are correct on the Edit > Optional > Actuated screen. The third and final step is to select "Estimation" on the Edit > Analysis screen, and then run the model. The original user-coded green interval durations on the Timing screen will be simulated as force-offs and/or maximum greens.

## **Fuel Consumption**

The TRANSYT estimate of fuel consumption is based on a linear combination of total travel, delay, and stops. The reported fuel consumption value includes fuel consumed at cruise (for links on which a length has been provided), idle, and acceleration/deceleration. It is recommended that excess fuel consumption be used as the primary definition of the disutility index (DI).

The fuel consumption model developed for TRANSYT meets these criteria:

- The model predicts fuel consumption based on the measures of effectiveness produced by the TRANSYT simulation model
- The model accounts for the fact that TRANSYT's optimization can alter the numerical

- relationship between delay and stops
- The model can be calibrated by researchers (see the upcoming table)

The fuel model was calibrated by the University of Florida Transportation Research Center (TRC), using studies involving measurement of fuel consumption in a variety of cruise and stop-and-go cycles. The resulting data were analyzed by a step-wise multiple regression, and following model resulted:

$$F = K_{i1} \bullet TT + K_{i2} \bullet D + K_{i3} \bullet S$$

where,

F = fuel consumed in gallons (liters) per hour

TT = total travel in vehicle miles (vehicle kilometers) per hour

D = total delay in veh-hr/hr

S = total stops per hour

Kij = model coefficients that are functions of cruise speed (Vi) on each link i:

where Aj1, Aj2, Aj3 are regression coefficients and Vi is the speed on link i.

The coefficients were further adjusted to generalize the model to be representative of the distribution of sales by vehicle size in 1977. It is important to understand the other limitations of the model:

1. The model parameters were determined from studies conducted with only one test vehicle, but the model coefficients were adjusted to be representative of an "average" vehicle as defined in the TRC research report [17].
2. No explicit consideration was given to factors such as traffic congestion, vehicle type mix (i.e., trucks and diesel engines) or geometric and environmental factors such as road gradient, curvature, surface quality, temperature and other factors.
3. The model coefficients have not been adjusted for vehicle fleet mix since 1983.

These limitations require that conclusions about the absolute values of the fuel estimates be used with appropriate caution. On the other hand, the relative estimates from the existing signal timing plan to the optimized timing plan can be expected to reasonably reflect the trend expected to occur. Undocumented studies conducted by the California Department of Transportation did, however, determine that the TRANSYT fuel model was quite realistic in the mid-1980s.

The Penic fuel consumption model [32] is also available for selection, as an alternative to the default University of Florida Transportation Research Center model.

It should be noted that TRANSYT computes the fuel consumption due to random-plus-oversaturation stops differently than the fuel consumption due to uniform stops. The model assumes that random-plus-oversaturation stops are made while vehicles are traveling at the queue speed (the one at which the queue dissipates), while uniform stops are assumed to be made from cruise speed. The queue speed is calculated as follows:

$$Queue\ Speed = \frac{Saturation\ Flow}{Queue\ Density}$$

#### Model Calibration

The fuel consumption models can be calibrated for local conditions on the Edit > Optional >

Global > Model Coefficients screen (or record type 8). Details regarding the coefficients used by these models are provided in the table below. Note that "Fld 2" in the table below (for record type 8) is associated with the first field of the Edit > Optional > Global > Model Coefficients screen.

The coding of this data must properly reflect the level of numerical precision ("Prec"). For example, in order to code AFCM(1,1) as 0.075283 based on E5 precision, it would be necessary to code '7528'.

Fld	Variable	Description	Prec	Range	Default
<b>2</b>	<b>'1'</b>	<b>Fuel Consumption Model (University of Florida)</b>			
3	AFCM(1,1)	$F(\text{total}) = K_1 TT + K_2 D + K_3 S$	E5	0-(-)9999	0.075283
4	AFCM(1,2)	$K_1(\text{cruise}) = A_{11} + A_{12} V + A_{13} V^2$	E6	0-(-)9999	-0.0015892
5	AFCM(1,3)	$K_2(\text{delay}) = A_{21} + A_{22} V + A_{23} V^2$	E8	0-(-)9999	1.5066 E-5
6	AFCM(2,1)	$K_3(\text{stops}) = A_{31} + A_{32} V + A_{33} V^2$	E4	0-(-)9999	0.73239
7	AFCM(2,2)	where F = fuel consumption (gallons or liters)	E3	0-(-)9999	0
8	AFCM(2,3)	TT = total travel (veh-mi or veh-km)	E5	0-(-)9999	0
9	AFCM(3,1)	D = total delay (veh-mi or veh-km)	E2	0-(-)9999	0
10	AFCM(3,2)	S = total stops (vph)	E3	0-(-)9999	0
11	AFCM(3,3)	V = cruise speed (mph or kmph) $A_{ij} = \text{AFCM}_{ij}$ -- the model coefficients	E9	0-(-)9999	6.1411 E-6
12	BUSFAK	factor to multiply fuel consumption for buses	E3	0-99999	2.7
Fld	Variable	Description	Prec	Range	Default
<b>2</b>	<b>'11'</b>	<b>Fuel Consumption Model (Penic and Upchurch)</b>			
3	AFCM(1,1)	$F(\text{total}) = K_1 TT + K_2 D + K_3 S$	E5	0-(-)9999	0.51242
4	AFCM(1,2)	$K_1(\text{cruise}) = A_{11} \exp(A_{12} * V) / V$	E6	0-(-)9999	0.024609
5	AFCM(2,1)	$K_2(\text{delay}) = A_{21}$	E3	0-(-)9999	0.0468
6	AFCM(3,1)	$K_3(\text{stops}) = (A_{31} V^{A_{32}} + A_{33} V^{A_{34}}) / 100,000$	E4	0-(-)9999	3.8424
7	AFCM(3,2)	where F = fuel consumption (gallons or liters)	E4	0-(-)9999	1.657
8	AFCM(3,3)	TT = total travel (veh-mi or veh-km)	E4	0-(-)9999	1.681
9	AFCM(3,4)	D = total delay (veh-mi or veh-km) S = total stops (vph) V = cruise speed (mph or kmph) $A_{ij} = \text{AFCM}_{ij}$ -- the model coefficients	E4	0-(-)9999	1.48922

## HCS Import Methodology

TRANSYT-7F is capable of importing HCS-Signals (\*.XHS) data files, and HCS-Interchanges (\*.XHI) data files. HCS files to be imported cannot have multiple period (".") characters in their filenames. To create a brand-new TRANSYT-7F (\*.TIN) input file from scratch, proceed to File > Open in the GUI, select "\*.XHS" or "\*.XHI" as the file type, and browse to open the desired data file. For an existing TRANSYT-7F data file, HCS-Signals files can be imported and added to the network by selecting File > HCS > Append an HCS-Signals intersection to this network. The GUI will prompt the user to specify a node number for the new intersection.



By default, brand-new TIN files generated by HCS import will request cycle length optimization, with a cycle length range of plus-and-minus 20 seconds from the imported cycle length. For single-period analyses, new TIN files generated by HCS-Signals import will request hill-climb cycle length optimization. New TIN files generated by multi-period HCS-Signals import, or by HCS-Interchanges import, will request genetic algorithm cycle length optimization. This is because the genetic algorithm is required for either "grouped node" optimization or multi-period optimization. If an HCS-Signals intersection is appended to an existing network with cycle length optimization, the cycle length range may be expanded to accommodate the imported cycle length.

When importing HCS-Interchanges files, a progression route is automatically coded, to facilitate time-space diagrams. Node #1 is automatically coded as the master controller. The user could optionally add link curvature and bitmap backgrounds within the Map View, and could use one-touch CORSIM animation to further analyze the two intersections.

The TRANSYT-7F input data structure contains a global default free-flow speed for external links, plus free-flow speeds specific to each internal link. When HCS-Signals or HCS-Interchanges files are imported to generate a brand-new TIN file, free-flow speeds for all links are automatically set equal to the stored value under Options > Cruise Speed. Subsequently, link-specific free-flow speeds can optionally be customized within TRANSYT-7F, on the Edit > Feeders screen.

When HCS-Signals files are imported, they are by default designated as "coordinated signals", to facilitate subsequent optimization. Since coordinated signals are either pre-timed or semi-actuated, the user should consider adjusting the data for any fully-actuated signals to be imported. Semi-actuated signals typically have one phase designated as non-actuated, and the reference interval is usually referenced to the end of green for the major street thru movement. If the user wishes to simulate existing conditions in which certain intersections are uncoordinated, it would be necessary to open the Edit > Timing screen, and uncheck the Coordinated Signal checkbox for certain intersections.

When importing non-integer interval times in the HCS file such as 27.4 seconds or 6.7 seconds, TRANSYT-7F will round off these numbers to the nearest integer. Any interval round-off errors can be corrected within the Edit > Timing screen. On the other hand, if it were necessary to evaluate an exact green time of 27.4 seconds using the TRANSYT model, the user could code a green time of 25.0 seconds in the Edit > Timing screen, along with an extension-of-effective-green of 2.4 seconds in the Edit > Traffic screen.

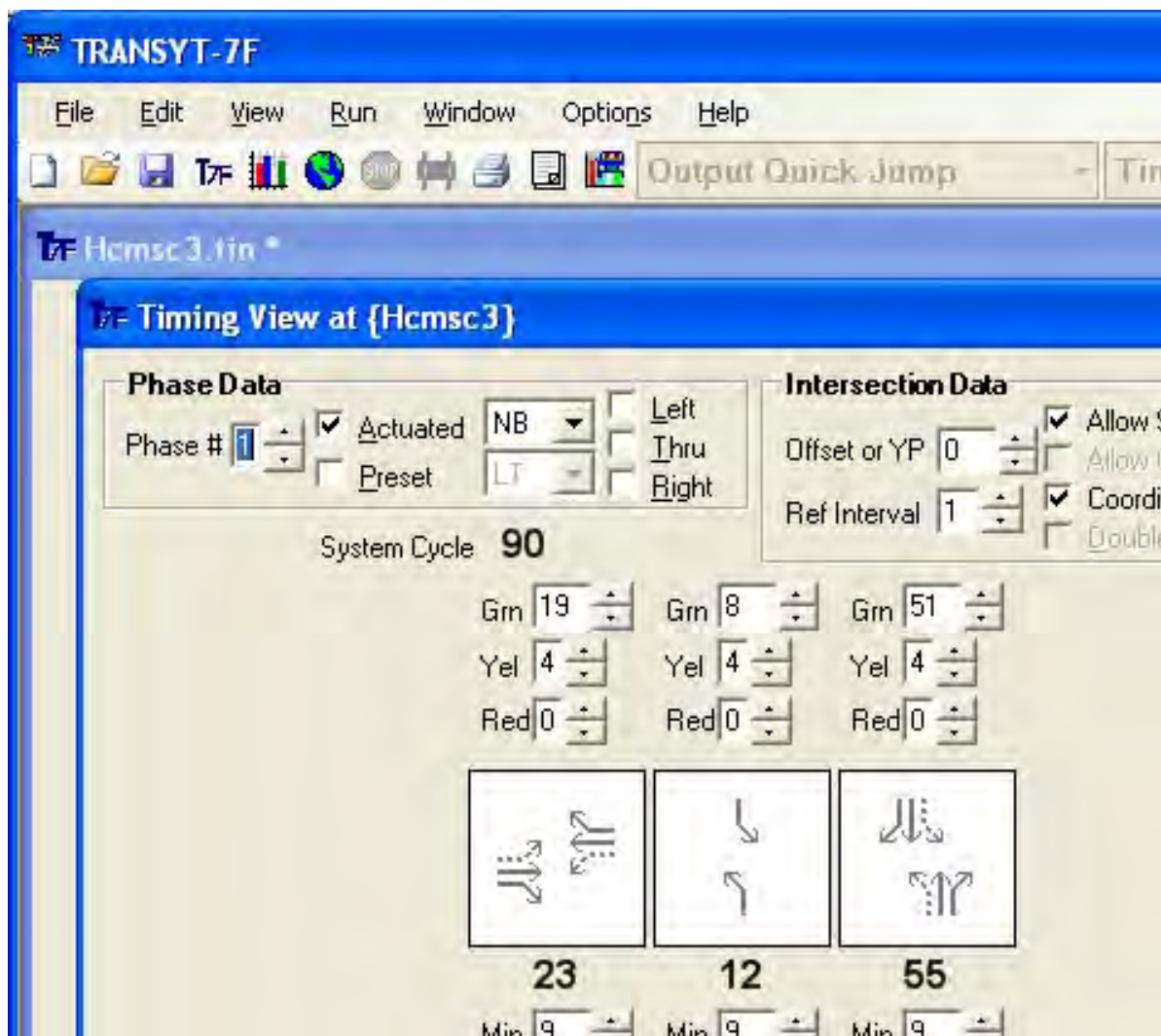
Engineers will sometimes compare results obtained from HCS, CORSIM, and TRANSYT. In a progressive system of pre-timed and semi-actuated signals, it is important that all three programs are consistent in how their timing plans are defined. This requires consistency among phasing sequences, and reference intervals.

Regarding phasing sequences, east-west phases always precede north-south phases in HCS. Phase #1 is always the first east-west phase in the timing plan. Therefore, when HCS files are imported into TRANSYT-7F, east-west phases always precede north-south phases within TRANSYT-7F. When using one-touch CORSIM animation for pre-timed signals from TRANSYT-7F, east-west phases always precede north-south phases within CORSIM. When using one-touch CORSIM animation for semi-actuated signals from TRANSYT-7F, minor street phases always precede major street phases within CORSIM.

Regarding reference intervals, TRANSYT-7F automatically sets these equal to interval #1 for imported HCS files. CORSIM always assumes that offsets are referenced to interval #1 at pre-timed signals. HCS-Interchanges assumes that offsets are referenced to interval #1 at both signals, regardless of actuation. Therefore pre-timed timing plans can usually be imported from HCS into TRANSYT-7F, and exported from TRANSYT-7F to CORSIM, without requiring any user changes to the reference intervals. An exception to this is when pre-timed timing plans are exported to CORSIM with dual-ring phasing (due to 13 or more intervals in the timing plan, or user-requested dual-ring export on the Options > Preferences screen).

For semi-actuated signals, offsets are typically referenced to the yellow interval (end of green) for the major street thru movement. This provides consistency both with CORSIM's assumptions, and with typical semi-actuated controllers. After importing HCS data, the Edit > Timing screen should be used to re-code offsets and reference intervals, if further analysis of actuated control is needed. In summary, the phasing sequences and reference intervals must reflect actuation, and which programs are being used.

The screen image below illustrates a few of these HCS import issues, such as the rounding-off of non-integer green times, the automatic coding of east-west phases prior to north-south phases, and reference interval #1. The HCS file that was imported had a green time of 50.7 seconds for the north-south thru movement phase, but this was rounded up to 51 seconds in TRANSYT-7F.



Report header information such as the analyst, agency, date of analysis, analysis time period, jurisdiction, and project title are imported from HCS when a brand-new TIN file gets created. Report header information is not imported when appending an HCS-Signals file onto an existing TIN file.

## Interference

Interference is a measure of the level that red intervals (steps) at an intersection on a defined artery interfere with progression on the artery. One "interference" is counted if leaving an intersection on a given green step results in arriving on red downstream. The value reported by TRANSYT for a given direction is actually the ratio of the summation of all interferences in that direction divided by the maximum possible interference. The maximum possible interference occurs if leaving on any green step at every intersection results in arrival on red downstream. Interference is only provided within the TRANSYT legacy report when one or more progression routes have been defined in the input file. Progression routes are either coded graphically (using the Map View), or numerically (using the Edit > Optional > Route screen).

## Macroscopic Simulation Methodology

The unopposed condition of simulation is explained here. The opposed traffic moving condition is explained in the Permitted Movement Methodology section. There are three traffic flow patterns: the "IN", "GO", and "OUT" patterns. Before discussing these in detail, it is necessary to explain the flow balancing done by TRANSYT.

### Traffic Flow Balancing

Each downstream link may be fed by up to four upstream links. Typically there will be at most three, but the program allows four for special cases. Since data are not collected at all intersections simultaneously, data collection by its nature is not precise, and since traffic leaves and enters links mid-block, it is not reasonable to expect upstream traffic volumes to balance exactly with those downstream. TRANSYT allows deviations in these counts, although gross differences are identified via warning and error messages.

It is important to understand what TRANSYT does about these differences. Consider the following numerical example, where the numbers shown are coded data:

Upstream left feeding volume: 200 vph  
Upstream through feeding volume: 500 vph  
Upstream right feeding volume: 100 vph  
Mid-block feeding volume: 50 vph  
Downstream volume: 1050 vph

Clearly the sum of the upstream inputs ( $200 + 500 + 100 + 50 = 850$ ) is not the same as the coded downstream value (1050). TRANSYT will adjust the flows as follows:

1. The downstream flow (1050 vph) and the mid-block flow (50 vph) are considered to be accurate. The mid-block flow is deducted from the total flow, leaving 1000 vph.
2. The upstream flows are adjusted proportionally to equal the remaining downstream total flow:

$200 * (1000 / 800) = 250$   
 $500 * (1000 / 800) = 625$   
 $100 * (1000 / 800) = 125$

which sum to 1000 vph.

These adjusted volumes are the actual volumes used internally by the program. In some cases this results in input flows temporarily exceeding the coded saturation flow rate.

The program checks the magnitude of the adjustment (either upward or downward) and if the

adjustment is greater than certain thresholds, either a warning or error message is issued. Very low volumes are not checked, because of the minimum requirement of 10 vph for any coded traffic volume.

Some input data pre-processors compute the link-to-link flows, and this balancing, automatically. The TRANSYT-7F graphical user interface (GUI) now features automatic flow balancing. In the Map View, when link connections are drawn graphically, link-to-link flows are automatically computed and coded in the background. On the Edit > Feeders screen, the user must simply type in the upstream feeding node number, and the GUI fills in the remaining data. For complex traffic flow patterns, these preprocessed feeding flows may need to be revised by the user. Note that feeding link flows may vary between time periods when performing multi-period simulation.

### The Arrival Flow Pattern (IN-Pattern)

The arrival flow pattern is expressed mathematically as follows:

$$IN_{it} = \sum_j^n F_{ij} (P_{ij} \bullet OUT_{jt'})$$

where,

IN<sub>it</sub> = the IN-pattern on link i for time step t

F<sub>ij</sub> = smoothing process related to platoon dispersion for flow to link i from link j

P<sub>ij</sub> = the proportion of the feeding link OUT-pattern that feeds the subject link

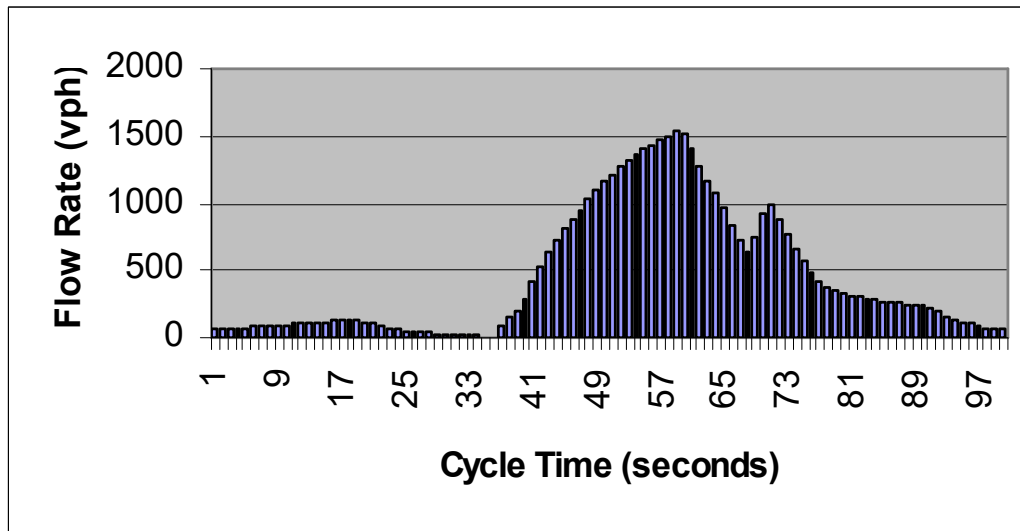
OUT<sub>jt'</sub> = the OUT-pattern of link j for step t'

t' = the time step t minus the travel time for flow to link i from link j

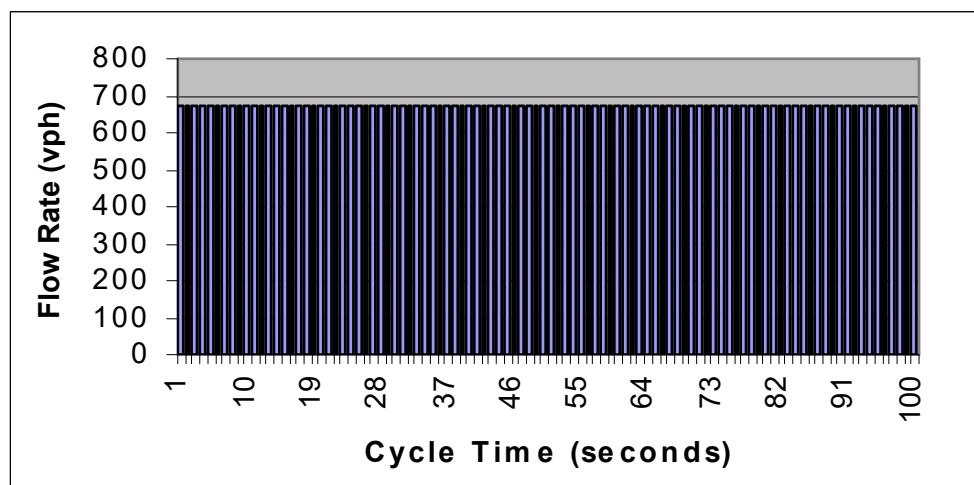
n = the number of links (j) that feed link i

The IN-pattern is computed for each step, t, in the cycle, thus forming a pattern, or flow profile. Flow profiles can be viewed by requesting them in the output file, or by using the Profile View and Spyglass features. A Spyglass output file called "inpat.out" contains arrival flow pattern information. In link-wise simulation, the flow arrival is assumed to occur at the stop line. In step-wise simulation, the arrival is assumed to occur at the dynamic back of queue location. Note that the IN-pattern for external (network input) links or mid-block input flows is always a uniform distribution. That is, random arrivals tend to "even out" over a period of time. For internal links, the IN-pattern is affected by platoon dispersion.

The figure below illustrates a sample flow profile on an internal link having 450 vph, at a signal whose cycle length is 100 seconds:



The figure below illustrates a sample flow profile on an external link having 675 vph, at a signal whose cycle length is 100 seconds:

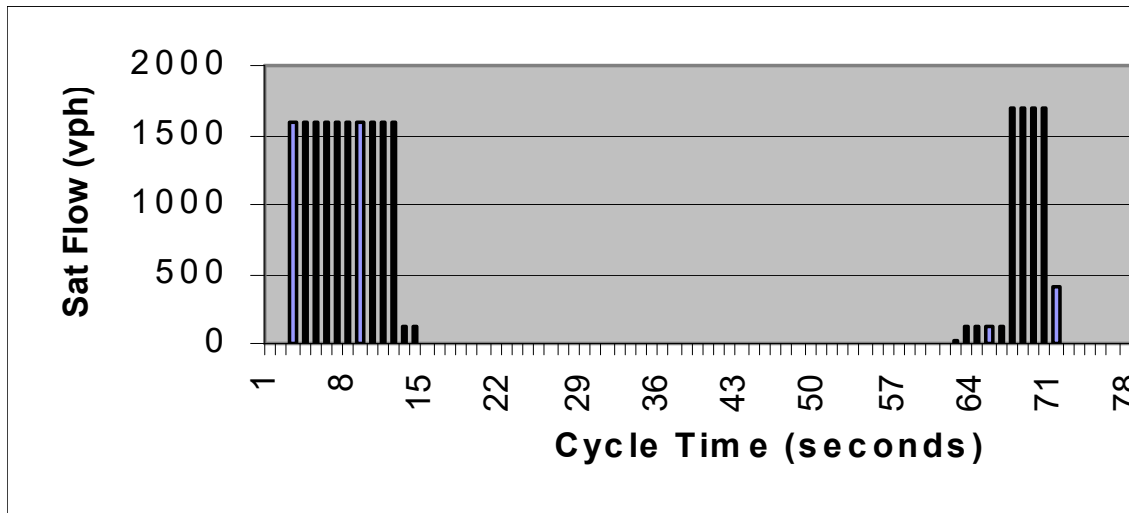


#### The Saturation Flow Pattern (GO-Pattern)

The GO-pattern is the flow rate at each step that would leave the stop line if there were enough traffic to saturate the green. This saturation flow rate is a user input. In step-wise simulation, queue spillback due to saturated downstream links and/or adjacent turn bays is considered by reducing the GO-pattern of the affected links. Links that are full during a given step receive an IN-pattern of zero in this case. For permitted, opposed movements, the GO-pattern is computed according to the permitted movement models.

The GO-pattern can also be plotted by using the Profile View and Spyglass features. A Spyglass output file called "gopat.out" contains this information. Once the GO-pattern is plotted, capacity of the given movement can be computed by taking the area under this saturation flow rate profile. The figure below illustrates a sample saturation flow rate profile on a link operating under protected-permitted

phasing:



#### The Departure Flow Pattern (OUT-Pattern)

The OUT-pattern is the profile of traffic actually leaving the stop line. It is usually equal to the GO-pattern as long as there is a queue. After the queue dissipates, it is equal to the IN-pattern for the remainder of the effective green. Permitted, opposed movements are simulated somewhat differently, as described in the Permitted Movement Methodology section.

The OUT-pattern can also be plotted by using the Profile View and Spyglass features. A Spyglass output file called "outpat.out" contains the departure flow pattern information. Once the OUT-pattern is plotted, the number of vehicles discharged (a.k.a. the "Flow" parameter from TRANSYT output) can be computed by taking the area under this departure rate profile. When conditions are undersaturated, queue blockage does not occur, and the number of vehicles discharged is typically equal to the user-specified input demand volume. When conditions are oversaturated, queue blockage may occur, and the number of vehicles discharged may be lower than the input volume.

To determine the OUT-pattern, the queue (or the number of vehicles held at the stop line during any time interval,  $t$ ) must first be determined by:

$$m_t = \max\{(m_{t-1} + q_t - s_t), 0\}$$

where,

$m_t$  = number of vehicles in the queue in time interval  $t$  on a given link  
(and similarly for  $m_{t-1}$ )

$q_t$  = number of vehicles arriving in interval  $t$ , given by the IN-pattern

$s_t$  = number of vehicles allowed to leave in interval  $t$ , given by the GO-pattern

When single-cycle simulation is requested, the queue length calculated represents only what would occur due to traffic arriving during that cycle. In order to simulate queues building over time, it would be necessary to request multi-cycle step-wise simulation.

The OUT-pattern is given for link  $i$  during time interval  $t$  by the expression:

$$OUT_{it} = M_{i,t-1} + q_{it} - M_{it}$$

### Pattern Initialization Cycle

One difficulty with implementing step-wise simulation is that, when starting the simulation, values of the OUT-pattern leaving upstream links in previous steps to arrive downstream at the simulated step are not available. In link-wise simulation, this is not a problem since each link is simulated for all steps in an upstream-to-downstream sequence; therefore, upstream link OUT-patterns are available when simulating the downstream links.

In the case of a 100% step-wise simulation approach, this would not be the case. To deal with this problem, a pattern initialization cycle in which the system is simulated on a link-by-link basis has to be performed. During the cycle, the OUT-pattern is initialized. No measures of effectiveness are calculated during this cycle.

### Simulation Initialization Period

The queuing model used in the pattern initialization cycle described above is the traditional ("vertical stacked") TRANSYT queuing model because the simulation is performed on a link-wise basis. To initialize the queues, another initialization period is simulated using step-wise simulation. The new queuing model and queue spillback logic are used in this period. No measures of effectiveness are calculated during this period.

## **Maximum Back of Queue**

This is the average or expected maximum extension of the queue upstream on the link during the cycle. It is not the number of vehicles queued at the beginning of green. The value is considered to be an average value because TRANSYT is a deterministic simulation process where approximately half of the real-world queues are expected to be higher than the average value, and approximately half of the queues are expected to be lower than the average value. The maximum back of queue includes vehicles that arrive during the green and join the back of queue, while the front of the queue is discharging during the initial seconds of effective green.

Although this value represents the average back of queue, it also represents a maximum back of queue. It represents the largest queue that occurred during simulation. When single-cycle simulation is requested, the largest queue that occurred during simulation is naturally the only queue that occurred during simulation. When multi-cycle simulation is requested, the maximum back of queue will potentially increase during simulation, if queue blockage or oversaturation is occurring.

Note that the uniform maximum back of queue value reported in the TRANSYT output file (\*.TOF) is not associated with the uniform number of queued vehicles that can be used to compute uniform delay (d1). In the first few seconds of effective green time for any movement, the maximum back of queue will typically continue to increase as the uniform number of queued vehicles decreases. The overall maximum back of queue for the entire analysis period is also reported in the legacy output file, and incorporates the random queue component.

In TRANSYT, when there are shared stopline links, the overall queue length for the lane group is reported for the primary link only. The program estimates an overall queue length for the shared lane group. Refer to the Edit > Optional > Global > Shared Lanes screen documentation for more details regarding primary and secondary shared stopline links.

### Link-Wise Simulation

When link-wise simulation is used, the maximum back of queue value can assist in determining the possibility of queue spillback into the upstream intersection. This is done by comparing the maximum back of queue value with the queuing capacity value. Queuing capacity values can optionally be coded for turn bays, and for links where it is critical that the queue does not extend beyond a certain point.

The traffic engineer is often concerned with whether or not the queue backs up far enough to "spillback" into the upstream intersection. Earlier releases of TRANSYT would accumulate vehicles to the back of queue until the queue clears. At that point, the back of queue resets to zero. This model does not adequately consider the dynamics of queue movement. As a result, it generally overestimates the extension of the back of queue. However, this model must still be used by link-wise simulation.

### Step-Wise Simulation

In step-wise simulation, the back of queue model considers the dynamics of queue movement more realistically. In this model, the queue forms as a shock wave (representing the back of queue) that propagates upstream at a rate defined by the arrivals to the back of queue. Vehicles in the queue are assumed to remain stationary until a backward shock wave reaches them, when they start moving at a rate dependent on the saturation flow rate. A detailed back of queue algorithm was developed to take into consideration the stop-and-go operation associated with permitted turns and congested conditions. Refer to the Back of Queue Algorithm section of the documentation for more details.

Also under step-wise simulation, if the uniform maximum back of queue exceeds the queuing capacity at any time, this triggers the simulation of queue spillback and queue spillover. Queuing capacity values can optionally be coded for turning bays, and for links where it is critical that the queue does not extend beyond a certain point.

Beginning with release 8, random-plus-oversaturation queue is added to the back of queue. The average random-plus-oversaturation queue in number of vehicles,  $N_o$ , is calculated according to an ARRB model [16,33] as follows:

$$N_o = 0.25cTX^2 \left\{ (X-1) + \sqrt{(X-1)^2 + \frac{mX}{cT}} \right\}$$

where all variables are defined in the documentation for Delay and Level of Service. Note that in single-cycle simulation, a deterministic component,  $N_d$ , is included in  $N_o$  and defined as follows:

$$N_d = \frac{(X-1)cT}{2}$$

This deterministic component is explicitly considered in multi-cycle step-wise simulation. Thus, in multi-cycle step-wise simulation,  $N_d$  is subtracted from  $N_o$ .

### 95th Percentile Queue

According to conventional wisdom, turn bay lengths and signal spacing should be long enough to accommodate the 95th percentile queue length. When this condition is satisfied, queue spillback and queue spillover are only expected to occur 5% of the time. Because TRANSYT reports the "expected" or "average" maximum back of queue, a conversion is required to obtain the 95th percentile value. This conversion is now taken from Highway Capacity Manual [12] Chapter 16, Appendix G, as follows:

$$Q_{95\%} = Q \cdot f_{B95\%}$$



where,  $Q_{95\%}$  = 95<sup>th</sup> percentile queue length (veh)  
 $Q$  = average queue length (veh)  
 $f_{B95\%}$  = 95<sup>th</sup> percentile back-of-queue factor

For pre-timed movements, the 95th percentile back-of-queue factor is calculated as follows:

$$f_{B95\%} = 1.6 + e^{\frac{-Q}{5}}$$

For actuated movements, the 95th percentile back-of-queue factor is calculated as follows:

$$f_{B95\%} = 1.5 + 0.6e^{\frac{-Q}{18}}$$

The 95th percentile queue length is available within the node output summary reports. Thus, it is only given for the 12 cardinal movements within the link numbering scheme. If it becomes necessary to obtain 95th percentile queue lengths for link numbers 13 and higher, the formulas above should be applied by hand.

#### Jam Spatial Headway and Optimal Spatial Headway

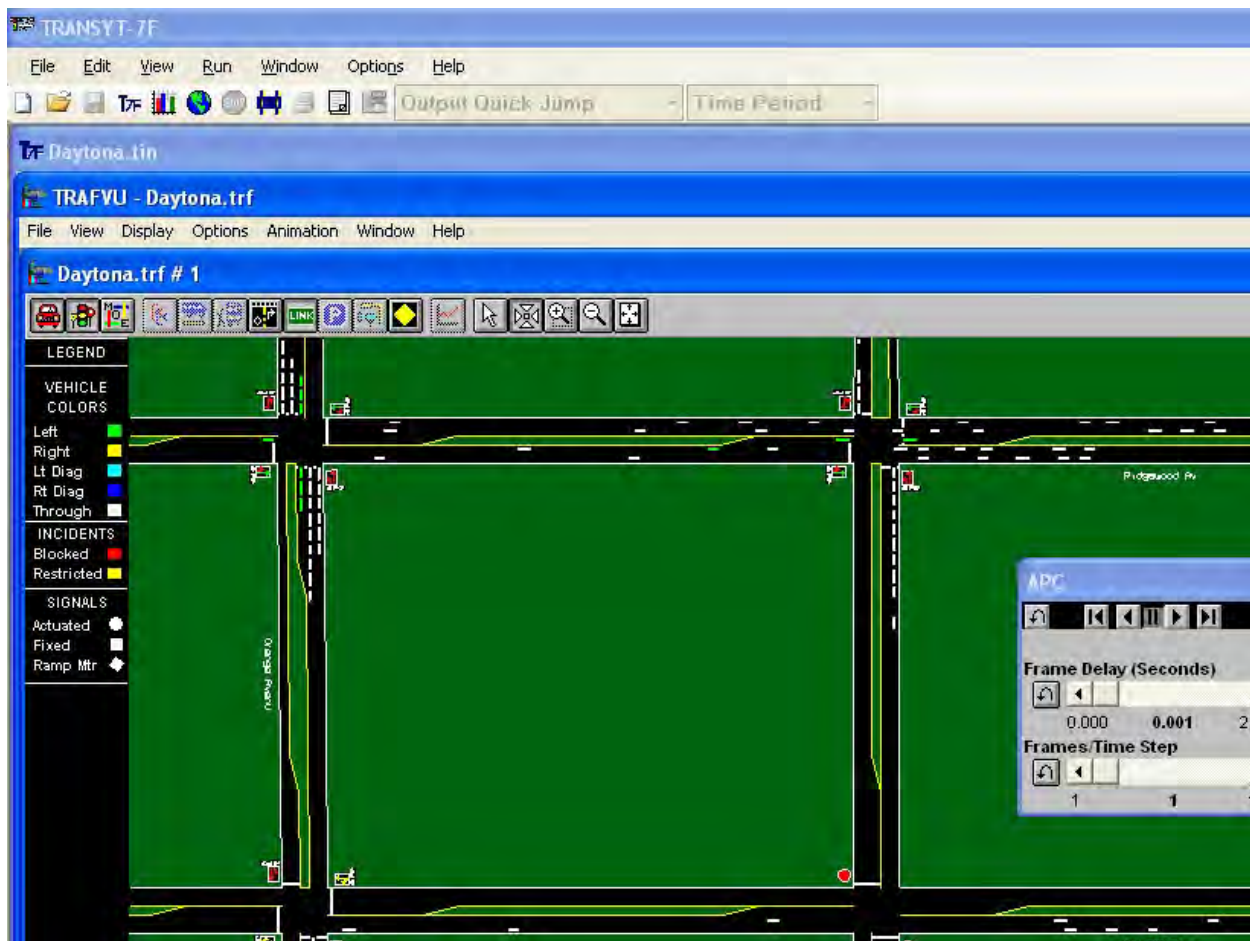
For both link-wise and step-wise simulation, the internal calculation of queuing capacity is affected by the jam spatial headway and link length. However, when using step-wise simulation, both the jam and optimal spatial headway are used in the simulation and calculation of the back of queue location. Refer to the Global Screen and Queuing Screen documentation for instructions on coding jam and optimal spatial headways, if they are unequal to the default values. Model results may be quite sensitive to the spatial headway values in networks where queue spillback or queue spillover is present.

## One-Touch CORSIM Animation

This feature requires TSIS-CORSIM to be currently installed (and run independently at least once) on the computer. TSIS-CORSIM is sold separately.

Clicking on the "CORSIM Animation" icon, or selecting "File > CORSIM > View animation for this network", will display CORSIM Animation (also known as TRAFVU) for the currently loaded TRANSYT data file. TRANSYT-7F will automatically generate a CORSIM input (\*.TRF) file, process the TRF file through CORSIM, and then launch TRAFVU to provide animation of the traffic network. This overall process may take a few seconds. Once TRAFVU is loaded, the user will need to:

1. Click on the red "Vehicle" icon to display vehicles
2. Zoom in on a certain part of the link-node diagram
3. Adjust the "APC settings" to slow down animation
4. Press "Play" to begin animation



CORSIM and TRANSYT contain different modeling methodologies, so the visible results (e.g., queue lengths) from CORSIM animation may not always be identical to those from TRANSYT. Any metric unit data will be automatically converted into English units for use in CORSIM.

TRAFVU graphics are very informative for learning about actuated control. By turning on the Sensors map adornment, it is possible to show typical detector layouts for semi-actuated or fully-actuated control. Clicking on an actuated signal will reveal the dual-ring phasing sequence. For semi-actuated signals, pie charts are available for graphic display of permissive periods and force-offs.

The following TRANSYT inputs are exported to CORSIM and displayed by TRAFVU: Peak adjusted traffic volumes, signal timing, lane configuration, adjusted saturation flow rates, percent heavy vehicles, free-flow speeds, link lengths, link curvature, bitmap backgrounds, duration of analysis, turn bay storage, East/West and North/South street names.

The following TRANSYT inputs are not exported to CORSIM: Average queue spacing, start-up lost times, extension of effective green times.

Link-specific mean discharge headways in CORSIM do not reflect any heavy vehicle or turn movement saturation flow rate adjustments from TRANSYT, because this would constitute "double-counting" their effect on results. Start-up lost times for CORSIM are set at 2.0 seconds for internal links, and 4.0 seconds for external links. Informal research has shown that setting the start-up lost times in this manner allows control delays from CORSIM to match control delays from HCS more closely.

The following data ranges are enforced:

- Turn pocket or turn bay length (20-9999 feet)
- Link length (50-9999 feet) (If no storage length is coded, the link is set to 2000 feet)
- Saturation flow rate (1000-4000 vehicles per lane per hour)
- Number of turn pockets (maximum of 3 left-turn and 3 right-turn pockets)
- Volume (maximum of 9999 vehicles per hour)
- Green plus yellow time must be at least 8 seconds for non-overlap phases
- Green time must be at least 5 seconds for non-overlap phases
- Pre-timed green interval duration (maximum of 120 seconds)

The following assumptions and limitations also exist: Negative node coordinates are allowed by TRANSYT, but are not allowed by CORSIM. If no queue storage lengths are coded in TRANSYT, link lengths are automatically set to 2000 feet. When a left-turn in TRANSYT is opposed by a right-turn, but is not opposed by a thru movement, assumptions are made in CORSIM regarding left-turn protection and signal phasing. Exclusive pedestrian phases from TRANSYT sometimes cannot be transferred automatically to CORSIM. If an intersection in TRANSYT has more than four approaches, only four approaches will be transferred automatically to CORSIM. For actuated coordination, if dual lagging left-turns are present on the major street, it is assumed that any protected left-turns on the minor street will also lag their associated thru movements. If a protected right-turn green arrow moves in the same phase as a conflicting thru movement in TRANSYT, CORSIM may not illustrate this. Green time in TRANSYT should not be given to any turning movements whose "adjusted" volumes are equal to zero. If six or more thru lanes are coded in TRANSYT, only five thru lanes will be transferred to CORSIM. If the TRANSYT data file contains a mixture of both actuated and non-actuated phases, semi-actuated control will be simulated in CORSIM if the number of non-actuated phases is equal to one; otherwise, fully-actuated control is assumed. If the TRANSYT data file contains only pre-timed phases, but the timing plan contains 13 intervals or more, isolated actuated control is used in CORSIM (with recall to maximum green, to emulate pre-timed operation).

Most of the items above are not limitations of CORSIM, but simply limitations of the automatic conversion process between TRANSYT-7F and CORSIM. If any of these assumptions or limitations do not allow the model to match conditions in the field, the user can select File > CORSIM > Generate TRF file for this network, and customize the CORSIM input data within TSIS. Here are the steps for editing the CORSIM network within TSIS:

1. Launch TSIS.
2. Select File > New > Project File, and type in the desired project name.
3. Load the TRANSYT-7F data file into TRANSYT-7F.
4. From TRANSYT-7F, select File > CORSIM > Generate TRF file for this network.
5. Browse to the folder that matches the TSIS project name, type in the desired file name, and click on Save.
6. Inside TSIS, locate the new project name on the left-hand side of the screen, and expand this project by clicking on the "plus sign".
7. Right-click on the TRF file and select Open With > Translator. This will create a TNO file with the same file name as the TRF file.
8. Edit the TSIS-CORSIM network by right-clicking on the TNO file, and selecting Open With > TRAFED.

If an error message says "This feature requires TSIS-CORSIM" even after TSIS has been installed, it may be necessary to first run TSIS in the standalone mode, so that TSIS-CORSIM can be properly registered on the computer. If the user does not have administrator rights, they may encounter an error message that says "ERROR: Failed to find file". This error message should not hinder animation, so the user should simply click "OK" and then view animation normally.

Link curvature data is purely cosmetic, and does not affect TRANSYT or CORSIM model results. Instead, model results are sensitive to user-coded link lengths on the Lanes or Traffic screens. When

performing one-touch CORSIM animation for the entire network, TRANSYT-7F will attempt to export link curvature and bitmap background information to CORSIM. TRANSYT and CORSIM use different types of input data to specify link curvature and bitmap backgrounds. If link curvature or bitmap scaling appears incorrect when using one-touch animation, the user can remove this information from the TRANSYT-7F Map View before transferring data to CORSIM. In the Options > Preferences screen, link curvature export to CORSIM can either be deactivated, or approximated by exporting mid-block dummy nodes.

In CORSIM, offsets for pre-timed signals are always referenced to the start of interval #1 of phase #1, and interval #1 is typically defined as a green interval. Offsets (yield points) for coordinated actuated signals are always referenced to the end of the major street thru movement green interval. In TRANSYT-7F, offsets and yield points can be referenced to any signal interval, because the user can code a "reference interval number" at each intersection. Offset and yield point values from TRANSYT-7F are exported directly to CORSIM without any numeric conversion, regardless of what reference interval numbers were coded in the TRANSYT-7F input (\*.TIN) file. If the reference intervals coded in TRANSYT-7F are consistent with CORSIM's assumed reference intervals as described above, this allows TRANSYT model results and time-space diagrams to be consistent with the visible results from TRAFVU animation.

In the Options > Preferences screen, pre-timed export to CORSIM can be handled using dual-ring timing with max recall. This causes an isolated actuated controller to mimic a pre-timed controller by always displaying its maximum green times, but coordination settings from the pre-timed controller (i.e., offsets and reference intervals) cannot be preserved. There are several advantages and disadvantages to exporting a pre-timed signal with dual-ring timing in CORSIM. The single-ring export process is limited to a maximum of 12 intervals in the timing plan, and green interval times of 120 seconds or less. Single-ring controllers are illustrated as square signal heads in TRAFVU, which provides a visual contrast to the circular signal heads used to illustrate dual-ring controllers. CORSIM assumes different offset reference intervals for single-ring and dual-ring controllers, as mentioned earlier. Certain unusual phasing sequences can only be handled by single-ring controllers.

In the Options > Preferences screen, right-turns on red (RTOR) export to CORSIM can be requested. If this option is not requested, RTOR will only be allowed (in CORSIM) if right-turns move in the same phase as a conflicting thru movement (in TRANSYT). For left-hand drive, LTOR export to CORSIM can be requested.

Also in the Options > Preferences screen, the user can enter a percentage of heavy vehicles to be exported to CORSIM. This percentage of heavy vehicles will be emitted at every entry node, and the corresponding number of longer vehicles will become visible within TRAFVU. In addition, mean queue discharge headways that are mapped to CORSIM will be automatically adjusted (decreased), to avoid double-counting the effect of heavy vehicles on saturation flow rates. Heavy vehicle satflow adjustments will be identical to those performed in the Satflow View and Saturation Flow Rate Calculator screens.

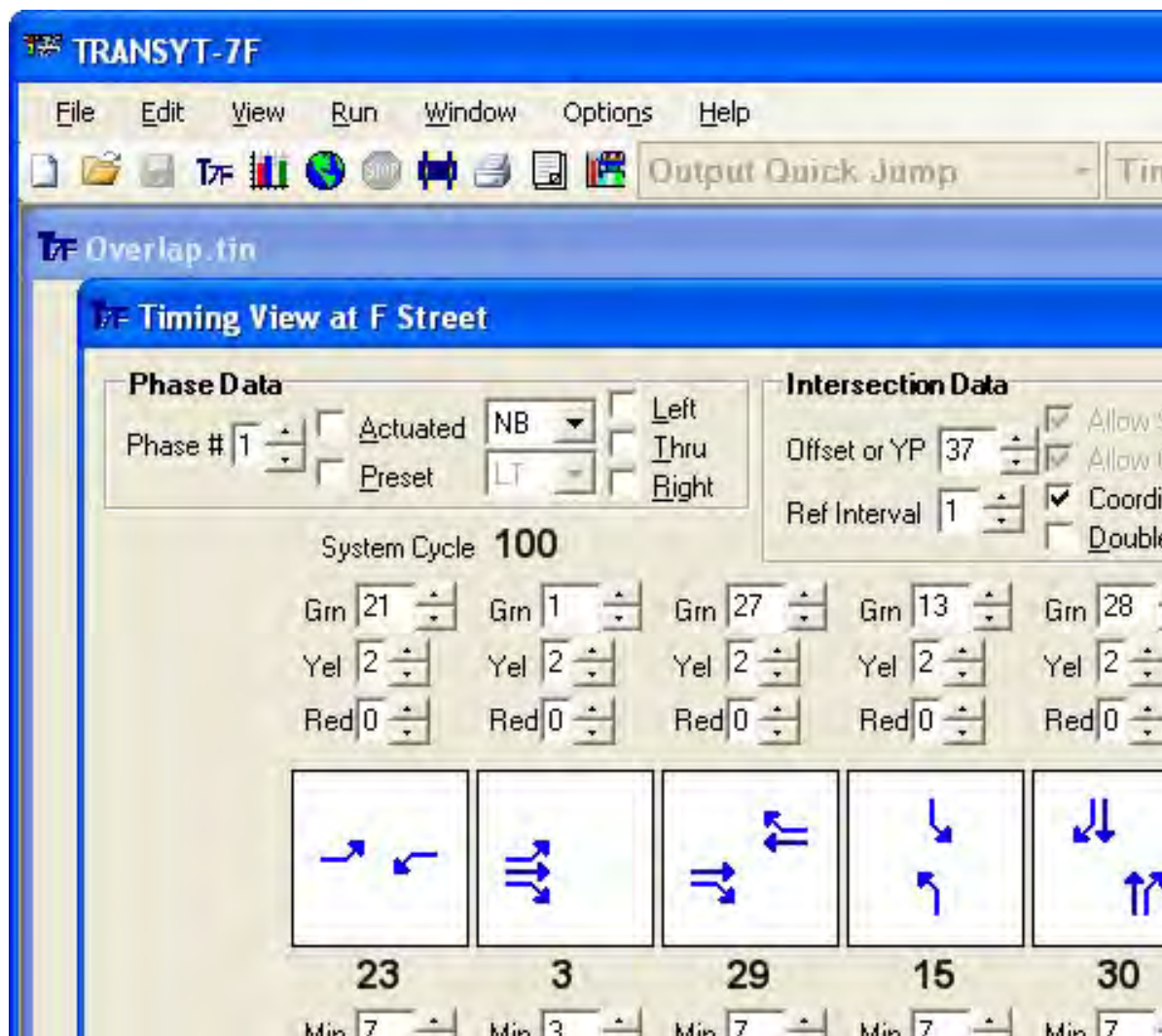
## Operating Cost

The estimate of total system operating cost includes all costs associated with vehicle operation in the system. It includes no estimate of system operation itself, such as control and surveillance equipment, communications, or operating personnel. Thus, this is an estimate of the road user costs, including vehicle operation, fuel consumption, and passengers time. See also the Total Operating Cost section of the documentation, for more information.

## Overlap Phasing

The screen image below illustrates overlapping left-turn movements, in the East/West direction, that do not terminate simultaneously. Single ring phase #2 is called the "overlap phase". TRANSYT can

simulate overlaps as short as one second, provided that the start-up lost times (SLT) and extension of effective green times (EEG) are coded correctly. Sample input data file "Overlap.tin" from the TRANSYT-7F installation illustrates this coding technique. For an overlap phase, it is acceptable to code a yellow time of 1 second, even if the duration of yellow displayed by the controller is longer than 1 second. Double checking the effective green times for each link (in the wide, 132-column format output file) is recommended. Effective green time ( $g$ ) is affected by actual green time ( $G$ ), SLT, EEG, and clearance time ( $Y+AR$ ). It is allowable for EEG to exceed  $Y+AR$ , when appropriate. Clearance time is always assumed to be used as green time by overlapping, continuing movements. Thus, effective green time can be computed as  $g = G - SLT + EEG$  for non-overlap movements, or  $g = G + Y + AR + G - SLT + EEG$  for overlap movements.

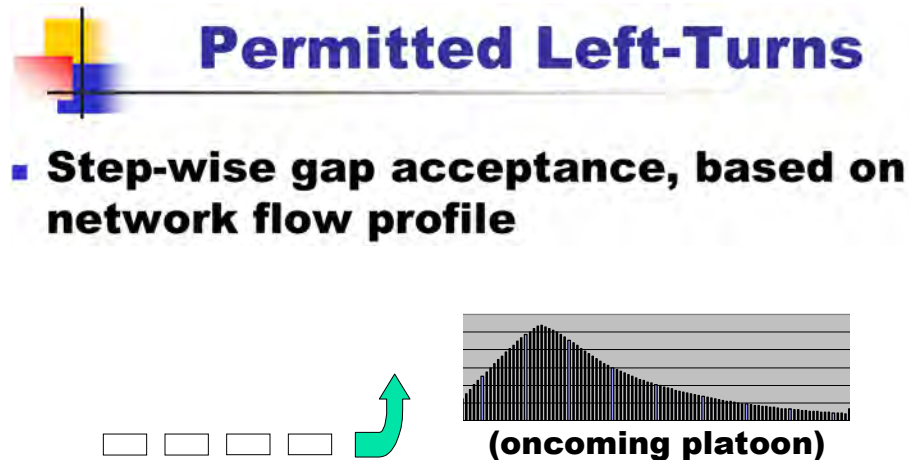


By default, SLT and EEG are both 2 seconds. However, for the westbound left-turn above, EEG was coded as 4 seconds. For the eastbound thru and right-turn, SLT was coded as 4 seconds, to prevent them from moving during the westbound left-turn. Therefore, effective green time for the westbound left-turn is  $21-2+4 = 23$  seconds. Effective green time for the eastbound left-turn is  $21+2+1-2+2 = 24$  seconds.

## Permitted Movement Methodology

There are three distinct periods of time for which TRANSYT simulates a permitted movement:

- 1) Opposing (or conflicting) queue service time: Although permitted moving vehicles search for gaps in the opposing traffic, they are actually not eligible to move until the opposing (or conflicting) queue has been served. This is consistent with conventional wisdom, which stipulates that permitted moving vehicles will not travel through an opposing queue discharging at rates such as one vehicle per two seconds.
- 2) Standard permitted movement model execution: However, once the opposing queue has cleared, additional green time will be available if the phase is not oversaturated. The additional green time is used to serve platoons in search of progression, or simply late arriving vehicles ("stragglers"). This is the ideal time for permitted movements to be discharged, and this is when the TRANSYT permitted movement models are exercised. The TRANSYT permitted movement models are quite realistic, because they are re-applied during each step of simulation, as a function of the opposing link's flow profile. This operation is illustrated in the figure below.



Permitted Movements and Shared Movements

- 3) Sneaker phase: Finally, when the effective green time has expired, this is when the "sneakers" are allowed to move through the middle of the intersection. For example, if 2.0 sneakers per cycle are coded by the user, low-volume permitted movements can actually perform well when the opposing movements are oversaturated, because these vehicles will be served regardless of the opposing traffic volume intensity. However, if there is no standing queue for the permitted movement at the time of the sneaker phase, it is possible that coding sneakers may have no effect on results. The Profile View screen can be used to determine if and when sneakers are moving. Note that the default value of sneakers within the software is zero, and that sneakers are typically not coded for right-turns (assuming right-hand drive).

Only non-permitted, unopposed links may be specified as opposing links. The software only exercises the permitted movement model on opposed links versus unopposed links. In the event that two opposing movements sequentially yield to one another during different parts of the cycle, the maximum flow rate (MFR) and/or phasing sequence may require calibration, to approximate this condition.

The default standard permitted movement model for signalized intersections in TRANSYT is one developed by the Australian Road Research Board (ARRB). The user may override this model and use any of the models described hereafter. Most TRANSYT permitted movement models can be calibrated for local conditions on the Edit > Optional > Global > Model Coefficients screen (or record type 8). Details regarding the coefficients used by this model are provided in an upcoming table, in the "Model Calibration" section.

All of the TRANSYT permitted movement models are subject to the following:

$$MFR(t) = 0 \text{ if the opposing flow is at its saturation flow rate at time } t$$

$$MFR(t) = MFR_u(t) \text{ if the opposing flow has cleared at time } t$$

where,

$MFR(t)$  = maximum flow rate at time  $t$ , and

$MFR_u(t)$  = MFR without an opposing queue at time  $t$

#### ARRB Model

This is used as the permitted-movement model at signalized intersections, unless the user specifically chooses another model. The ARRB model [27] was developed analytically, but has shown good agreement with actual traffic, and is consistent with other methods. The form of the model is as follows:

$$MFR(t) = \frac{L \cdot S \cdot \exp[-(T - D)L]}{1 - \exp(-H \cdot L)} \cdot 3600$$

where,

$$L = \frac{v_o(t)}{1 - \frac{D \cdot v_o(t)}{N}}$$

$$S = \left(1 - \frac{D \cdot v_o(t)}{N}\right)^N$$

and

$v_o(t)$  = the opposing flow rate at time  $t$

$H$  = minimum departure headway, in seconds

$D$  = minimum headway of opposing traffic lanes, in seconds

$T$  = acceptable critical gap, in seconds

$N$  = number of opposing lanes

$\exp$  = natural base of logarithms

For the ARRB model, the software estimates the number of opposing lanes for permitted movements. Estimation of the number of opposing lanes is based on the opposing links' saturation flow rates. A weighted average, based on the coded percentages of opposing flow, is used to compute the opposing saturation flow rate of up to three opposing links. If the opposing saturation flow rate is lower than 2000 vehicles per hour of green (vphg), then one opposing lane is assumed. If the opposing saturation flow rate is between 2000 and 4400 vphg, then two opposing lanes are assumed. If the opposing saturation flow rate is greater than 4400 vphg, then three opposing lanes are assumed. The user can optionally override these assumptions by manually coding the number of opposing lanes on the Edit > Optional > Permitted screen (or record type 29).

### FHWA/TRC Model

This is used as the permitted-movement model at unsignalized intersections, unless the user specifically chooses another model. This model, which was also featured for signalized intersections within TRANSYT-7F releases 4 through 6, was developed as a joint field calibration study conducted by the University of Florida Transportation Research Center (TRC) and FHWA. The TRC designed the study, FHWA collected the data, and the TRC performed the statistical analyses. The complete details of the study are documented in a separate report [13].

Based on the field calibration and subsequent regression analyses, the following general model form was selected:

$$MFR(t)_i = A_i \bullet \exp(-B_i \bullet v_o(t)^{C_i})$$

where,

$MFR(t)_i$  = the maximum flow rate for permitted traffic at time  $t$  for condition  $i$   
(see below)

$A_i$  = statistically derived intercept, MFR at zero opposing flow, for condition  $i$

$B_i, C_i$  = statistically derived model parameters

$v_o(t)$  = the opposing flow rate at time  $t$

Regarding "condition  $i$ ", individual models were developed for the following (right-hand drive) conditions:

- permitted left-turns
- protected-plus-permitted left-turns
- unsignalized, stop sign controlled movements

Additionally, models were developed for differing speeds, namely less than 40 mph (64 km/hr) and greater than or equal to 40 mph (64 km/hr); and for one opposing lane vs. two or more lanes. The complete family of models is given in the table below, which assumes right-hand drive.

For other permitted movements not explicitly given above, a linear model similar to the one in TRANSYT/9 [6] was used. This model and its coefficients are also in the table below.

Even though these models were based on field data, they are consistently more pessimistic than other models commonly used and a number of users have expressed concern with it. Accordingly, it was replaced with the ARRB model as the default model for signalized intersections in TRANSYT. The user may, of course, still optionally select these models.



Movement/ Condition	Opposing Lanes	Opposing Speed	Model		Coefficients			
			Form	Number	A	B	C	D
<b>FHWA/TRC Models:</b>								
Permitted-Only	1	<40 mph	E	11	1227	2.44E-02	0.71	-
Left-Turn		>=40 mph	E	11	1227	2.44E-02	0.71	-
	2+	<40 mph	E	12	1436	6.60E-04	1.22	-
		>=40 mph	E	13	1649	1.01E-02	0.83	-
Protected-Permitted	1	<40 mph	E	14	1463	6.60E-04	1.25	-
Left-Turn		>=40 mph	E	14	1463	6.60E-04	1.25	-
	2+	<40 mph	E	15	1577	1.97E-04	1.36	-
		>=40 mph	E	16	1423	2.66E-03	1.01	-
Unsignalized	1	All	E	17	1364	4.54E-03	0.94	-
Left-Turn (Artery)	2+	All	E	17	1364	4.54E-03	0.94	-
Free Right-Turns	All	All	L1	18	715	0.22	-	-
Unsignalized	1	All	L1	18	715	0.22	-	-
Cross Street Thru	2+	All	L1	18	715	0.22	-	-
Unsignalized	1	All	L1	18	715	0.22	-	-
Cross Street Left	2+	All	L1	19	600	0.21	-	-
<b>Minnesota Model:</b>	1	All	L2	31	1165	1.245	1.40E-05	-
	2	All	L2	32	1145	0.875	1.20E-05	-
<b>Ohio State Model:</b>	1	All	L3	41	1404	1.632	8.47E-04	2.14E-07
	2	All	L3	42	1393	1.734	9.17E-04	1.96E-07

where the model forms are as follows:

$$E : MFR = A \bullet \exp\left(-B \bullet v_o(t)^c\right)$$

$$L1 : MFR = A - Bv_o(t)$$

$$L2 : MFR = A - Bv_o(t) + Cv_o(t)^2 \bullet T$$

$$L3 : MFR = A - Bv_o(t) + Cv_o(t)^2 - Dv_o(t)^3$$

#### Texas A&M Model

This model [28] was developed by the Texas Transportation Institute (TTI) of the Texas A&M University system. The model is of the form:

$$MFR(t) = v_o(t) \frac{\exp\left(\frac{-v_o(t) \bullet T}{3600}\right)}{1 - \exp\left(\frac{-v_o(t) \bullet H}{3600}\right)}$$

where,

vo(t) = the opposing flow rate at time t  
H = minimum departure headway, in seconds  
T = acceptable critical gap, in seconds  
exp = natural base of logarithms

The model has two different sets of defaults for T and H: one based on field studies [28] and one based on simulation [29]. See the upcoming Model Calibration section for these default values.

#### Minnesota Model

This model [30] was developed at the University of Minnesota based on regression techniques similar to the FHWA/TRC model. It has the linear form:

$$MFR = A_i - [B_i v_o(t)] + [C_i v_o(t)^2 \cdot T]$$

where Ai, Bi, and Ci are regression coefficients for condition i (the number of opposing lanes). The regression coefficients are listed in the earlier table.

#### Ohio State Model

This model [31] was developed at Ohio State University and has a form similar to the Minnesota model, namely:

$$MFR = A_i - [B_i v_o(t)] + [C_i v_o(t)^2] - [D_i v_o(t)^3]$$

where Di is a fourth regression coefficient. The regression coefficients are listed in the earlier table.

#### HCM Model

The 1985 Highway Capacity Manual (HCM) model [12] may also be specified:

$$MFR(t) = 1400 - v_o(t)$$

#### Permitted Model Implementation

In step-wise simulation, the opposing flows can be obtained on a step-by-step basis from the most recent simulation of opposing links. On the other hand, under link-wise simulation, a three-pass procedure is needed because information about the opposing flows does not become available until a simulation pass of the whole system is completed:

1. The system is initially simulated without the permitted movement model. For permitted links, during permitted phases, the GO-pattern used is the user-coded (or default) MFR upper limit, which is based on the HCM model above. Further, the GO-pattern accounts for sneakers.
2. Following initial simulation (with the initial timing plan) and following the system-wide re-simulation at the end of each hill-climb step, the permitted movement model is exercised to re-simulate the permitted links only. The GO-pattern is recalculated according to the selected permitted movement model.
3. In the final simulation pass all unopposed links are re-simulated to correct the OUT-to-IN patterns that may have shifted somewhat in pass 2 above.

#### Sneakers

The treatment of sneakers is handled analytically, rather than empirically. It is assumed that sneakers, or the latent left-turns (assuming right-hand drive) that occur at the end of a permitted phase, discharge at a saturation flow rate equivalent to that of an unopposed movement. Thus the network-wide value of saturation flow rate (coded on the Edit > Optional > Global > Network Parameters screen) applies. Since the average number of sneakers per cycle is coded by the user, the time it takes to discharge the sneakers is calculated as:

$$t_s = \frac{3600 \bullet S}{s \bullet r \bullet N}$$

where,

ts = time to discharge sneakers, in steps  
 S = user-specified number of sneakers per cycle  
 s = saturation flow rate, in vphg  
 r = seconds per step resolution  
 N = number of left-turn lanes (assuming right-hand drive)  
     estimated based on the user-specified maximum flow rate

If a queue remains at the end of effective green, it can thus be discharged at rate S for ts steps. If no queue remains at the end of effective green, the specified sneakers may have no effect on results. For most cases, the user is cautioned to avoid coding sneakers for links that do not represent left-turn movements (assuming right-hand drive). In addition, sneakers should not be used as a substitute for the "extension of effective green" (EEG) parameter, nor should the EEG be artificially increased to allow for sneakers.

### Model Calibration

As stated earlier, the default standard permitted movement model for signalized intersections is one developed by the Australian Road Research Board (ARRB). For the overall network, the user may override this model and choose among *most* of the available permitted movement models as the new global default model. The desired global permitted movement models can be selected on the Edit > Optional > Global > Model Coefficients screen (or record type 8), using the "Global Number" in the table below. Model numbers 17, 18, and 19 are designed for minor stop-controlled movements, and cannot be selected as the global model. In addition, the table below assumes right-hand drive (e.g., "RTOR" to indicate the right-turns-on-red model), such that the "Permitted-only left-turn" model could be chosen to model permitted-only right-turns in a country with left-hand drive.

For individual movements, the user may override the global model and use *any* of the permitted movement models. The desired permitted movement models can be selected on the Edit > Optional > Permitted screen (or record type 29), using the model numbers in the table below.

Most TRANSYT permitted movement models can be calibrated for local conditions on the Edit > Optional > Global > Model Coefficients screen (or record type 8). Details of the model coefficients are listed in the table below, and some of their default values are listed in another table earlier (see FHWA/TRC model). Note that "Fld 2" in the table below (for record type 8) is associated with the first field of the Edit > Optional > Global > Model Coefficients screen. Also note that for this field, permitted movement model numbers are in the range of 100-150. For example, coding a model number of '11' requests the Penic fuel consumption model [32], whereas coding '111' requests calibration of the FHWA/TRC permitted-only model.

Calibration of the individual permitted movement models, for differing speeds and opposing lanes, is accomplished differently than the selection of an alternative global default permitted movement model. As discussed in the Global Screen documentation, to select a global default permitted movement model that is different than the ARRB model, the model number should be coded as '4' ("miscellaneous model parameters"), and the global permitted movement model number (see table below) should be coded as

the first "model coefficient". In this situation, global default permitted movement model numbers are in the range of 0-50.

The coding of this data must properly reflect the level of numerical precision ("Prec"). For example, in order to code HEADWY as 2.57 based on E4 precision, it would be necessary to code '25700'.

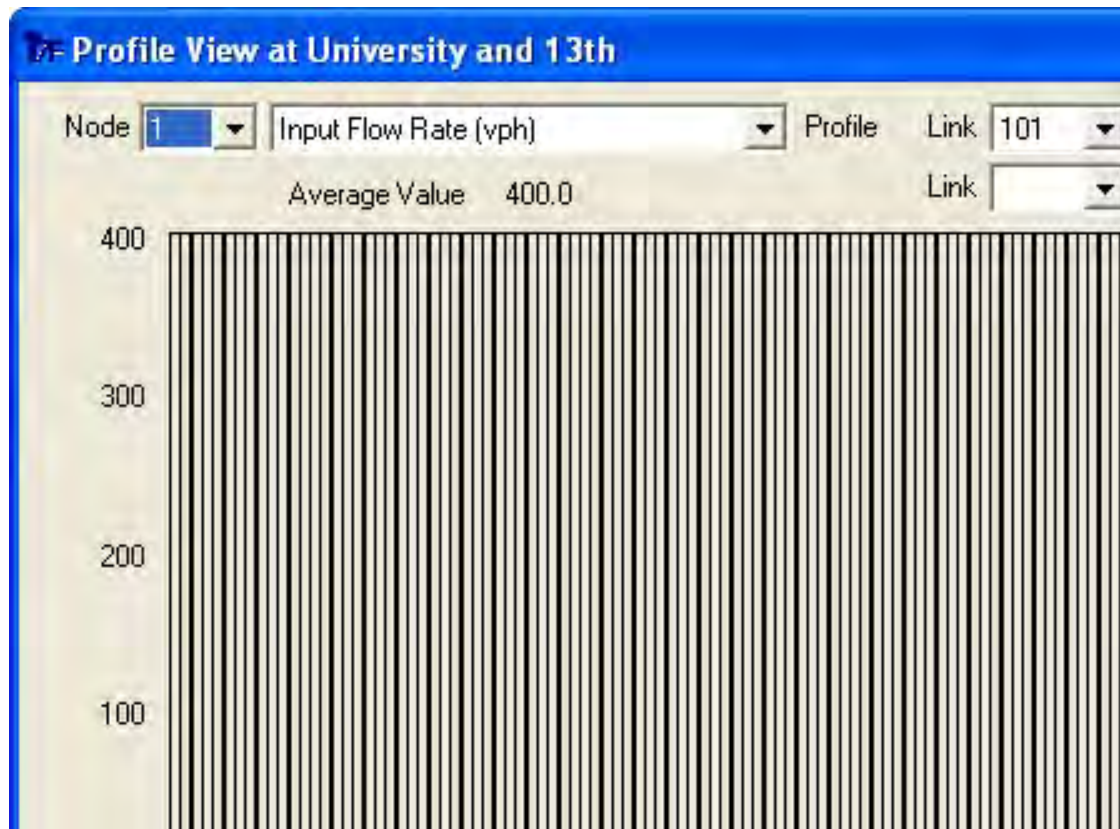
Fld	Variable	Description	Prec	Range	Default		
2	'100+X'	<b>Permitted Movement Models</b> where 'X' is the model number as defined below	--	100-150	--		
Model Type		Speed (mph):	1 Lane		2+ Lanes	Global No.	
			<40	>=40	<40	>=40	
ARRB Model			0				0
FHWA/TRC Model (all)							10
Permitted-only left-turn			11	11	12	13	
Protected + permitted left-turn			14	14	15	16	
Unsignalized left-turn (artery)			17	17	17	17	
RTOR & unsig. X-street thru			18	18	18	18	
Unsignalized X-street left-turn			19	19	19	19	
Texas A&M Model							
Field study parameters (all)			21				21
Simulation parameters (all)			22				22
Minnesota Model			31		32		30
Ohio State Model			41		42		40
HCM 1985 Model (all)			50				50
3+		the parameters depend on the model selected in field 2 they are summarized by model name and number below					
<b>ARRB Model (number 0)</b>							
3	HEADWY	minimum departure headway (H), sec.			E4	1-9999	2.57
4	CRTGAP	critical acceptable gap (T), sec.			E4	1-9999	4.00
<b>FHWA/TRC Models (numbers 11-19)</b>							
3	IMDLCA(I)	model coefficient 'A'			E0	1-99999	see table
4	RMDLCB(I)	model coefficient 'B' for models 11-17			E6	1-99999	see table
4	RMDLCB(I)	model coefficient 'B' for models 18 and 19			E5	1-99999	see table
5	RMDLCC(I)	model coefficient 'C'			E4	1-99999	see table
<b>Texas A&amp;M Model Field Data (number 21) / Simulation (number 22)</b>							
3	HEADWY	minimum departure headway (H), sec.			E4	1-9999	2.5/3.6
4	CRTGAP	critical acceptable gap (T), sec.			E4	1-9999	4.5/5.4
<b>Minnesota Model (numbers 31 and 32)</b>							
3	IMDLCA(I)	model coefficient 'A'			E0	1-99999	see table
4	RMDLCB(I)	model coefficient 'B'			E3	1-99999	see table
5	RMDLCC(I)	model coefficient 'C'			E7	1-99999	see table
6	CRTGAP	critical acceptable gap (T), sec.			E4	1-99999	4.50
<b>Ohio State Model (numbers 41 and 42)</b>							
3	IMDLCA(I)	model coefficient 'A'			E0	1-99999	see table
4	RMDLCB(I)	model coefficient 'B'			E3	1-99999	see table
5	RMDLCC(I)	model coefficient 'C'			E7	1-99999	see table
6	RMDLCD(I)	model coefficient 'D'			E10	1-99999	see table

## Platoon Dispersion

The start-and-stop operation of signals tends to create platoons of vehicles that travel along a

link. TRANSYT models the dispersion of these platoons as they progress along a link. In link-wise simulation, vehicle arrivals are assumed to occur at the downstream stop line. In step-wise simulation, vehicle arrivals are assumed to occur at the dynamic back of queue location.

Platoon dispersion is only simulated on internal links having a specific upstream signal. On external links, where no feeding links are defined, vehicle arrivals are assumed to be random in nature, resulting in a uniform flow profile over the cycle. The flow profile illustrated below represents the type of uniform flow profile to be expected on external links.



On internal links, for each time interval (step),  $t$ , the downstream arrival flow is determined by the following recurrence equation:

$$v'_{(t+\beta T)} = F \cdot v_t + [(1 - F) \cdot v'_{(t+\beta T-1)}]$$

where,

$v'(t+BT)$  = predicted flow rate in time interval  $t+BT$  of the predicted platoon

$v_t$  = flow rate of the initial platoon during step  $t$

$B$  ("beta") = an empirical factor, generally 0.8 (which can be calibrated on the Edit > Optional > Global > Model Coefficients screen)

$T$  = the cruise travel time on the link, in steps

$F$  = a smoothing factor, where:

$$F = (1 + \alpha \cdot \beta \cdot T)^{-1}$$

where,

a ("alpha") = an empirically derived constant, called the platoon dispersion factor (PDF)

Note that the T using the above equation is the cruise travel time on the link from the upstream stop line to the downstream stop line (link-wise), or from the upstream stop line to the downstream back of queue (step-wise).

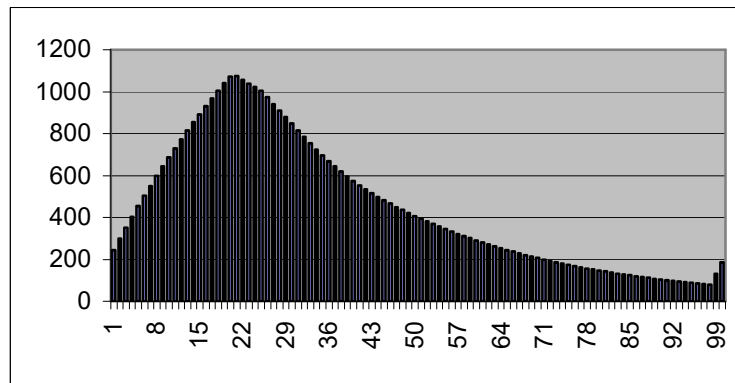
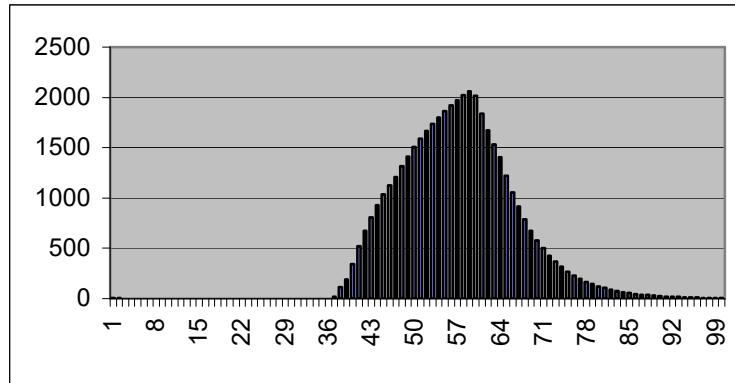
The effect of the platoon dispersion factor (PDF) is based on field studies by Hillier and Rothery [8]. The factor alpha has been found by researchers to best represent measured dispersion on typical urban streets in the U.S. when it was set at 0.35. TRANSYT assumes a default value of 0.35, which is also based on empirical studies performed by the Transport and Road Research Laboratory in the United Kingdom [3]. This PDF will vary to consider site-specific factors such as grade, curvature, parking, opposing flow interference, and other sources of impedance. It should be noted that TRANSYT uses a different formula to disperse buses, which accounts for the more sluggish movement of buses at bus stops along a link (if any) and in regular traffic, regardless of whether the bus is moving in mixed traffic or in its own lane.

Users are encouraged to calibrate the PDF so as to achieve more realistic results. The PDF value can be adjusted globally on the Edit > Optional > Global > Network Parameters screen (record type 10), or locally (link-specific) on the Edit > Optional > Weighting > Platoon Dispersion Factors screen (record type 39). Ideally, the PDF would be calibrated such that flow profiles from the model would closely match observed flow profiles in the field. In the absence of field data, research from PRC Engineering [17] and the University of Florida [20] suggests that the table below could be used to obtain reasonable estimates for the PDF.

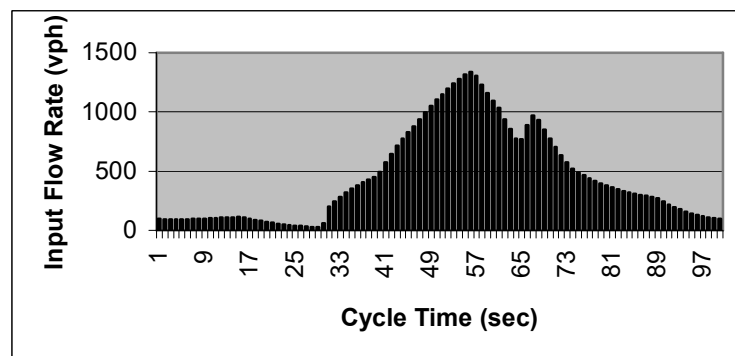
PDF	Roadway Characteristics	Description of Conditions
0.5	heavy friction	Combination of parking, moderate to heavy turns, moderate to heavy pedestrian traffic, narrow lane width. Traffic flow typical of urban CBD.
0.35	moderate friction	Light turning traffic, light pedestrian traffic, 11 to 12 foot lanes, possibly divided. Well-designed CBD or fringe arterial.
0.25	low friction	No parking, divided, turning provision, 12 ft. lane width. Suburban high-type arterial.

Flow profile diagrams and other diagrams can be requested within the TRANSYT input file. The Profile View screen and Spyglass utility program allow the user to view flow profiles and ascertain the degree of platoon dispersion on any given link. The Spyglass file "inpat.out" contains the step-wise input flow rate for each link during each second of simulation. In addition, the Spyglass file "travel.out" contains the step-wise travel times from the upstream stop line to the subject link's dynamic back of queue location.

The diagrams below illustrate the nature of platoon dispersion. The first diagram illustrates a platoon after it has traveled one quarter-mile, or 1320 feet (402 m), after being stopped at the upstream signal. The second diagram illustrates the same platoon after traveling one full-mile, or 5280 feet (1608 m). Notice that after a full-mile, the most intense portion of the platoon is a rate slightly higher than 1000 vehicles per hour (vph), whereas after a quarter-mile the most intense portion of the platoon is approximately 2000 vph. In addition, after a full-mile, the platoon has spread out to cover a larger portion of the cycle.



The shape of the platoons can become more complex when there are multiple feeding links. The diagram below illustrates a sample flow profile diagram on an internal link having multiple feeding links.



The impact of platoon dispersion on traffic network performance is significant, which is one of the reasons that TRANSYT is an effective model. The degree of platoon dispersion has an intimate effect on the percentage of vehicles arriving on green, which in turn directly affects uniform vehicle delay, stops, queuing, and other measures of effectiveness. The degree of platoon dispersion also affects traffic-actuated phase times. Because of all this, platoon dispersion has a significant impact on what the optimal signal timing plan must be.

If the user tries to calibrate the TRANSYT platoon dispersion model to match observed flow profiles in the field, or uses time delay moving vehicle runs to verify travel times obtained from the model,



it is possible that the PDF alone might not be sufficient to reconcile the two sets of results. The intensity of the platoon should be correct, as should the time at which the most intense part of the platoon arrives. In this case, it is advisable to double-check the link-to-link flows, as well as the cruise speeds, coded on the Edit > Feeders screen (or record type 28). In addition, mid-block source flows (coded on the Edit > Traffic screen or record type 28) could be disrupting progression in either the model, or in the field. Finally, field measurement of average traffic-actuated phase times (across several signal cycles) is recommended, to see if these agree with the average phase times coded within TRANSYT.

### Platooning Index

There are several other aspects of the flow profiles that are of interest. The first is the platooning index, defined as follows:

$$Plt\ Index = \frac{\sum_t |v_t - v|}{C \bullet v}$$

where,

Plt Index = a measure of deviation of the periodic flow rate from the average flow rate over the entire cycle

$v_t$  = flow rate at step  $t$

$v$  = average flow rate for the cycle

$C$  = cycle length in steps

The platooning index is an output of the program. It is a value between 0.0 and 2.0. A perfectly uniform flow would have an index of zero, since there is no variation. A high value of the platooning index indicates bunching of traffic and suggests that such links will benefit from coordination. Links with low platooning index values may be suitable for locating section boundaries, if it is necessary to partition the network.

### Platooning Moment

In release 8, TRANSYT-7F added a moment-based platooning index that is based on the concept of moment in the field of mechanics [34,35]. In the context of a traffic platoon, moment is the product of flow rate and time. The moment-based platooning index (Plt Momt) is defined as follows:

$$Plt\ Momt = (1 - R_{PM}) \times 2$$

where  $R_{PM}$  is the platooning moment ratio, defined as the ratio of the minimum platooning moment ( $M_{min}$ ) divided by the average platooning moment ( $M_{avg}$ ). The factor of 2 is added to the equation above so that the platooning moment will have the same range as the platooning index described earlier. The minimum and average platooning moments are further defined below.

The platooning moment of one step is determined by the sum of the moments of other steps with respect to this specific step. For example, for a cycle length that is divided into  $N$  steps, if  $N$  is an even number, the platooning moment of one step can be determined by the sum of the moments of  $(N/2)-1$  steps backward, plus the sum of the moments  $N/2$  steps forward; if  $N$  is an odd number, the number of steps calculated is  $(N-1)/2$  for both directions. Using this method, the platooning moment of each step can be calculated and the resulting minimum value is the minimum platooning moment.

The average platooning moment is derived from the average flow (i.e, uniform flow), computed as follows:

$$M_{avg} = \left(\frac{N}{2}\right)^2 \times F_{avg} = \frac{N}{4} \sum_{i=1}^N F_i \quad \text{when } N \text{ is an even number}$$

$$M_{avg} = \left(\frac{N^2 - 1}{4}\right) \times F_{avg} = \frac{N^2 - 1}{4N} \sum_{i=1}^N F_i \quad \text{when } N \text{ is an odd number}$$

where,

$M_{avg}$  = average moment of the flow profile

$N$  = number of steps per cycle

$F_i$  = flow rate at  $i$ th step

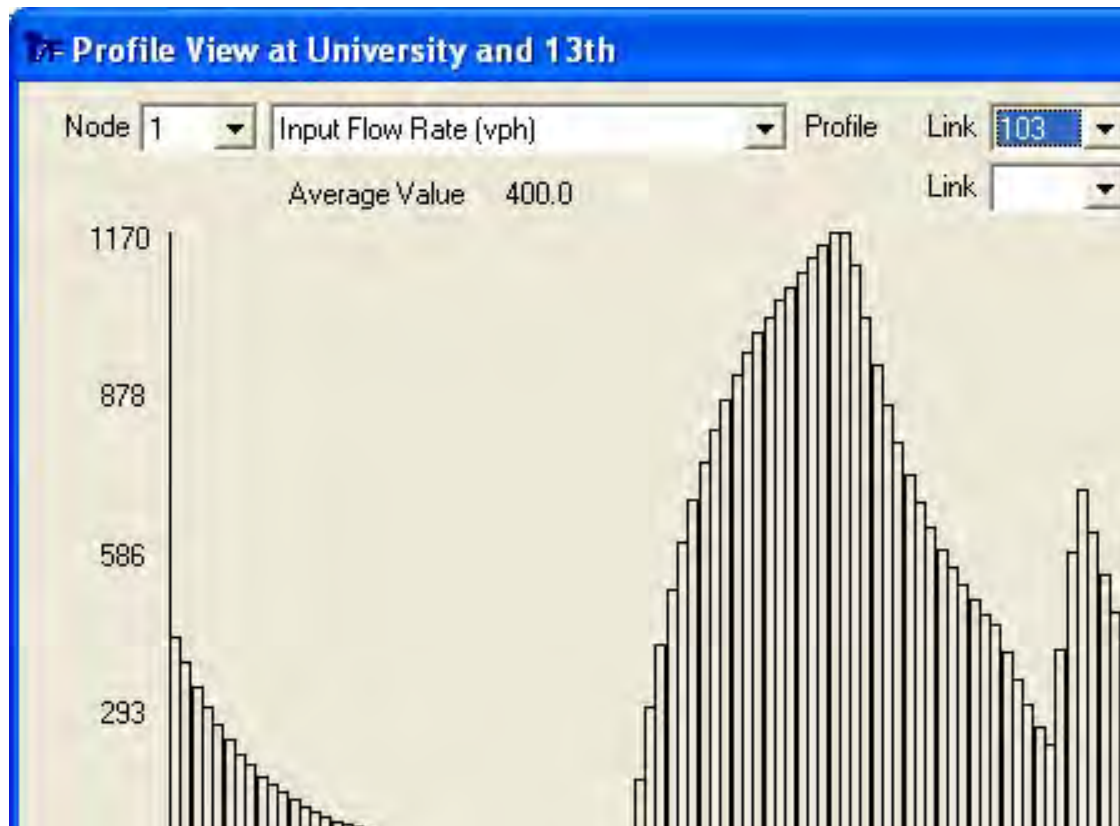
$F_{avg}$  = average flow rate of the flow profile, calculated as:

$$F_{avg} = \frac{1}{N} \sum_{i=1}^N F_i$$

Although the platooning index and platooning moment both aim to measure the concentration of the platoon, research has shown that the platooning moment is a more robust measure than the platooning index, because it is able to describe certain types of platoons more accurately [35].

## Profile View

The Profile View screen can be used to view static graphics that describe TRANSYT simulation results. These results must initially be generated by running the Spyglass utility program (by clicking on the "bar graph" icon, or selecting Run > Spyglass). The user can select View > Profiles to immediately load the Profile View screen without re-running Spyglass, if the raw simulation data have already been generated recently. This screen can be sent to the printer, or to an image file, by pressing Ctrl-P. For tutorial examples involving the use of this screen, refer to the Progression Analysis Examples, Queue Spillback and Queue Spillover Examples, Multi-Period Analysis Examples, and Sign Control Examples sections of the documentation.



Profile View simulation results can be used to illustrate single-cycle step-wise simulation, multi-cycle step-wise simulation, or multi-period step-wise simulation. The Profile View cannot be used to illustrate link-wise simulation results. Note that multi-period simulation is automatically a form of multi-cycle simulation. Also note that flow profile information is not displayed for the simulation initialization cycle. On the X-axis, the network-wide background cycle length is used to compute flow profile dimensions. Therefore, for uncoordinated intersections, the X-axis dimensions may not reflect the local cycle length.

To operate this screen, the user first selects their desired node number within the network. Second, the user selects their desired profile. The available flow profiles include input flow rate, output flow rate, saturation flow rate, uniform queue length, maximum back of queue length, link full flag, and intra-link travel time. Finally, the user selects their desired links for plotting flow profiles. Up to four links can be selected if the user wishes to view aggregated results for an entire lane group, or an entire approach, containing multiple links. Link numbers are designated in accordance with the link numbering scheme. If needed, the user can load the Edit > Traffic screen to observe which link number is associated with which turning movement.

Note that the step-wise maximum back of queue profile illustrated in this screen is actually associated with the uniform maximum back of queue listed in the legacy (\*.TOF) report. The overall maximum back of queue for the entire analysis period is also reported in the legacy output file, and incorporates the random queue component. However, the maximum back of queue profile illustrated in this screen is not associated with the uniform number of queued vehicles that can be used to compute uniform delay (d1). The area under the uniform queue profile illustrated in this screen can be used to compute uniform delay (d1).

The Average Value displayed at the top of the screen represents the average profile value across every step of simulation. This can be useful for validating some of the simulation results. For example,

the average input flow rate value of 400 vph shown in the screen image above suggests that the coded volume for this link was 400 vph, despite the platoon dispersion that temporarily caused flow rates as high as 1170 vph. The average saturation flow rate value can be used to determine the capacity of any link, which can further be used to verify the degree of saturation. The average Link Full Flag value could be used to determine the Time Full%, the average uniform queue length could be used to determine uniform delay, and so forth.

## PROS

PROS is an acronym for progression opportunities [9,10], and represents the opportunities for vehicles to progress through multiple signals on green. PROS is one of the available performance indexes or objective functions available for selection for TRANSYT optimization. PROS counts the time slices in which a vehicle (or vehicles) can proceed through a downstream intersection without stopping. Therefore, a particular signal timing plan can potentially achieve a high value of PROS even when the available bandwidth does not stretch from one end of the arterial street to the other. This differs from bandwidth optimization programs such as PASSER, in which the program must obtain a green band all the way from one end of the arterial street to the other.

PROS represent the opportunity to progress between two or more intersections. The number of PROS in a given direction, for a given time period (or step), are the number of successive green signals that will be encountered at the design speed without stopping. The aggregate PROS is found by summing the PROS overall time periods in both directions:

$$PROS = \sum_{k=1}^2 \sum_{j=1}^N \sum_{t=1}^C PROS_{kjt}$$

where,

k = direction of travel

j = intersection number, of which there are N

t = time in units common to the model, up to a cycle length, C

A subset of the above is to consider only the PROS<sub>kjt</sub> where t ranges from Tk1 to Tkt and these limits respectively represent the leading and trailing edges of the arterial through green bands in each direction. Optimization of this subset would have the effect of explicitly optimizing maximal bandwidth.

Another measure related to the PROS concept is defined as cycle progression opportunities (CPROS). In the absence of any signal, the full cycle would be available as progression opportunities. This hypothetical measure is obtained for both directions as follows:

$$CPROS = C \bullet N(N-1)$$

The ratio PROS/CPROS, called effective PROS (PROSe), is analogous to the bandwidth efficiency reported by programs such as PASSER. By nature of the definition, PROSe will always be less than 1.0.

TRANSYT can be instructed to use performance indices such as Disutility Index (DI-only), PROS-only, PROS & DI, or PROS/DI. In reality, whenever PROS are used, it is really PROSe, because otherwise in a cycle length optimization the PROS would always rise with the cycle length.

The weighting strategy for directions and arteries employs a modification to the formulas:

$$PROSR_i = \frac{PROS_{i1}}{PROS_{i1} + PROS_{i2}}$$

and:

$$WDR_i = \frac{WD_{i1}}{WD_{i1} + WD_{i2}}$$

where,

PROSR<sub>i</sub> = relative PROS in the "forward" direction (k = 1) on artery i

PROS<sub>ik</sub> = actual PROS in direction k on artery i

WDR<sub>i</sub> = relative weighting for the "forward" direction (k = 1) on artery i

WD<sub>ik</sub> = weighting factors for direction k on artery i

The desired directional factor for artery i is defined as:

$$DDF_i = \frac{\min(PROSR_i, WDR_i)}{\max(PROSR_i, WDR_i)}$$

and the resulting definition of the effective PROS (PROSe) is:

$$PROS_e = \frac{\sum_{i=1}^A DDF_i \sum_{k=1}^2 WD_{ik} \sum_{j=1}^{N_i} \sum_{t=1}^C PROS_{ikjt}}{\sum_{i=1}^A \sum_{k=1}^2 WD_{ik} \sum_{j=1}^{N_i} \sum_{t=1}^C CPROS_{ikjt}}$$

where,

A = number of arteries

j = intersection number, of which there are N<sub>i</sub> on the artery i

t = time in the cycle, of which there are C seconds

For straight PROS optimization (referred to as "PROS-only"), the resulting performance index is simply equal to PROSe.

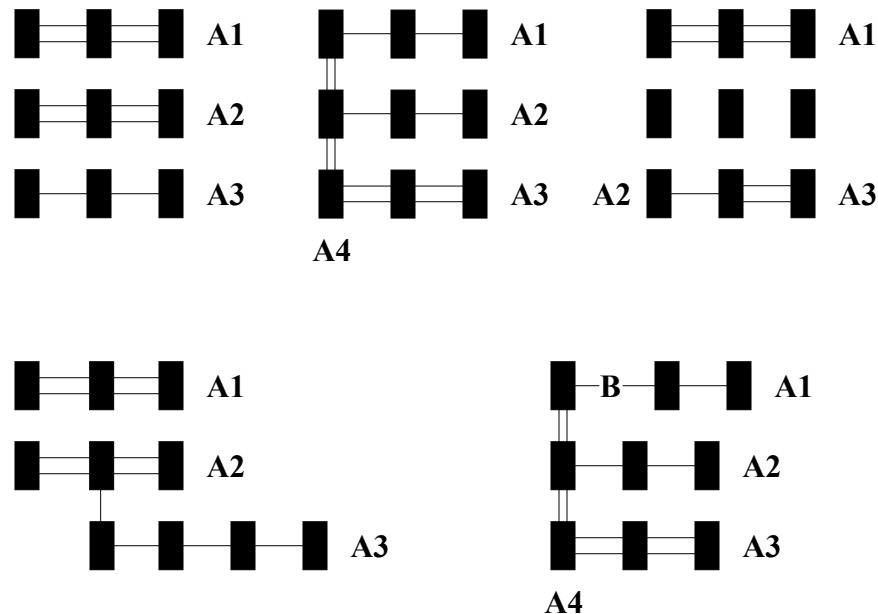
Thus, weighting the arteries relative to each other and/or the two directions of travel along each artery may allow the user to achieve a desired design or policy. For the PROS/DI objective function, there is also the ability to apply relative weights to PROS, vis-a-vis the DI, and to prioritize certain arteries and/or directions. Refer to the PROS Optimization section of the documentation for more details.

The text-based Time Location Diagram (TLD) within the TRANSYT legacy report displays some of the PROS calculations. The TLD flag is coded on the Edit > Optional > Route > Route Parameters screen. If the Output Quick Jump is used to jump to the Route Performance section of the report, the TLD can be found toward the bottom. The total PROS amount in each direction can be found by jumping to the Progression section of the legacy report.

### Proper Arteries for PROS

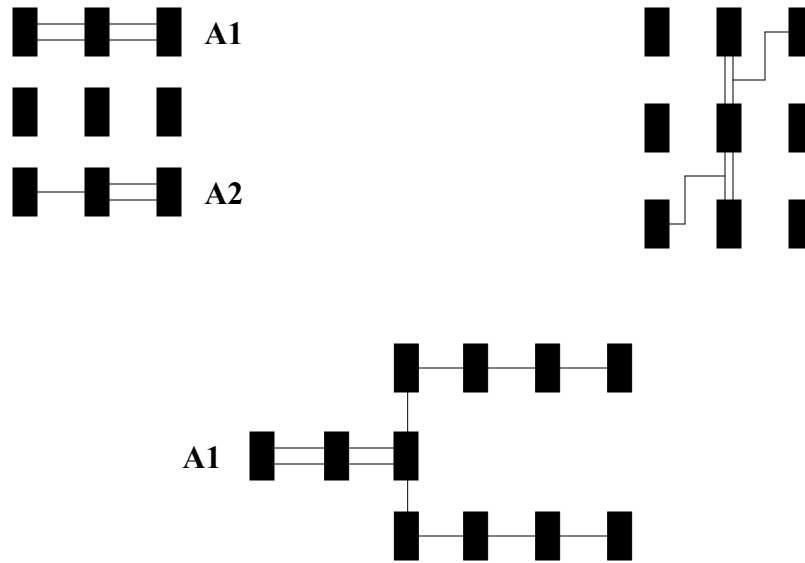
The figure below illustrates examples of proper arteries for PROS. Naturally, any linear artery of intersections within the network, such as artery "A1" in the upper left-hand corner of the figure, can be analyzed on its own. Networks with multiple arteries, such as A1/A2/A3 in the upper left part of the figure,

can also be analyzed (or optimized) in terms of PROS. It is perfectly acceptable for one-way and two-way arteries to intersect, as in the top center part of the figure. Arteries A2 and A3, shown in the upper right part of the figure, must be defined as separate arteries. On the bottom left of the figure, all three arteries must be defined separately. Arteries may turn at corners, so long as the "opposing" links enter the same node. The bottom right of the figure shows that bottlenecks and 100% green links may be included, and that arteries may be of any length.



### Improper Arteries for PROS

The figure below illustrates examples of improper arteries for PROS. The upper left part of the figure shows that mixing one-way and two-way streets within one artery (A2) is not allowed. The upper right part of the figure is intended to demonstrate that no more than four "artery links" may enter an intersection. Finally, the bottom part of figure implies that artery A1 must be split into three arteries, because mixing of one-way pairs with two-way streets is not allowed.



## Percent of Vehicles Arriving on Green

An important measure used capacity analysis, including the Highway Capacity Manual [12], is the percentage of vehicles arriving on green (PVG). The PVG is reported along with the platooning index on the flow profile diagrams. The Arrival Type output parameter is also used to reflect the PVG, and is reported in the wide, 132-column output format. TRANSYT simulates the flow profile very carefully, so it can use the following equation to compute the PVG:

$$PVG = \frac{v_g}{v}$$

where,

$v_g$  = volume arriving during effective green as measured by the flow profile  
 $v$  = total volume

## Queue Service Time and Green Extension Time

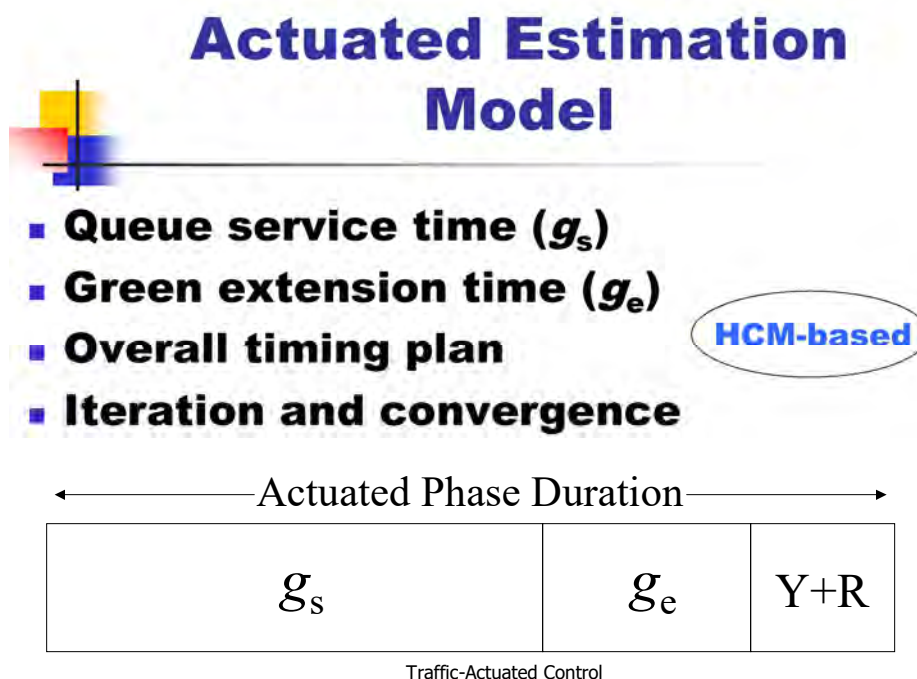
The traffic-actuated signal timing estimation model within TRANSYT-7F shares a common, underlying computational foundation with a similar model from the Highway Capacity Manual [12]. Specifically, the queue service time -- green extension time structure for computing individual actuated phase times, along with a procedure of iteration and convergence, is documented within HCM2000 Chapter 16 (Appendix B). TRANSYT-7F now implements a similar underlying structure for estimating actuated phase times. However, the queue service time and green extension time models recommended for use within TRANSYT-7F are markedly different, because they have been enhanced for network-wide application, as opposed to isolated intersection application. The methodologies for critical link search and green time reallocation are also different, although these are not documented within this section.

The queue service time model calculates the amount of time required to serve the initial queue,

after the signal has turned green. The "T7F" selection (on the Edit > Optional > Actuated screen) indicates that queue service times will be obtained directly from TRANSYT step-wise simulation. Other models available for selection are existing models from the literature. The "T7F" model has been implemented as the default model within the actuated estimation model based on its modeling structure and its correlation with CORSIM. Queue service times are taken directly from the simulated uniform queue profile, which can be plotted using Spyglass and the Profile View screen. During this process, an additional adjustment is made to account for stochastic effects. Numerically computed queue service times are reported in the estimation output file "t7fact.out".

The green extension time model calculates the amount of time until a suitable gap has been located at the detector, following the queue service time. Models whose names begin with the letters "GP" were developed exclusively for use with TRANSYT-7F. Models whose names begin with the letters "GL" are existing models from the literature. The "GP1" model has been implemented as the default model for actuated estimation based on its modeling structure and its correlation with CORSIM. Green extension times are computed using applied probability analysis of the simulated input flow rate profile, which can be plotted using Spyglass and the Profile View screen. During this process, flow profile adjustments are made to account for stochastic effects. Numerically computed green extension times are reported in the estimation output file "t7fact.out".

Technical aspects of the queue service time and green extension time models (illustrated below) are summarized in Appendix A [37], and detailed in the supplemental research report "Modeling Traffic-Actuated Control with TRANSYT-7F" [36].

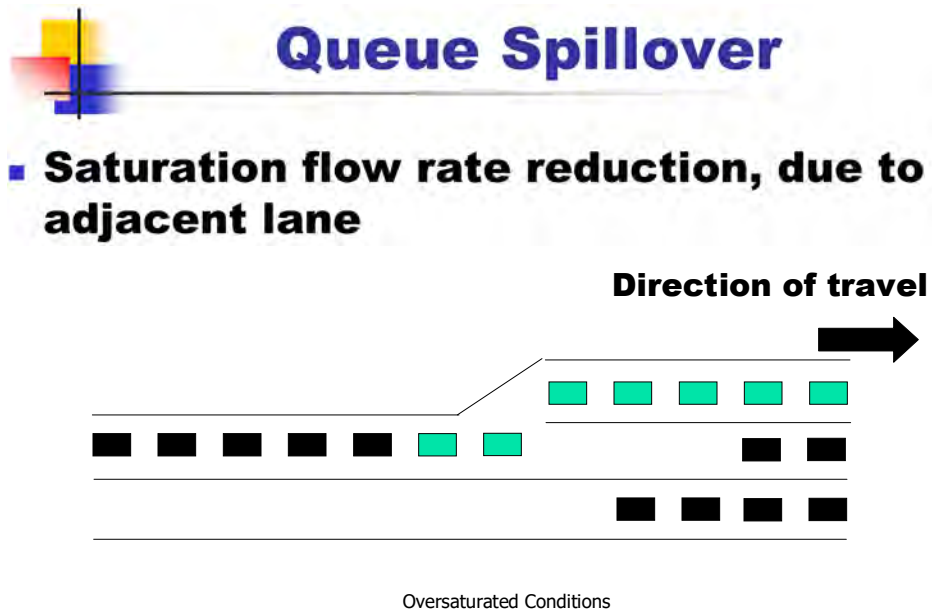
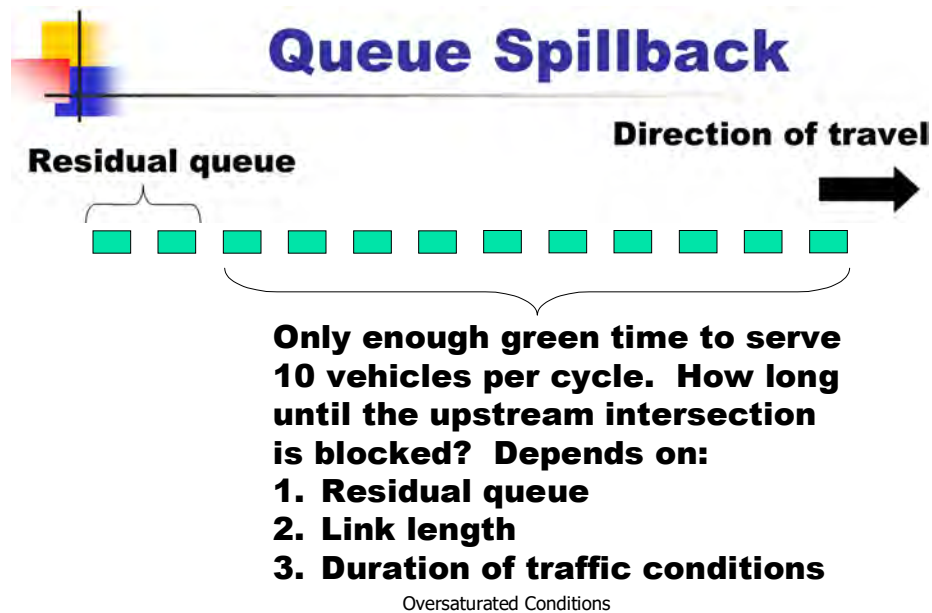


## Queue Spillback and Queue Spillover

"Queue spillback" indicates an extension of the queue into the upstream intersection, which may disturb operations at the upstream intersection. "Queue spillover" indicates an extension of the queue into an adjacent lane, usually caused by a turn bay or turn pocket, which may disturb operations in the



adjacent through lane. The figures below illustrate the operation of queue spillback and queue spillover.



In link-wise simulation, queue spillback is not considered; however, the user can request for the excess maximum back of queue to be considered in the optimization process. It is important to note that

if spillback actually does occur, link-wise simulation does not give realistic results. In this case, step-wise simulation should be used.

In step-wise simulation, spillback due to saturated downstream links and/or adjacent turn bays is explicitly considered by reducing the GO-patterns of affected links. In addition, links that are fully occupied during a given step get an IN-pattern of zero in that step.

The value for maximum back of queue capacity, or queuing capacity, may be coded on a link-by-link basis, or the program will calculate a value for each link. If an explicit value is not coded, the program will calculate a capacity as follows:

$$qc_i = \frac{L_i \bullet N_i}{H}$$

where,

qci = queuing capacity on link i

Li = length of link i in feet or meters

Ni = number of lanes on link i, derived from the coded saturation flow rate

H = average vehicle spacing in the queue (default 25 feet or 7.6 meters)

In release 8, when TRANSYT-7F was upgraded to simulate oversaturated conditions, adjustments were made to the calculation of queuing capacity. Under link-wise simulation, since it was not possible to explicitly consider oversaturated conditions, an automatic reduction (20%) to the queuing capacity was implemented. The adjustment is intended to make traffic flow patterns more realistic in response to random queue and queue blockage effects. Under step-wise simulation, a more detailed queuing capacity adjustment, based on degree of saturation and duration of simulation, was implemented.

Input parameters such as the Average Vehicle Spacing, and the Queuing Capacity Multiplier, can be used to calibrate queuing behavior for individual links or the entire network. In addition, the Queuing Capacity input parameter can be used to override the internally computed values. Queuing data can be coded on the Edit > Optional > Queuing screen (or record types 28 and 291).

TRANSYT reports the number of links with maximum back of queue exceeding queuing capacity. This measure of effectiveness provides an indication of the level of spillback problems experienced by the overall system or route. A positive value for this measure of effectiveness indicates the existence of links with spillback problems. Several of the available objective functions are especially designed for optimization of oversaturated networks.

For a tutorial analysis, refer to the Queue Spillback and Queue Spillover Examples section of the documentation.

## Sensitivity Parameters

Previous versions of TRANSYT-7F have allowed the user to customize the hill-climb optimization sensitivity parameters. This can still be accomplished in the current release, but this is no longer recommended or fully supported. The default optimization sensitivity parameters used by the program are effective for most practical applications, and genetic algorithm optimization is now available for better performance. The TRANSYT-7F Edit screens are not capable of generating the necessary inputs, so it is necessary to use the record type editor.

Sensitivity parameters, and the process described in this section, are only applicable to hill-climb optimization using link-wise simulation, and are not applicable to step-wise simulation, in the current

version of the program. Sensitivity parameters may be coded by the user on record type 6.

Since it is based on TRANSYT/7, TRANSYT-7F offers a significant improvement over earlier versions in terms of computer run time. This is due to the addition of the sensitivity parameter. The sensitivity parameter affects the "distance" through which changes in the flow profiles of upstream links will be propagated downstream.

Consider that each time a signal offset or phase length is changed, the change will affect the OUT-patterns and thus the arrival patterns at downstream intersections. In earlier versions of TRANSYT (prior to TRANSYT/7) this effect was propagated throughout the network for each iteration. Clearly, the propagated effect of a change at a single node may not necessarily extend too far away from the subject node, perhaps only as far as one or two nodes.

For any link the OUT-pattern is stored as a flow rate for each time increment (step),  $t$ . This may be represented by  $X_t$ . Similarly, the OUT-patterns may be calculated after a timing change at some node, represented by  $X'_t$ . The change in the profiles from the previous timing ( $X_t$ ) to the revised timing ( $X'_t$ ) is computed using the following formula:

$$d_x = \sum_t \frac{|X_t - X'_t|}{X_t}$$

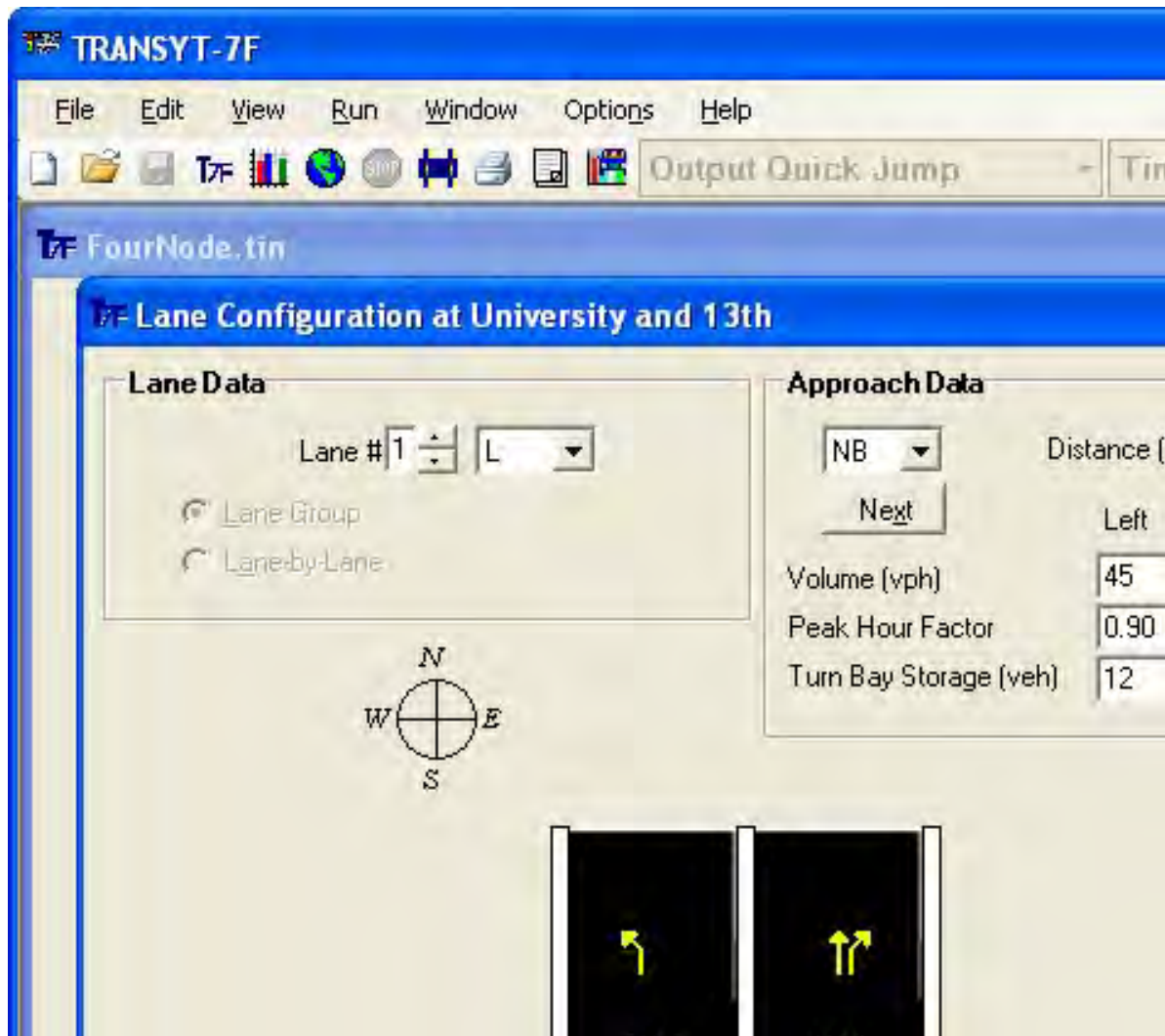
The computed value,  $dx$ , which represents the degree of change in the profiles (expressed in percent), is compared with the sensitivity parameter. When  $dx$  is greater than the current sensitivity parameter, new profiles for the next downstream link are calculated. If  $dx$  is smaller, the change in profile is considered negligible and the propagation along this link "branch" is terminated.

## Shared Lanes Methodology

The current version of TRANSYT contains two different models for shared lanes. The first model, which has been available within the software for many years, is designed to analyze shared lanes where one of the turning movements is permitted (i.e., yields to opposing or conflicting movements). The second model, which is relatively new, is designed to analyze shared lanes where none of the turning movements are permitted (a.k.a. "protected shared lanes"). For both models, a "primary" link number is typically used to represent the thru movement, and "secondary" link numbers are used to represent any turning movements. In the output files (summary reports and legacy report), results listed for the primary link reflect performance of the entire shared lane group, including all turn movements.

To invoke either one of these two models, the user must simply paint a shared lane within the graphical Lane Configuration screen, and then specify the appropriate phasing sequence (and turn movement protection) in the Timing View screen. Both models require the coding of primary and secondary link numbers on the Edit > Optional > Global > Shared Lanes screen (or record type 7). However, when the Lane Configuration screen is used, these shared link numbers are coded automatically by the graphical user interface (GUI).

In the United States, where vehicles drive on the right-hand side of the road, permitted shared lanes usually contain the thru+left movements, and the left-turn is permitted. Protected shared lanes could contain left-turn movements, or right-turn movements, or both, depending on the phasing sequence. In TRANSYT, if right-turns yield to pedestrians but not any traffic movements, these right-turns are not coded or modeled as permitted movements. Instead, they are coded and modeled as protected movements, and a saturation flow rate adjustment (to reflect pedestrian impedance) could be applied. The Lane Configuration screen below illustrates an exclusive left-turn lane next to a protected shared right-turn lane.



Finally, there is also a third technique available for modeling shared lanes, but without using either one of the above-mentioned models. Under this technique, the user manually codes a combined shared lane volume and adjusted saturation flow rate for one link, but no other turning movements or secondary shared links are coded. There is no significant modeling disadvantage to this technique, but it makes the input data coding process more difficult. The Lane Configuration and Timing View screens do not display the secondary turning movement, and the automatic flow balancing process (from the Map View, and the Feeders screen) does not account for the shared turning volume (the user might need to code secondary feeding flows manually).

For example, considering the shared right-turn lane illustrated above, the combined volume for northbound thru link 101 would be  $360 + 45 = 405$  vph. The manually adjusted saturation flow rate for link 101 would be  $(360 \cdot 1800 + 45 \cdot 1800 \cdot 0.85) / 405 = 1770$  vphg. The secondary shared link, northbound right-turn link 109, would be either completely removed from the dataset, or not coded to begin with.

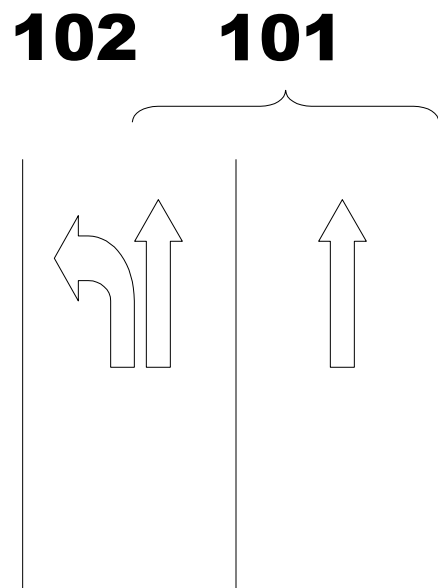
Until recently, this had been the long-standing technique recommended for coding and modeling protected shared lanes in TRANSYT. This technique would circumvent the original shared lanes model designed to handle permitted shared lanes. If a user tried to model protected shared lanes using the original model, results would often be unrealistic. In the new version though, it should always be possible

to obtain realistic results for shared lanes under step-wise simulation, regardless of which coding technique is used.

### Permitted Shared Lanes Methodology

In the U.S. (right-hand driving), the most common case of a permitted shared lane is that of a shared through and left-turn lane, where only the left-turning vehicles must yield to opposing traffic. Modeling of a shared, permitted left-turn and through lane is a very complex issue. The interaction of turning movements, subject to opposing throughs, and concurrent through traffic is highly stochastic.

When one of the turning movements within a shared lane is a permitted movement, it is necessary to assign unique link numbers to the turning movements, in order to simulate the different behavior of each movement. Performance is estimated for the entire shared lane group, based on the individual movement behaviors. For example, results for the shared lane group are listed in connection with the primary (protected) shared stopline movement (e.g., link 101 shown below).



To model this complex condition, the shared lanes model from the 1985 Highway Capacity Manual was investigated. While it is oriented toward determining capacity rather than periodic maximum flow rates, this procedure was adapted within TRANSYT to approximate permitted shared lane conditions. The adapted procedure can only handle one permitted link in the lane group. In addition, sign-controlled links cannot be simulated under this procedure. Refer to the Sign Control Examples section of the documentation for an example of how to model permitted shared lanes at unsignalized intersections. The permitted shared lanes procedure also determines whether the shared lane is likely to operate as a "de-facto" left-turn lane, as per the Highway Capacity Manual (HCM) [12].

The procedure is to assign the GO-patterns to be consistent with the HCM shared lanes model. The through (unopposed) component of the shared lane has a GO-pattern calculated as:

$$GO_u = S_s \bullet \frac{v_{us}}{v_e}$$

where,

GO<sub>u</sub> = GO-pattern (vphg) for the unopposed link(s) in the shared lane  
(any other lanes are added at their full saturation flow rate based on S<sub>s</sub> below)  
S<sub>s</sub> = ideal saturation flow rate (vphg) assuming unopposed traffic  
(calculated by dividing the coded saturation flow rate by the number of lanes)  
v<sub>e</sub> = total effective shared lane volume (vph)  
(with left-turns adjusted according to the 1985 HCM)

$$v_e = \frac{v_u + E_L \bullet v_l}{N}$$

where,

v<sub>u</sub> = total unopposed volume on the link(s) (vph)  
E<sub>L</sub> = equivalent left-turn factor per the 1985 HCM  
v<sub>l</sub> = permitted turn volume (vph)  
N = number of lanes on the approach

v<sub>us</sub> = unopposed volume in the shared lane (vph)

$$v_{us} = v_e - E_L \bullet v_l$$

The saturation flow rate for the permitted movement is calculated step-wise as:

$$GO(t)_p = \min \left\{ S_s \bullet \frac{v_l}{v_e}, MFR(t) \right\}$$

where,

GO(t)<sub>p</sub> = GO-pattern (vphg) of the shared permitted movement  
MFR(t) = maximum flow rate during time t per the permitted-movement model

For right-hand drive, it is assumed that through traffic does not interfere with left-turn traffic, but left-turn traffic does interfere with through traffic in the shared lane. This interference is modeled by reducing the capacity of the through movement, and should cause an increased degree of saturation for the through movement.

The GO-patterns for the shared lane links are determined as follows:

1. As long as the left-turn movement is protected (or unopposed), the GO-pattern for the entire lane group is equal to the user-coded saturation flow rate.
2. During any phase in which the left-turn is permitted, the initial maximum flow rate for the entire lane group is determined according to the GO<sub>u</sub> equation above.
3. Whenever the permitted-movement model is exercised, the left-turn movement will be modeled separately in that the GO-pattern is computed step-wise as the lesser of the values in the GO(t)<sub>p</sub> equation above.

#### Protected Shared Lanes Methodology

The new model for protected shared lanes is quite simple compared to the permitted shared lanes model. In this model, a step-wise volume-weighted average saturation flow rate is computed for the entire shared lane group. User-coded primary (thru) saturation flow rates should not reflect the proportion

of turns, and user-coded secondary (left-turn or right-turn) saturation flow rates are ignored. All turning movement volumes contribute an 85% weighting of the primary link saturation flow rate. The thru movement volume contributes a 100% weighting of its own saturation flow rate. As with the single-link, manual coding technique mentioned earlier, the adjusted saturation flow rate would be computed internally as  $(360 \cdot 1800 + 45 \cdot 1800 \cdot 0.85) / 405 = 1770$  vphg for the sample data. The adjusted saturation flow rate is applied to all vehicles in the shared lane group, regardless of their turning movement, on a step-wise basis.

The step-wise volume-weighted averaging of saturation flow rates allows the model to react to queue spillover and queue spillback. These disruptions potentially modify the GO-pattern of any shared lane link, thus affecting the overall lane group saturation flow rate. For example, queue spillback could disrupt an upstream shared right-turn movement, or queue spillover could disrupt an adjacent shared thru movement.

Although this TRANSYT treatment of protected shared lanes is consistent with the HCM recommendation of a 0.85 adjustment factor for protected right-turns, it is slightly inconsistent with the HCM recommendation of a 0.95 adjustment factor for protected left-turns. Although protected shared left-turns are less common than protected shared right-turns in right-hand drive countries, if the protected shared left-turn volume were heavy in a certain situation, a slight increase to the primary link saturation flow rate would be warranted. In left-hand drive countries, if the protected shared right-turn volume were heavy in a certain situation, a slight increase to the primary link saturation flow rate would be warranted.

This new model for protected shared lanes is applied on a step-wise basis, and is not available when link-wise simulation is used. This becomes yet another reason that step-wise simulation is recommended for most applications of TRANSYT. When link-wise simulation must be used, it may be preferable to also use the manual coding technique for lane groups represented by a single link, as described earlier. To accomplish this, one would use the Edit > Traffic screen instead of the Edit > Lanes screen. This coding technique would avoid the pitfall of applying the permitted shared lanes model to analyze a protected shared lane.

## Sign Control Methodology

Refer to the Sign Control section of the documentation for details on how to apply the unsignalized models in TRANSYT, and refer to the Sign Control Examples section for tutorial examples.

In TRANSYT, all vehicles are assumed to stop on permitted movement links. The model was derived analytically, and the assumption was made that there is no substantive functional difference between stop and yield control. The user can imply the difference between these by calibrating the maximum flow rates on permitted links.

Within the program, any permitted link at a sign controlled intersection will have its OUT-pattern calculated as follows:

$$OUT(i, t + d) = \min\{MFR[v(i, t + d)], IN(i, t + d)\}$$

where,

$OUT(i, t + d)$  = OUT-pattern of sign controlled link  $i$  at time step  $t + d$

$MFR[v(i, t + d)]$  = maximum flow rate for link  $i$  at time  $t + d$  based on opposing flow  $v$

$IN[v(i, t + d)]$  = IN-pattern for link  $i$  at time  $t + d$

$d$  = the minimum delay to sign controlled vehicles

$d$  is computed as the deceleration time from the cruise speed as:

$$d = \frac{u}{2a} + \frac{u}{2a'}$$

where,

u = speed, in feet or meters per second

a = acceleration rate (default = 5 fpss or 1.6 mpss)

a' = deceleration rate (default = 10 fpss or 3.3 mpss)

The above model has two aspects that warrant discussion. The first aspect of interest involves the default acceleration and deceleration rates. The default acceleration rate is obtained from Figure 1X-34 of the 1990 AASHTO Green Book [14], based on achieving a speed of 30 mph (48 km/hr) from a dead stop. The default deceleration rate is obtained from the calculation of change intervals suggested in the 1983 Traffic Control Devices Handbook [15]. Default acceleration and deceleration rates for buses are obtained from the corresponding rates indicated for trucks.

The second aspect of interest is that acceleration delay is considered in the delay from the arrival until the departure. Because this is delay associated with a stop at the current intersection, it is appropriate to include it "here" and not downstream. Since the stopped vehicles must project the acceptance of a gap to a point at which they are, in fact, already moving, it is not too much of a bias to include all the time before the OUT-pattern "reacts" when calculating delay at this intersection.

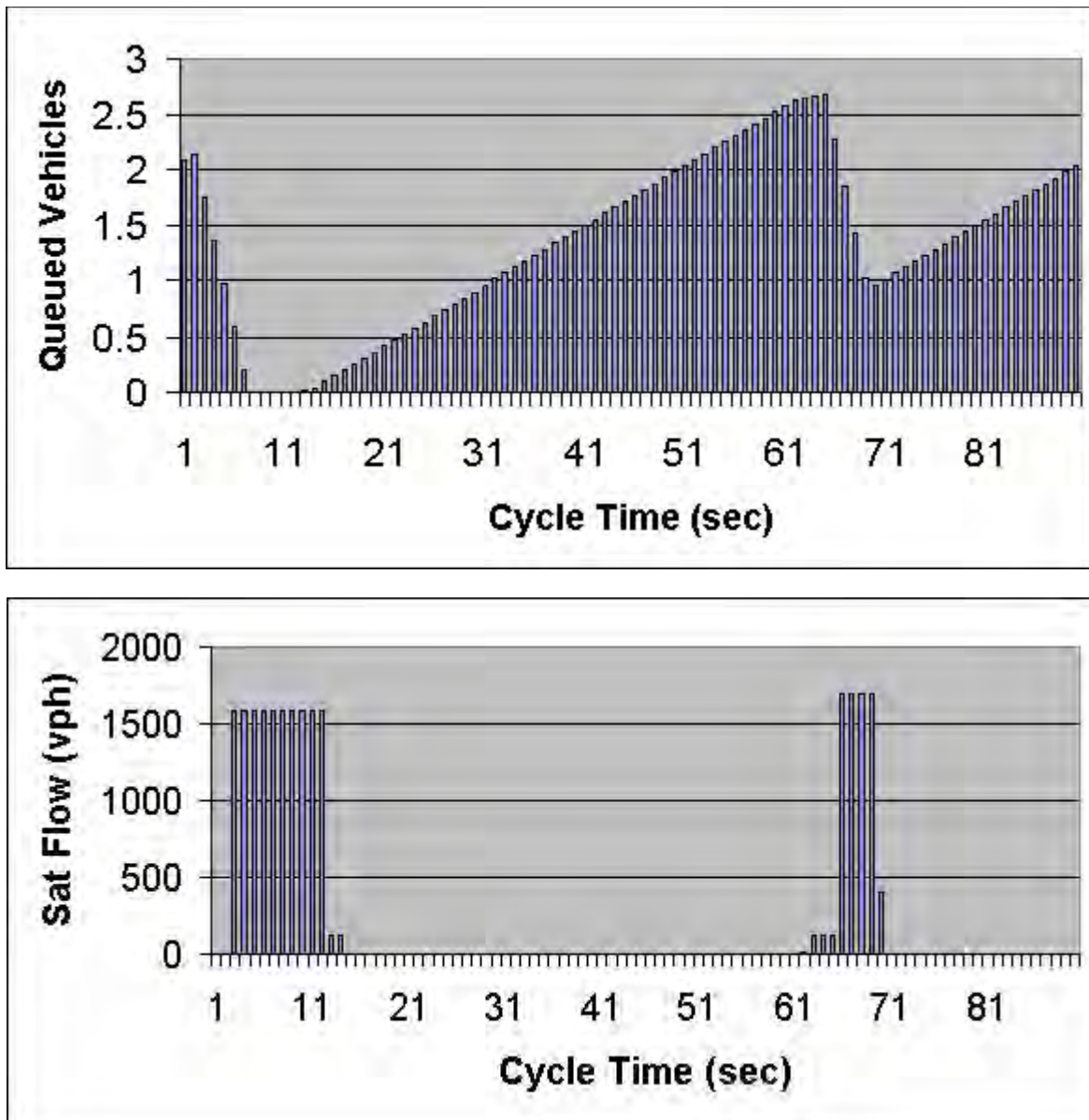
Note that error messages may occur during the run if sign-controlled links are coded on the Edit > Optional > Global > Shared Lanes screen (or record type 7). Refer to the Sign Control Examples section of the documentation for instructions on modeling permitted shared lanes at unsignalized intersections.

## Spyglass Internal Data Viewer

Spyglass is a utility program created to assist in preparing the TRANSYT internal array data for further analysis. After Spyglass processing, these data are easily viewed by using the Profile View screen. When "Run Spyglass" is selected from the TRANSYT-7F interface, a TRANSYT run is first performed, and Spyglass then generates several data-filled files for further analysis. Note that Spyglass outputs will currently reflect step-wise simulation, even if link-wise simulation is specified in the TRANSYT input file.

TRANSYT writes specific pieces of information into "Spyglass.out" for each link and for each step of simulation. This information can be extremely useful in understanding what occurred during the simulation run, although such extensive analysis may not be necessary for all projects. In order to analyze this information effectively, the data must be analyzed using spreadsheet software, or the Profile View screen. Prior to spreadsheet analysis, it is necessary to sort the information by link and by variable. Spyglass automatically parses the immense Spyglass.out file into several variable-specific text files that are more manageable. These sub-files must be further sorted by link number within the spreadsheet program. This is easily accomplished by sorting the link numbers alphabetically, or by using the A-Z button inside Microsoft Excel. Following this, the user can interpret the results directly, or create their own graphs and diagrams within the Profile View screen, or the spreadsheet program (as shown below).





Spyglass and the Profile View screen are crucial components of TRANSYT-7F, either as teaching tools, or to assist the practitioner in calibration or validation of results. Useful plots and graphics can be generated within seconds when manipulating the Spyglass data within spreadsheet software such as Microsoft Excel. Sample uses of Spyglass include:

- Plotting the flow profile (IN-pattern) to ascertain the level of progression, or to calibrate the platoon dispersion factor (PDF) aspect of the TRANSYT platoon dispersion submodel.
- Plotting the link-full flag of a downstream link with respect to the saturation flow rate profile (GO-pattern) of an upstream or adjacent link, to ascertain the nature of queue spillback or spillover effects. The out-pattern also reflects queue blockage effects.
- Plotting the queue profile, or queue accumulation polygon (QAP) to view progression effects, permitted turning movement effects, the movement of "sneakers", etc. (the area under this plot is used to compute uniform delay,  $d_1$ ).

- Plotting the saturation flow rate profile (GO-pattern) to ascertain effective green times, queue blockage effects, window of opportunity for sneakers, etc. (the area under this plot is used to compute capacity).
- Plotting the number of queued vehicles with respect to the back of queue length. Typically the maximum back of queue will continue to increase in the opening seconds of a green phase, at the same time that the number of queued vehicles begins to decrease. Once the "shockwave" reaches the back of queue, then the back of queue will begin to decrease as well.

The variable-specific text files generated by Spyglass are the following:

- BOQ.OUT: Maximum back of queue during each step of simulation
- GOPAT.OUT: Go-pattern (saturation flow rate) during each step of simulation
- INPAT.OUT: In-pattern (input flow rate) during each step of simulation
- LINKFULL.OUT: Link full flag during each step the simulation
- OUTPAT.OUT: Out-pattern (output flow rate) during each step of simulation
- QUEUE.OUT: Uniform queued vehicles during each step of simulation
- TRAVEL.OUT: Travel time from the upstream intersection to the back of queue location on the subject link

For a single-cycle simulation run with a simulation stepsize of 1-second, the number of output values written for each link should be double the cycle length, due to the pattern initialization cycle. If conditions are sufficiently stable, internal array information may be identical during the initialization and simulation cycles. If the internal array information differs between the initialization and simulation cycles, then the second half of data from the actual simulation cycle should be analyzed. Spyglass is also capable of generating data for multi-cycle and multi-period simulation runs.

Note that the Spyglass.out file may also be requested by coding a '2' in field 4 of record type 0 of the standard TRANSYT input (\*.TIN) file. However, Spyglass outputs are only produced by TRANSYT when step-wise simulation is requested by the input file. Spyglass outputs are not available for link-wise simulation. Spyglass requests via record type 0 are not recommended for optimization runs, because this will significantly increase the computer running times, and produce Spyglass.out files that are sometimes too large to be sorted or interpreted. Instead, use Spyglass to analyze the "punch" file that performs simulation-only of the optimal timing plan. Record type 0 is automatically inserted within the standard TRANSYT-7F input file during actuated estimation runs, so that the internal array information can be printed out and used in computing actuated phase times. The actuated estimation model uses queue information from QUEUE.OUT to assist in computing queue service times, and uses input flow rate information from INPAT.OUT to assist in computing green extension times. For simulation-only runs, it is acceptable to leave in record type 0 if the user wishes to view internal array information.

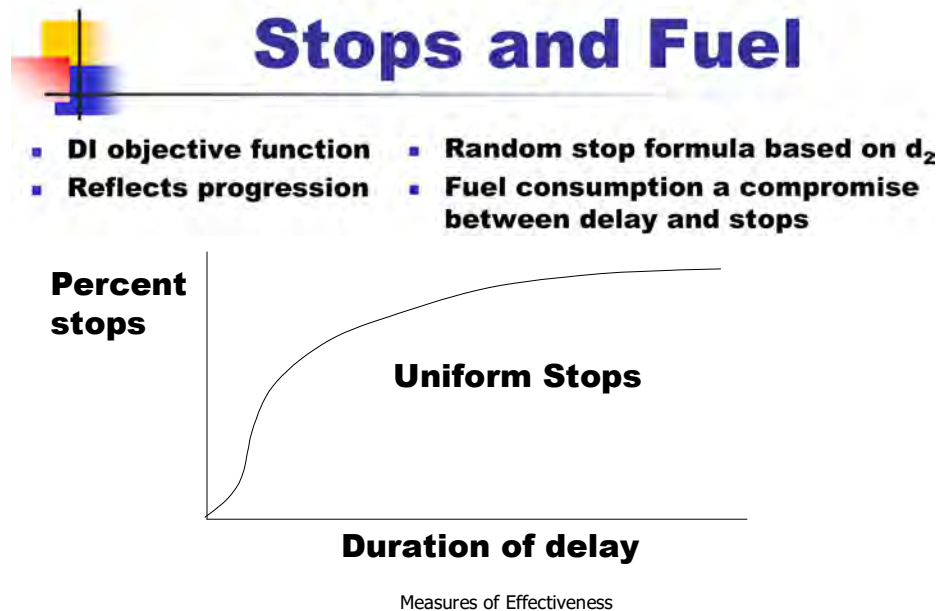
## Stops

TRANSYT assumes that vehicles which are delayed are also stopped. Anyone familiar with actual traffic flow knows this is not always the case, but the question is how to properly model "slowdowns" when the vehicles do not actually stop. The uniform delay model assumes that vehicles approaching the stop line or the back of queue stop instantaneously, then restart instantaneously.

### Uniform Stops

Once the back of queue begins to move, further arrivals are not fully stopped. Arriving vehicles would only experience a change in speed without complete stops. TRANSYT assumes that such vehicles partially stop (delay is still estimated as described previously) and calculates the effective stops by a filtering algorithm. Studies suggest that short periods of delay can be expressed as fractions of

stops for the vehicles affected. The number of stops can be deduced by the length of delay. Empirical studies by TRRL produced the relationship between percentage of stops and length of delay shown below. TRANSYT has a built-in facility to generate this reduction of stops function, or the user may code the data (see the upcoming "Model Calibration" section).



When using single-cycle simulation, TRANSYT does not accurately predict stops on near-saturated or oversaturated links. TRANSYT calculates the number of stops by multiplying the percentage of vehicles stopped by the number of vehicles that leave the stop line. When a link volume exceeds the link capacity, not all vehicles that arrive on the green can leave on the same cycle. Thus, the number of stops calculated based on the multiplication above does not take into consideration the extra vehicles that arrive but could not leave during the green. To account for this, the extra vehicles are assumed to experience "full stops". Thus, the total stops under saturated conditions are obtained by summing the number of stops as calculated from the multiplication above and the number of extra vehicles. Multi-cycle simulation explicitly considers stops from residual queue vehicles.

### Random Stops

TRANSYT calculates random-plus-oversaturation number of stops per hour in accordance with a model [16,33] from the Australian Road Research Board (ARRB):

$$h = \frac{3240N_o}{C}$$

where C is the cycle length in seconds and  $N_o$  is the random-plus-oversaturation queue. The calculation of random-plus-oversaturation queue is described in the documentation of Maximum Back of Queue.

As with second-term delay, the deterministic component of random-plus-oversaturation stops is subtracted from the calculations when multi-cycle simulation is requested. This component,  $h_d$ , is calculated as:

$$h_d = \frac{q \bullet N_d}{Capacity \text{ Per Cycle}}$$

where  $q$  is the arrival rate in vehicles per hour, and  $N_d$  is the deterministic component of the second-term queue. The calculation of  $N_d$  is also described in the documentation of Maximum Back of Queue.

Random-plus-oversaturation stops are essentially "queue move-ups" [39] following a phase failure, where there is not enough green time to serve the queue. As such, individual vehicles could presumably stop multiple times before clearing the intersection, although the queue move-ups would not be considered "full stops" for the purposes of computing fuel consumption. In order to be considered a full stop, a vehicle must transition between the cruise speed and zero speed. TRANSYT computes the fuel consumption due to random-plus-oversaturation stops differently than the fuel consumption due to uniform stops. The model assumes that random-plus-oversaturation stops are made while vehicles are traveling at the queue speed (the one at which the queue dissipates), while uniform stops are assumed to be made from cruise speed. The queue speed is calculated as follows:

$$Queue \text{ Speed} = \frac{Saturation \text{ Flow}}{Queue \text{ Density}}$$

Total stops percentages are sometimes reported as above 100%, and this would be a feasible result based on the above discussions of individual vehicles possibly stopping multiple times. The decision of whether to include random stops within the disutility index (DI) is subjective. By default, the DI includes random stops in addition to uniform stops, but the user can omit random stops from the DI by using the Edit > Optional > Global > Model Coefficients screen (or record type 8). However, note that random delay cannot be omitted from the DI.

#### Model Calibration

The TRANSYT stops reduction model can be calibrated for local conditions on the Edit > Optional > Global > Model Coefficients screen (or record type 8). Details regarding the values used by this model are provided in the table below. The stops reduction feature may be suppressed entirely by coding '4' as the model number and leaving the remainder of the screen blank; however, this is not recommended. Note that "Fld 2" in the table below (for record type 8) is associated with the first field of the Edit > Optional > Global > Model Coefficients screen. Also note that the table below references the default stops reduction figure earlier labeled "Stops and Fuel".

Fld	Variable	Description	Prec	Range	Default
2	'3'	<b>Stops Reduction Model</b>			
3	ISTOPL	the maximum delay associated with reduced stops	15	1-14	10
4-16	STPVAL(10)	stop reduction factors for each second of delay the default list is as follows (also see figure)	15	0-100	see list

Sec. of Delay	Percent Stop
1	20
2	50
3	65
4	76
5	83
6	88
7	93
8	95
9	97
10	99

## Throughput

When conditions are undersaturated, all vehicles arriving at the stop line are able to leave the stop line during each cycle. However, this is not the case when the system becomes saturated, either because the arrival rate is greater than the capacity of the link, or due to spillback from downstream links or adjacent turn bays. Throughput on a given link is defined as the integration of the number of vehicles leaving the stop line of that link during the simulation period. This is the area under the OUT-pattern, and is measured in vehicles. When the wide, 132-column legacy report format is selected, throughput is reported as "Flow".

## Time Full and Critical Link

When using step-wise simulation, the "link full" flag is turned on whenever the queuing capacity is exceeded. Time Full% represents the percentage of the cycle during which the link is full of vehicles. This percentage can be observed directly by plotting the Link Full Flag in the Profile View screen.

Critical link (Y/N) indicates that this is an oversaturated internal link, potentially causing queue blockage of upstream links.

Time full and critical link outputs are only available in the "wide" 132-column output format, where the narrow output and wide output formats can be requested on the Edit > Analysis screen (or record type 1).

## Time Jammed

TRANSYT reports a system-wide time jammed percentage. This measure of effectiveness provides an indication for the relative amount of "lost time" in the system due to "effective reds", i.e., vehicles cannot move during the green phase because the downstream link is full. This is a good overall indication of the level of queue spillback problems within the system. The percentage is obtained through summation of the jammed time experienced by each link, divided by the total available time in the system,

times 100. The total available time in the system is computed as follows:

$$TAT = T \bullet C \bullet N$$

where,

TAT = total available time in seconds

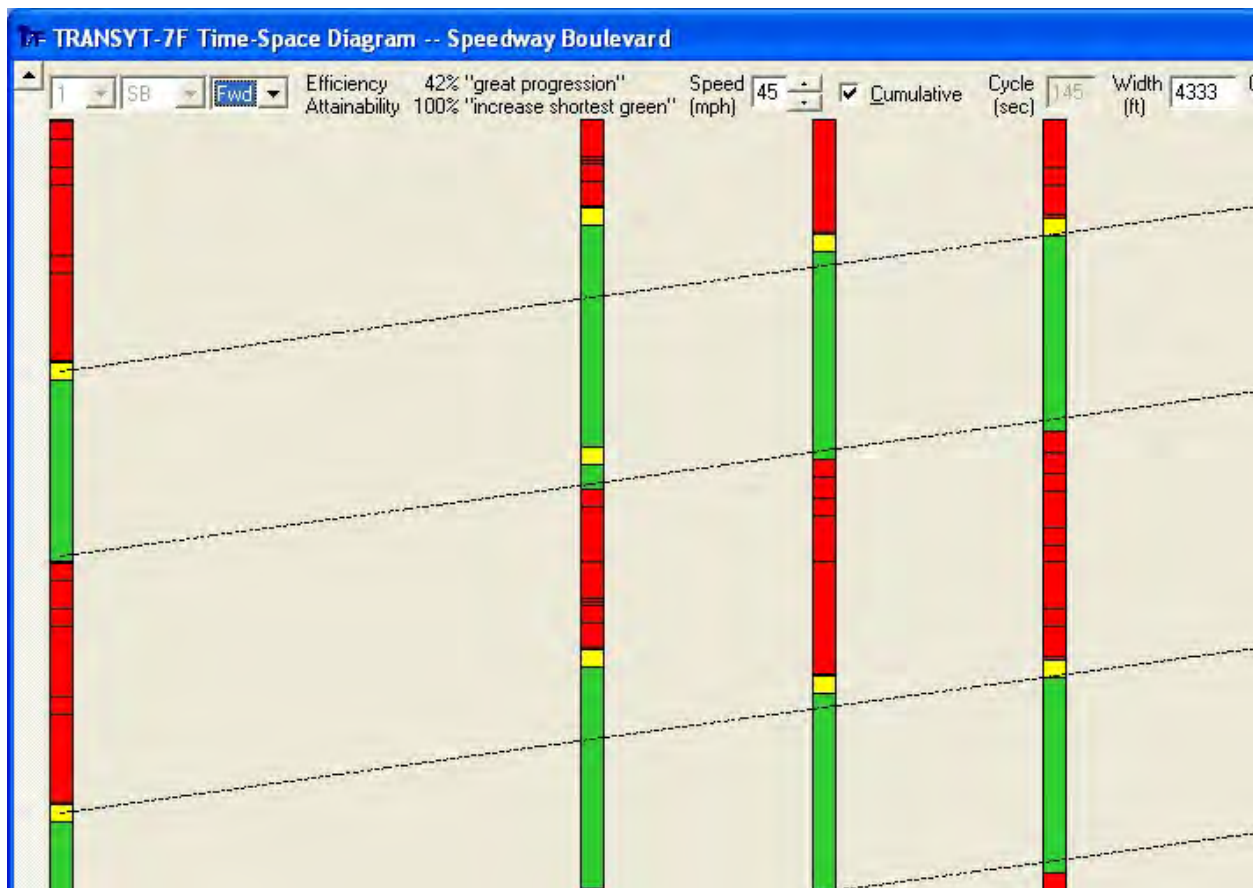
T = number of cycles (e.g., T = 1 for single-cycle simulation)

C = cycle length in seconds

N = total number of links in the system

## Time-Space Diagram

The Time-Space Diagram screen provides a mechanism for visualizing the available progression between intersections along the major street. The Time-Space Diagram screen is activated by clicking on the dark blue "arterials" icon (originally from HCS-Arterials) at the top of the TRANSYT-7F graphical user interface, or by going to View > Time-Space Diagram. The screen is disabled when the active TRANSYT-7F data file does not have any progression routes. The time-space diagram should reflect the timing plan of the active time period. The screen only displays progression for major routes that have been defined by the user. If there are multiple progression routes defined within the network, the user can switch between them at any time, by using the combo box in the upper left-hand corner of the screen. An example time-space diagram is displayed below. This screen can be sent to the printer, or to an image file, by pressing Ctrl-P. For a tutorial example that involves the use of this screen, refer to the Progression Analysis Examples section of the documentation.



When the Time-Space Diagram screen is first loaded, the first progression route within the data file is displayed. If the user switches to a different progression route, and then closes and reopens the screen, the most recently viewed progression route is displayed. The initial view is a "fit to screen" view, in which all intersections on the route should be visible on the screen at one time. The initial view also tries to display enough signal cycles to reflect arterial travel time. Assuming nonzero bandwidth efficiency, at least one green band should be undisturbed from left to right, with no vertical "wrapping" around the screen top and bottom.

### Timing Plan Representation

Signal timing plans are painted vertically, with only the major route green times and yellow times colored green and yellow. Green and yellow intervals for all minor movements are painted red, usually resulting in the appearance of numerous discrete and consecutive red intervals. To further emphasize the difference between intervals, beyond the simple use of color, the user can activate "cross hatching" by going to the Options > Preferences screen. Node numbers are listed in the white circles at the bottom of the Time-Space Diagram screen.

If a two-way stop controlled intersection is included within a progression route, the entire vertical distance will be painted green, so as not to interrupt the progression bands. If an uncoordinated intersection is included within a progression route, the entire vertical distance will be painted red, which will terminate any incoming progression bands. An intersection is always considered uncoordinated if it has been marked as uncoordinated on the Timing View screen. For purposes of drawing the time-space diagram, an intersection is also considered uncoordinated when user-coded green times do not sum to the system cycle length, and the node is not double-cycled. This differs from the TRANSYT simulation engine, which would attempt to automatically revise green times at any signal marked as coordinated (whose green times did not sum to the system cycle length).

Signal timing plans are fitted to the screen vertically, such that the vertical distances of each "bar diagram" represent the background cycle length of the progression route. Thus, the green time painted for each bar diagram represents the proportion of the cycle length occupied by the major street through movements. The vertical location of a green interval is a function of the offset and the reference interval. Scrollbars at the side of the screen can be used to simultaneously change vertical locations of all intervals and green bands, but have no net effect on the offsets and reference intervals. Regarding the link length (or intersection spacing) that governs the distance between bar diagrams, these are obtained from the Edit > Traffic screen (or record type 28). These link lengths can obviously be unequal between intersections, and can be changed at specific locations when necessary.

The Time-Space Diagram screen does not contain an initial timing model, so all interval times must be taken directly from the TRANSYT input (\*.TIN) file. If the user wishes to obtain a time-space diagram that reflects initial timing model calculations, or optimization calculations, the punch (\*.PUN) file should be loaded first before observing the diagram.

When progression bands are being drawn in one direction (forward or reverse) only, the Time-Space Diagram screen is capable of painting multiple green intervals for major route movements. For example, if the forward direction is Eastbound, and the eastbound through movement begins before the westbound through movement, the eastbound movement will have multiple forward progression green intervals. When progression bands are being drawn in both directions, only the coordinated phase (with both major route links moving simultaneously) is painted green.

### Interactive Coding of Offsets

Moving the mouse cursor inside any of the vertical bar diagrams will allow the user to click and drag timing plans vertically. This automatically modifies the offset within the TIN file. For multi-period analyses, offsets are modified for either a certain time period or for all time periods, depending on

whether Options > Time Period > Select All is activated. Moving the mouse cursor inside the bar diagrams of two-way stop intersections or uncoordinated intersections will have no effect.

By adjusting the offsets interactively, it might be possible to quickly achieve an excellent bandwidth. If this is difficult to accomplish manually, the PROS/DI objective function can be used to optimize bandwidth as well. When adjusting the offsets interactively, it is also helpful to take note of the efficiency and attainability values, as discussed later.

### Dynamic Diagram Updating

The Time-Space Diagram updates itself in response to data changes on other screens. For example, if the user specifies a new value within Options > Cruise Speed and then clicks on "OK", the time-space diagram should update itself instantly and accordingly. Similarly, if the user makes changes to the timing plan on the Edit > Timing screen, dynamic updating of the time-space diagram should occur. Changes to the start-up lost times and extension of effective green times on the Edit > Traffic screen would affect where the progression bands are drawn. Finally, if the user switches between time periods by using the Time Period Quick Jump (or Options > Time Period > Select), the time-space diagram should dynamically reflect the timing plan of the active time period. Certain data (cycle lengths, phasing sequences, start-up lost times, extension of effective green times) are always read from time period #1, but offsets and green times are obtained from the "active" time period.

### Forward Direction

The arterial forward direction (NB, SB, EB, or WB) is displayed in a disabled combo box, in the upper left-hand corner of the screen. This forward direction cannot be changed here, but can be changed in the Map View, or in the Edit > Optional > Route > Route Parameters screen.

The progression band direction (Fwd, Rev, or Both) is displayed in a second combo box. When the Time-Space Diagram screen is first loaded, progression bands are drawn in the "forward" direction of travel only. The progression bands are drawn as dashed lines across the screen, and their slope is a function of link length, and arterial cruise speed. The user can switch the progression band combo box to draw these lines in the reverse direction, or in both directions, if desired. For one-way progression routes, this combo is disabled, and only displays the forward direction.

### Efficiency and Attainability

When optimizing offsets interactively, the efficiency and attainability values at the top of the screen are updated automatically. By paying attention to comments such as "fair progression" and "increase shortest green", it can be possible to determine whether further improvements are needed. If an uncoordinated intersection is included within a progression route, the screen will state that "This artery contains uncoordinated nodes". Efficiency and attainability will always be zero for arteries that contain uncoordinated nodes, because it is not possible to achieve reliable progression through an uncoordinated intersection. For more details, refer to the documentation sections on efficiency and attainability.

As stated earlier, the Time-Space Diagram screen does not contain an initial timing model. Therefore, efficiency and attainability will reflect user-coded green times within the TRANSYT input (\*.TIN) file. If the user wishes to obtain an efficiency and attainability that reflect initial timing model calculations, or optimization calculations, the punch (\*.PUN) file should be loaded first before observing these values.

### Speed (mph or kmph)

By default, arterial cruise speeds (that govern the slope of the progression bands) are not obtained from the Edit > Feeders screen (record type 28). Instead, for the purposes of drawing the progression bands, arterial cruise speed is taken from the Options > Cruise Speed pop-up screen. The user can then change speeds directly on the Time-Space Diagram screen, and immediately observe the



effect.

In cases where the cruise speed does not remain constant along the progression route, an option exists to use the exact speeds that were coded in the data file. Refer to the Options > Preferences screen documentation in Chapter 3 for more details. When this option is used, speeds cannot be changed directly on the Time-Space Diagram screen, and only "cumulative" bandwidths can be displayed (as opposed to "constant" bands; see below).

### Cumulative Bands

If this checkbox is turned on, the initial progression bandwidth is based on the major street through movement's green time at the "first" intersection only. As new intersections are encountered, it is possible to observe where the bandwidth is becoming partitioned, and gradually reduced. If this checkbox is turned off, the constant artery-wide bandwidth(s) will be displayed.

At the very first intersection on the progression route, it should be possible to notice that progression band does not begin and end exactly when the major street green time begins and ends. Instead, the progression band begins a few seconds after the onset of green, and ends a few seconds after the onset of yellow. This is a reflection of the effective green time, as indicated by the user-coded start-up lost times and extension of effective green times.

### Width

As mentioned earlier, when the Time-Space Diagram screen is first loaded, the initial view is a "fit to screen" view. All intersections on the route should be horizontally visible on the screen at one time. The horizontal distance from one end of the progression route to the other is displayed in either feet or meters, depending on whether or not English units have been selected. This distance is displayed using the Width field. Although the width of the software screen remains constant, the displayed value (in units of feet or meters) indicates the "represented" real-world distance of the progression route. If desired, the user can modify the Width, to obtain a closer view between intersections.

### Origin

When the Time-Space Diagram screen is first loaded, the timing plan for the first intersection (coded on the progression route) is painted on the left-hand side of the screen. The remaining timing plans are painted further to the right, as a function of link lengths and intersection spacing. The default Origin value is zero, which corresponds with the first intersection being painted on the left-hand side of the screen. If desired, the user can code a non-zero Origin value, which will change the horizontal location of all intersections. The Origin value is displayed and coded in units of feet or meters, depending on whether English units have been selected.

### Number of Cycles

A drop-down combo box can be used to specify how many signal cycles will be displayed on the diagram. Note that displaying multiple cycles can cause the screen to operate more slowly, which could affect speed of the vertical scrollbars, or interactive offset adjustment.

### Input Coding Requirements

Progression routes are either coded graphically on the Map View or numerically on the Edit > Optional > Route screen (record types 42, 60, route title record, and 61). For example, when creating a brand-new file, if the user coded these routes on the Map View (like in the Progression Analysis Examples tutorial), the arterial links would automatically be defined on the Edit > Optional > Route > Arterial Routes screen (record type 42). If the arterial links are not coded on record type 42, or if the remaining required data on the other record types (60, route title record, and 61) are not available, the diagram cannot be displayed.

Route numbers should begin at '1', and should not skip any numbers. For example, if there are 4 progression routes in the network, they should be numbered 1, 2, 3, and 4. Unusual route numbers or route names can be documented by using the route title record.

The time-space diagram can only be displayed if there are non-zero green times coded within the input data file. If all green times within the data file are zero, and the initial timing flag is being used to supply green times automatically, it would be necessary to create and load the punch file containing non-zero green times.

In addition, link connection information for the diagram is obtained from the Edit > Feeders screen (or record type 28). If the link connection information has not been coded, the diagram cannot be displayed. If the link connection information has been partially coded, such that the link length between certain intersections is still zero, the diagram might display an "eclipse" situation where certain node numbers overlap each other.

## Total Operating Cost

The cost model was updated to 1997 in release 8. TRANSYT can explicitly optimize excess operating cost when requested by the user. The model is:

$$TOC_i = \left\{ \frac{K_{i1}TT_i + K_{i2}S_i + C_d d_i}{1000} + C_f f_i + OC_i \left( \frac{TT_i}{V_i} + d_i \right) \right\} \bullet I$$

where,

TOCi = total operating cost in dollars for link i

TTi = total travel on link i in vehicle miles (vehicle kilometers)

Si = total stops on link i in vehicles per hour

di = total delay on link i in vehicles per hour

fi = total fuel consumption on link i in gallons (liters)

O = average vehicle occupancy in passengers per vehicle

Vi = cruise speed on link i in miles per hour (kilometers per hour)

I = inflation rate, multiplied cumulatively from 1987

Cx = unit costs of vehicle delay (d) excluding passenger time, fuel consumption (f), and passenger time (t), respectively

Kij = model coefficients that are functions of speed on each link i:

$$K_{ij} = A_{j1} + A_{j2}V_i + A_{j3}V_i^2 + A_{j4}V_i^3$$

where Ajk are also regression coefficients, but different than the ones used in the fuel consumption model. For future updates to the inflation rate, see reference [21], which is carried at most public libraries.

### Model Calibration

The TRANSYT operating cost model can be calibrated for local conditions on the Edit > Optional > Global > Model Coefficients screen (or record type 8). Details regarding the coefficients used by this model are provided in the table below. Note that "Fld 2" in the table below (for record type 8) is associated with the first field of the Edit > Optional > Global > Model Coefficients screen. Also note that some of the operating cost coefficients can also be edited on the Edit > Optional > Global > Network Parameters screen (or record type 10).

The coding of this data must properly reflect the level of numerical precision ("Prec"). For example, in order to code AOCM(1,2) as 0.32508 based on E4 precision, it would be necessary to code '3251'.

FId	Variable	Description	Prec	Range	Default
<b>2</b>	<b>'2'</b>	<b>Operating Cost Model</b>			
3	AOCM(1,1)	$C(ost) = [(K_1T + K_2S + DC * D) / 1000$	E1	0-(-)9999	145.1
4	AOCM(1,2)	$1000 + FC*F + O*TC(T/V + D)] + I$	E4	0-(-)9999	0.32508
5	AOCM(1,3)	$K_1(cruise) = A_{11} + A_{12}V + A_{13}V^2 + A_{14}V^3$	E5	0-(-)9999	0.014065
6	AOCM(1,4)	$K_2(stops) = A_{21} + A_{22}V + A_{23}V^2 + A_{24}V^3$	E7	0-(-)9999	0.000192
7	AOCM(2,1)	where C = cost (dollars)	E4	0-(-)9999	-0.68567
8	AOCM(2,2)	I = inflation rate (RTINFL), or currency	E4	0-(-)9999	0.3465
9	AOCM(2,3)	conversion	E5	0-(-)9999	0.0374
10	AOCM(2,4)	T = total travel (veh-mi)	E7	0-(-)9999	0.00032
11	DC	DC = unit cost of veh. delay (\$/1000 veh-hr)	E2	0-99999	165.99
12	COSTGA	D = total delay (veh-mi)	E4	0-99999	1.25
13	TC	S = total stops (vph)	E4	0-99999	1.4916
14	RTINFL	FC = unit cost of fuel (COSTGA, \$/gal)	E4	0-99999	1.0
		F = fuel (gal)			
		O = average vehicle occupancy (persons per vehicle)			
		TC = unit cost passenger time (\$/pers-hr)			
		V = cruise speed (mph)			
		A <sub>ij</sub> = AFCM <sub>ij</sub> -- the model coefficients			

## Total Travel and Total Travel Time

The total travel in vehicle miles (or vehicle kilometers) will be constant for any given network and demand distribution. This is simply the aggregate of the product of link volumes and link lengths:

$$TT_i = v_i \bullet L_i$$

where,

TT<sub>i</sub> = total travel on link i in vehicle miles (vehicle kilometers) per hour

v<sub>i</sub> = traffic volume on link i in vehicles per hour

L<sub>i</sub> = length of link i in miles (kilometers)

In TRANSYT releases prior to release 8, the v<sub>i</sub> value used above was the user-coded traffic volume on link i. When conditions are congested, not all vehicles can leave the upstream stop line, due to queue blockage and/or heavy opposing volumes. In these cases, traffic volume actually using the link might be lower than the volume coded by the user. Thus, when step-wise simulation is requested in release 8 or higher, v<sub>i</sub> is determined based on the simulated traffic volume.

The total vehicle-miles of travel will be constant for a network with a given set of traffic volumes. It is simply the product of link traffic volume and link length.

Similar to total travel, this system-wide measure of effectiveness (total travel time) is the product of link volumes and total time spent on the links, including delay:

$$TTT_i = v_i \left( \frac{L_i}{V_i} + d_i \right)$$

where,

TTT<sub>i</sub> = total travel time in vehicle hours per hour on link i

v<sub>i</sub> = traffic volume in vehicles per hour on link i

L<sub>i</sub> = length in miles (kilometers) of link i

V<sub>i</sub> = average cruise speed in miles per hour (kilometers per hour) on link i

d<sub>i</sub> = total delay in vehicle hours per hour on link i

As with the calculation of total travel, the traffic volume on link i is the summation of the flow actually arriving on the link, when step-wise simulation is used. The total travel time in the network should decrease as the signal timing is improved to reduce delay.

The total travel time is an important measure of effectiveness in evaluating alternative control strategies, particularly if the coded speeds change between "before" and "after" scenarios. Total travel time includes both the time spent moving and the time spent delayed for all vehicles on each link.

Since the delay estimates do not account for changes in cruise speed on links per se, it is important to use total travel time to evaluate improvements that lead to cruise speed increases.

The average cruise speed used in the total travel time calculation above is computed using a volume-weighted average of the cruise speeds from multiple feeding links. If there are no feeding links coded, the external link cruise speed (coded on the Edit > Optional > Global screen or record type 10) is used. Link-specific external link speeds can optionally be coded on the Edit > Feeders screen (or record type 28).

## Uncoordinated Operations Methodology

Checkboxes to indicate any uncoordinated signals are located on the Edit > Timing (View) and Edit > Optional > Timing screens. A checkbox to indicate that the entire network is uncoordinated is located on the Edit > Analysis screen, but this should only be used if all signals are uncoordinated.

The lower cycle length from the Edit > Analysis screen (or record type 1) indicates the one and only network cycle length when performing simulation-only runs, or when performing optimization runs constrained to a single cycle length. Typically the network-wide cycle length is the same for every signalized intersection in the network; however, some signals may operate with a double cycle. If any signals are to be double cycled, all candidate cycle lengths to be evaluated (in a cycle length optimization run) must be even numbered.

Uncoordinated intersections are always simulated with uniform flow profiles, because there is no reason to expect platoons to consistently arrive during any given window within the signal cycle. However, if feeding links are also coded, it remains possible to simulate network-wide queue spillback, fuel consumption, travel time, etc.

If the "Uncoordinated Operations" checkbox from the Edit > Analysis screen is switched on, all signalized intersections in the system are simulated with uniform flow profiles. When uncoordinated operations are in effect, the user-specified signal timing plans on the Edit > Timing screen (or record types 1X and 18) indicate the local isolated cycle lengths. To simulate some intersections as coordinated

and some intersections uncoordinated, it is best to specify coordinated operations for the network, and then indicate the uncoordinated intersections on the Timing screen.

Uncoordinated operations can also be optimized if desired. Refer to the Uncoordinated Optimization section for more details.



# Chapter 5: TRANSYT-7F Optimization Process

## Initial Timing Model

If requested, TRANSYT-7F is capable of calculating phase lengths that may serve as initial splits prior to optimization. This might be necessary, for example, if one or more brand-new signals are being installed in the field, and there is really no "existing" timing plan for these new signals. For a re-timing project, using the initial timing model is optional. Since immediate knowledge of movement capacity and degree of saturation is needed, the initial timing model cannot be used in conjunction with direct CORSIM optimization.

The initial timing plan can potentially affect the results (i.e., the recommended signal timing plan design) of optimization. Generally speaking, the closer the initial timing plan is to the global optimum solution, the higher the probability is that the global optimum solution will be reached. When it comes to locating the global optimum solution, the genetic algorithm is more effective than the hill-climb method is, as described later. However, poor initial timing plans may drastically increase the amount of computer time required, or may prevent the program from being able to locate the global optimum solution at all.

So for a re-timing project, the question becomes, can the initial timing model generate a timing plan that is closer to the global optimum solution than the existing timing plan? This can be difficult to guess correctly. If enough time is available for the project, it might be preferable to try both the initial timing model and the existing timing plan as starting points, to see which one works better. But note that for actuated signals, the initial timing model is particularly effective at setting actuated phases to operate in the 80%-90% degree of saturation range, which tends to facilitate network-wide optimization.

The initial timing model is not a substitute for a standard optimization run because the initial timing model does not have access to TRANSYT simulation model results. All initial timing computations must be performed prior to simulation and optimization. For example, although the initial timing model tries to equalize the degrees of saturation for all critical movements at a pre-timed signal, these same degrees of saturation could potentially be impacted by queue spillback. Only by performing simulation could the full impact of queue spillback be realized. Further, permitted movement operations and platoon progression have an impact on what the optimum signal timing plan must be, but network operations like these can only be accurately measured by simulation. Therefore the initial timing model, which is more of a quick-and-dirty design procedure than an actual model, is only intended to provide a reasonable and effective starting point prior to optimization.

### Pre-Timed Phase Lengths

The TRANSYT initial timing procedure for pre-timed signals is somewhat similar to the green time estimation procedure described in Highway Capacity Manual Chapter 10, Appendix A [12]. Initial phase lengths are calculated by equalizing the degrees of saturation on conflicting critical movements. The degree of saturation is defined as:

$$X_i = \frac{v_i \bullet C}{s_i \bullet g_i}$$

where,

$X_i$  = degree of saturation on link  $i$   
 $v_i$  = total volume coded for link  $i$  in vehicles per hour  
 $C$  = system cycle length in seconds  
 $s_i$  = saturation flow rate of link  $i$  in vphg  
 $g_i$  = effective green for link  $i$

Following this computation, the degree of saturation on each conflicting critical movement may not always be exactly equal due to: 1) minimum phase length constraints, or 2) round off. These same computations are also performed for isolated, fully-actuated signals, at which every phase is actuated. For coordinated, semi-actuated signals, at which one phase is non-actuated, a unique procedure is used as described below.

#### Traffic-Actuated Phase Lengths

The original methodology for actuated phase time design is referred to as the "target degree of saturation" model, and its mathematics are somewhat similar to the initial timing model for pre-timed signals. The original target degree of saturation model, employed by TRANSYT for many years, is primarily effective for optimization runs. This model facilitates location of the optimal actuated phase times. For more information on applying this model, refer to the Traffic-Actuated Control section of the documentation.

The original methodology, added with release 6, was originally developed by the Institute for Transportation Studies (ITS), at the University of California-Berkeley (UCB). The initial software code was also accomplished by ITS and was incorporated into TRANSYT by the University of Florida Transportation Research Center. Some background on the subject and UCB's methodology [19] are given below.

Traffic-actuated signals in a coordinated system operate on a common background cycle length. Synchronization is provided through the yield point, which is a fixed point in the cycle, normally during or at the end of the non-actuated (also called synchronization, or sync) phase. The non-actuated phase has some minimum green time, plus all slack time not used by the actuated phases. The green times on the actuated phases, however, vary on each cycle between a minimum green and a maximum green time, depending on the arrival rate of vehicles and the value of the extension interval. Fixed force-off points in the background cycle are used to terminate the duration of the actuated phases. If an actuated phase terminates early, then any green time not used by the actuated phase is transferred to the non-actuated phase, as noted above.

The target degree of saturation method is designed to optimize the average green times for actuated controllers. The procedure is summarized in the steps below:

1. Effective green times are computed so that the degree of saturation on the actuated phases does not exceed the user-specified value percent. The average effective green time ( $g_i$ ) for an actuated phase can be calculated as follows:



$$g_i = \frac{v_i \bullet (C - L)}{s_i \bullet X}$$

where,

$v_i$  = volume on the critical movement for actuated phase  $i$  (vehicles per hour)

$s_i$  = saturation flow rate on the critical movement for actuated phase  $i$  (vphg)

$L$  = total lost time for the cycle (seconds)

$C$  = system cycle length (seconds)

$X$  = degree of saturation on the actuated phases (percent/100)

2. The effective green time,  $g'$ , for the non-actuated phase is estimated as:

$$g' = (C - L) - \sum_{i=1}^n g_i$$

where  $n$  is the number of actuated phases.

3. Next, the non-actuated phase is checked for oversaturation:

$$\sum_{i=1}^n g_i \leq (C - L) \left( 1 - \frac{v_m}{0.90 s_m} \right)$$

where,

$v_m$  = volume on the critical movement for the non-actuated phase (vehicles per hour)

$s_m$  = saturation flow rate on the critical movement for the non-actuated phase (vphg)

If the constraint above is not satisfied, then the effective green times on the actuated phases are reduced in proportion to their traffic volumes. These computed effective green times cannot be lower than the coded minimum phase lengths, and their sum should be equal to the system cycle length, minus the lost time.

In the case of more than one non-actuated phase (e.g., intersecting through phases in grid networks) the calculations are more involved. First, an initial estimate of the green times is made based on equalizing the degrees of saturation. Next, average green times for the actuated phases are calculated according to step 1 above, and adjusted as appropriate for minimum greens. The green time for each non-actuated phase is then computed considering the extra time from the preceding actuated phases, as specified in the phasing sequence.

If all phases are coded as actuated (i.e., isolated, fully-actuated operation), then the program reverts back to the model for pre-timed phase lengths, and attempts to equalize the degrees of saturation on conflicting critical movements.

TRANSYT-7F now offers a detailed new methodology for predicting traffic-actuated phase times, where queue service times and green extension times can be obtained from simulation. The actuated estimation model is effective for simulation runs, and evaluation of existing conditions. The actuated estimation model computes traffic-actuated phase times that are likely to materialize in the field, based on the user-specified maximum green and force-off settings. For more information on applying the actuated estimation model, refer to the Traffic-Actuated Control section of the documentation.

Initial Signal Timing Flag (associated with the Edit > Analysis screen, or record type 1)

If the initial timing flag is not coded, the user must provide offset (or yield point) values, as well as the durations of all fixed and variable intervals on the Edit > Timing screen (or record types 1X and 18).

Initial timing flag '1' indicates that the user will provide the durations of all fixed intervals only. Variable green interval durations will be provided by the program according to the initial timing models described earlier.

Note that "variable" intervals are green intervals that can be optimized. On the other hand, "fixed" intervals are not subject to optimization. Fixed intervals include yellow, all-red, and occasionally green (e.g., flashing don't walk) intervals. Refer to the Timing Screen documentation for additional details.

When using actuated control, initial timing flag '1' assigns green time to the actuated phases to achieve a target degree of saturation. All unused green time is donated to the major street non-actuated phase. The target degree of saturation model is effective for optimization runs, and facilitates location of the optimal actuated phase times. To ensure that the optimal actuated phase times materialize in the field, it is necessary to set the force-offs and maximum greens appropriately. Refer to the Traffic-Actuated Control section of the documentation for more details.

When using the actuated estimation model, initial timing flag '1' will cause force-offs or maximum greens to be estimated prior to the analysis. This is not necessarily recommended. The actuated estimation model is most appropriate for simulation runs, and evaluation of existing force-offs and maximum greens. The actuated estimation model computes traffic-actuated phase times that are likely to materialize in the field, based on the user-specified maximum green and force-off settings.

Initial timing flag '2' is the same as '1', except the user-specified offsets and yield points are retained. When initial timing flag 1 is used, the offset (or yield point) of each controller is set equal to the interval #1 offset at the master controller. For example, in the tables below, the interval #1 offset is 48 seconds at the master controller. The second table (after applying the initial timing model) illustrates that the re-designed offsets at each signal, although they have various values and reference intervals, are all equivalent to an interval #1 offset of 48 seconds.

master controller = node #6  
cycle length = 120 seconds

	interval durations										offset	reference interval	intvl #1 offset
node 6	0	5	0	5	0	6					105	4	48
node 3	0	6	0	6	0	6					57	4	89
node 9	0	5	0	5	0	6					57	4	91
node 12	0	6	0	1	0	5	0	6	0	6	3	4	83
node 15	0	6	0	7	0	7	0	6	0	6	6	8	39

	interval durations										offset	reference interval	intvl #1 offset
node 6	16	5	36	5	52	6					105	4	48
node 3	31	6	51	6	20	6					16	4	48
node 9	7	5	74	5	23	6					14	4	48
node 12	6	6	28	1	28	5	16	6	18	6	88	4	48
node 15	11	6	22	7	5	7	29	6	21	6	15	8	48

For optimization runs, initial timing flag '1' eliminates the need to specify offset or yield point values and variable green interval durations at signals to be optimized. This information must be provided, however, for signals that will not be optimized (i.e., nodes not listed on the optimization node list, or "grouped" nodes at which splits are fixed). The initial timing model (option '1') is automatically applied by the program whenever cycle length optimization is performed (except in the case of simultaneous genetic algorithm optimization of cycle length and splits). The initial timing model (option

'2') is automatically applied by the program whenever phasing sequence optimization is performed without cycle length optimization (except in the case of simultaneous genetic algorithm optimization of phasing sequence and splits).

When a multi-period simulation file is loaded, the initial timing flags can be coded uniquely among different time periods by using the record type editor. Although the initial timing flag can be used for multi-period simulation runs, it cannot be used for multi-period optimization runs. Thus, if multi-period cycle length or phasing sequence optimization is requested, optimization of splits becomes mandatory. Also note that the initial timing flag cannot be used to generate initial timings for uncoordinated signals.

## Optimization Node List

This list specifies the node numbers for which signal timing is to be optimized during an optimization run. By default, all node numbers are included on this list for a brand-new input data file. However, it is sometimes very helpful to remove all non-critical intersections (i.e., those already operating at level of service A or B) from the node list, prior to an optimization run. Including non-critical intersections on the optimization node list can sometimes drastically increase the amount of time required for the program to locate the global optimum solution, or can sometimes prevent the program from being able to locate the global optimum solution.

In general, small input files with 20 nodes or fewer are recommended, because this allows for better optimization. If it is necessary to simulate a large network, the user should remove as many intersections as possible from the optimization node list, so the program can concentrate on those critical locations.

When hill-climb optimization is used, the node list also specifies the *sequence* in which signals will be optimized. Nodes may be repeated on the node list to force them to be optimized more than once. This may reduce the chance of particular nodes being trapped in a local optimum solution.

In a cycle length optimization run, all nodes should be included on the optimization node list. In addition, the initial timing model can only be applied to intersections that are present on the optimization node list.

The optimization node list can be edited on the Edit > Optional > Global > Optimization Node List screen (or record type 2).

## Hill-Climb Optimization

The traditional optimization technique used in TRANSYT is referred to as a "hill-climbing" technique. This is an iterative, gradient search technique that requires numerous simulation runs by the computer. TRANSYT-7F also offers a genetic algorithm search technique, described elsewhere in the documentation. The hill-climb optimization procedure is briefly described in this section.

The optimization search technique differs from the optimization objective function. Both the search technique and the objective function can affect the final result. In general, the search technique governs which timing plans will be examined as candidates to be the optimal solution. On the other hand, the objective function governs the judgment of whether any given timing plan is effective or not.

Hill-climb and genetic algorithm optimization are examples of search techniques, whereas the disutility index (DI) and progression opportunities (PROS) are examples of objective functions. In TRANSYT terminology, an objective function is often referred to as the performance index (PI).

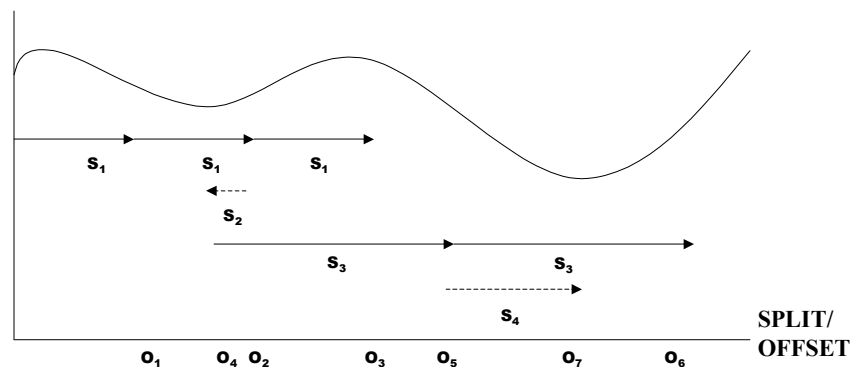
## Hill-Climb Optimization of Offsets and Yield Points

The steps below are followed for offset optimization, assuming the PI involves minimizing the DI:

1. An initial signal timing plan is simulated by the traffic model and the initial PI is calculated.
2. At signal "i" (initially the first signal on the optimization node list coded by the user), the offset is increased by an amount specified on the optimization "stepsize" list explained below. The resulting traffic flows are re-simulated on downstream links, subject to the sensitivity parameter threshold values, and a new PI is calculated.
3. The new PI is compared with the previous value (i.e., before the last signal timing change) as follows:
  - a. If the new PI is less than the previous value, the program continues to increase the offset by the same amount, as long as the PI continues to decrease. When the PI again increases, the program goes to step 4.
  - b. If the new PI is greater than the previous value, the program will decrease the offset by the same amount and continue to decrease the offset by this amount as long as the PI continues to decrease. When the PI again increases, continue with step 4.
4. When no further improvement can be made by varying the offset at this intersection, the process goes to the next intersection and begins at step 2 with the same optimization stepsize. Changes in offset are examined at each controller in the network in turn, for the same optimization stepsize.
5. Steps 2 through 4 are repeated for all of the optimization stepsizes specified.

The concept of optimization "stepsize" is best explained by use of an example. Suppose that the goal is to optimize the offset of one signal relative to a second signal, which has its timing fixed (in other words, this is a two-signal system). If the system was simulated for every integer offset between 1 and the cycle length, the resulting values of the PI might appear as shown by the curve in the figure below.

**OBJECTIVE FUNCTION**



Clearly, it would be extremely time-consuming to evaluate every one of these potential offsets to

find the absolute minimum value of the PI, particularly for a system with many intersections. For example, a system of only six signals would require over 46 billion trials for a 60-second cycle. To overcome this problem, Robertson [3] described a special application of the hill-climb concept using different stepsizes. The stepsizes in the figure above are S1, S2, S3, and S4. This is the list of optimization stepsizes. For the first step of the hill-climb process, S1, the change in offset is a medium-sized step of about 15% of the cycle. This change is from the initial offset of zero to O1. The PI is clearly lower, so O1 is "better". Add S1 again, which has the same result, so O2 is the new offset. Adding S1 again results in an offset of O3, but here the PI increases, so the best offset thus far is O2.

Let the next offset stepsize, S2, be small, say one second. When S2 is added to O2, the PI increases, so the direction is reversed. Clearly, the offset will continue to "improve" until the valley of the curve at O4 is eventually found. This is the best solution thus far, but it is clearly not the global optimal solution. This solution is referred to as a "local optimum solution".

Next, use a large offset stepsize, or about 40% of the cycle, S3. The PI at the resulting offset, O5, is lower than before, so the search has escaped from the local minimum. Adding S3 would get the search back into the first peak at O6, so O5 is retained.

Finally, repeat the series of small stepsize searches using S4, and the best solution will eventually be found at O7. Although the global optimum solution was found in this example, there is no guarantee that the global optimum will always be found.

Note that hill-climb optimization of offsets-only cannot be requested on the Edit > Analysis screen. This is strictly a limitation of the interface, since the model still supports this type of optimization. To get around this interface limitation, the user can either manually code record types 4 and 50 (using the record type editor) as described in the upcoming Hill-Climb Optimization Stepsizes section, or deactivate the "Allow Split Optimization" checkbox (on the Edit > Timing screen) at each node.

### Hill-Climb Optimization of Splits

The phase length (split) optimization process is similar to the offset optimization process described earlier, except that changes in the phase length at each signal are examined. One or more step sizes in the list of optimization step sizes may be designated for phase length optimization.

Be careful about optimizing splits at any traffic-actuated signal. The optimal splits may not occur in the field unless the maximum green times and force-offs are set appropriately. Refer to the Traffic-Actuated Control section of the documentation for more details.

Optimization changes to the phase lengths are subject to:

1. The sum of the phase lengths must equal the cycle length.
2. No phase length may be less than the user-specified minimum phase time.

In release 7, an improvement was added to the split optimization routine. TRANSYT typically optimizes splits by shifting the phase-change points of each phase sequentially. In some cases, this might lead the optimization process to be "locked in" on a localized minimum. For example, assume for the following sequence, the corresponding phase times, minimum phase lengths and v/c ratios are:

Phase Number	1	2	3	4
Phase Time	35	10	25	10
Minimum	15	10	15	10
v/c Ratio	35	55	95	55

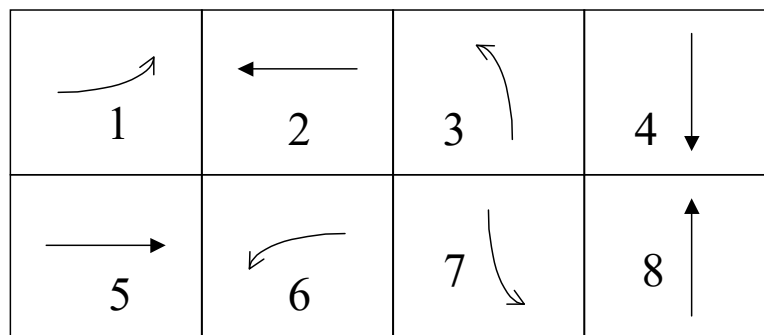
The traditional TRANSYT split optimization algorithm may fail to shift time from the first phase to

the third phase, even if the shift improves the performance. The sequential shifts of time do not work in this case because the phases adjacent to the third phase are at their minima.

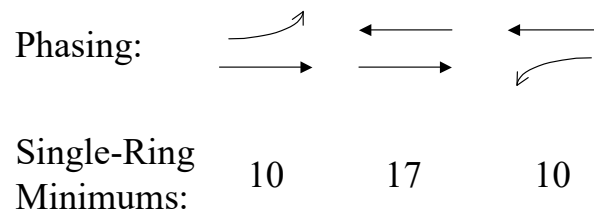
Since release 7, additional split adjustments are performed after the first split optimization step size to transfer time from the phase with the lowest degree of saturation to the one with the highest. In many cases, this additional optimization step allows the program to do a better job of allocating green time.

#### Split Optimization with Overlap Phasing

TRANSYT's traditional split optimization (as described above) required treating all phases as if they were equivalent pre-timed. Generally, this was acceptable, but in the case of dual ring control, particularly with lead-lag phasing, it was overly restrictive. Consider, for example, the case of a simple NEMA-type configuration of lead-lag phasing as illustrated below.



Suppose the minimum desired phase (or more properly, movement) duration (including change period) for left-turns is 10 seconds, the leading through is 25 seconds, and the lagging through is 27 seconds. The phase subsequence east-west and phase minimums for typical pre-timed or single-ring operation might be as follows.



The problem with this is the (middle) overlap phase might operate better with a shorter phase. Beginning with release 7, TRANSYT-7F recognizes the minimum phase duration coded as a variable, rather than as a constant. The value that should be coded for the overlap minimum phase time is '-xxyy', where 'xx' and 'yy' represent the minimum greens of the critical movements of the leading and lagging phase, respectively. (This coding technique should be avoided when running genetic algorithm optimization.)

The program calculates the "effective minimum phase durations" during the split assignment as follows (assuming the critical movements are the through movements):

$$MIN'_{ol} = \max \{ (MIN_{ld} - G_{ld}), (MIN_{lag} - G_{lag}) \}$$

where,

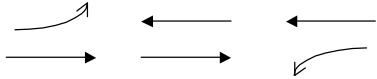
MIN'ol = the calculated effective minimum value for the overlap phase

MINi = the minimum green of the critical movements of the leading (ld) and lagging (lag) phases

Gi = the current optimized phase lengths for the leading (ld) and lagging (lag) phases

(MINi - Gi) = the candidate effective minimum phase durations due to the time remaining from the leading (ld) and lagging (lag) phases

The coding and resulting optimized splits based on the above example might be as follows:

Phasing:	
Coded Minimums:	10      -2527      10
Optimized Splits:	20      9      18

Notice that the combination of the overlap and lagging phases were (possibly) constrained by the minimum of 27 seconds for the westbound through movement. TRANSYT requires the minimum phase duration to be the sum of the fixed intervals, plus one second for the variable interval. Thus, assuming a change period of four seconds, the absolute minimum phase time that could be coded for a given phase appears to be five seconds. However, the optimal overlap time can sometimes be very short.

TRANSYT can handle coded overlaps as short as two seconds, and can simulate 1-second overlaps if the start-up lost times and extension of effective green times are coded correctly. Sample input data file "overlap.tin" from the TRANSYT-7F Example folder illustrates this coding technique. Typically the start-up lost times and extension of effective green times must be calibrated for the affected links, and the overlap green and yellow times are both coded as 1 second. For an overlap phase, it is acceptable to code a yellow time of 1 second, even if the duration of yellow displayed in the field is longer than 1 second. Double checking the effective green times for each link (in the wide, 132-column format output file) is recommended.

### Bandwidth Optimization

Two special optimization procedures that combine several features of the so-called "maximal bandwidth" and "minimal disutility" approaches to signal timing design have been incorporated into TRANSYT. The first procedure involves the simultaneous use of PROS and DI within the objective function. The second procedure requires use of other models or methods that provide a typical "time-space" solution, prior to using TRANSYT.

Hill-climb optimization of offsets and phase durations may be constrained so that an initial

bandwidth solution, with a given through bandwidth in each direction on each intersecting street (limited to four, two-way approaches), cannot be violated. For example, if a bandwidth program such as PASSER II has been used to obtain offsets which maximize the perceived through progression, the band "coordinates" may be entered into TRANSYT and the optimization will try to improve network operations without decreasing the prescribed through bandwidths. The so-called "bandwidth constraint" algorithm was developed by FHWA [11]. Because explicit band coordinates (i.e., the "offsets" to the beginning and end of the bands, referenced to the system-wide time reference base) must be entered for this procedure, it may only be used in a single-cycle optimization run.

The second procedure is described as follows:

1. Code the requisite traffic data for both TRANSYT and the separate bandwidth optimization model.
2. Perform the bandwidth optimization, including phasing sequence (if feasible), cycle length and offsets to obtain a best bandwidth solution. If multiple arteries are involved, this may require a series of runs to determine the optimal background cycle length and timing parameters.
3. Code the timing parameters into TRANSYT, including the band coordinates, then optimize the network to allow adjustments of all offsets and phase durations at all intersections; however, if the bandwidth constraint is imposed, no offset or split adjustment will be allowed to interfere with the coded through bands.

Note that the band coordinates can be coded in the Edit > Optional > Route > Bandwidth Constraints screen (or record type 41).

#### Hill-Climb Optimization Stepsizes

Previous versions of TRANSYT-7F have allowed the user to customize the hill-climb optimization stepsizes. This can still be accomplished in the current release. However this is no longer recommended or fully supported, because the default optimization stepsizes used by the program are effective for most practical applications, and because genetic algorithm optimization is now also available for better performance. The TRANSYT-7F graphical user interface (GUI) is not capable of generating the necessary inputs, so it is necessary to use the record type editor.

Hill-climb optimization stepsizes are coded on record type 4. If record type 4 is coded, it must be used in conjunction with record type 50 or 54 as described below.

When coding the optimization stepsizes, a value of '2' means one simulation step, while '1' means one second. Values greater than '2' represent a percentage of the cycle length. The maximum stepsize must not exceed 50% of the cycle length. Positive numbers within the list mean that the offset is to be varied from node to node. Negative numbers indicate that phase lengths (splits) are to be varied during this optimization step. If '-3' is coded, this tells the program to perform split optimization based on shifting time from phases with low degrees of saturation to phases with high degrees of saturation, in addition to the traditional shifting of time between adjacent phases.

When record type 4 is not used, the program may apply one of two default stepsize lists, depending on which option of record type 5X is controlling the optimization run. The default optimization stepsize lists are as follows:

Record type 52 or -52 ("normal")	15	40	-3	15	40	2	-2	2
Record type -53, -54, 53 or 54 ("quick")	15	40	15	2	-2	2		

Occasionally the user may wish to optimize offsets only, or splits only. To do this, it is necessary to code record type 5X appropriately, and code record type 4 as follows:



Offsets only	4	40	15	2
Splits only	4	-2		

Experience with TRANSYT has shown that the default sequence provided for a "normal" optimization run will normally produce an acceptable signal timing plan. "Quick" optimizations runs should usually be made for preliminary runs (such as analyzing different phasing sequences or cycle lengths). Final design runs should always be made with the "normal" optimization default list. Because the default stepsize list for a "quick" optimization run is shorter than the default stepsize list for a "normal" optimization run, less computer time will be required to complete the hill-climb process.

### Sensitivity Parameters

Previous versions of TRANSYT-7F have allowed the user to customize the hill-climb optimization sensitivity parameters. This can still be accomplished in the current release. However this is no longer recommended or fully supported, because the default optimization sensitivity parameters used by the program are effective for most practical applications, and because genetic algorithm optimization is now also available for better performance. The TRANSYT-7F GUI is not capable of generating the necessary inputs, so it is necessary to use the record type editor.

Sensitivity parameters, and the process described in this section, are only applicable to hill-climb optimization using link-wise simulation, and are not applicable to step-wise simulation, in the current version of the program. Sensitivity parameters may be coded on record type 6.

The sensitivity parameter record may be used to code a list of parameters to reduce the run time of the program. These parameters limit the number of downstream links for which flow profiles will be recalculated, after a signal timing adjustment to upstream node is made during the hill-climb optimization process. If record type 6 is coded, it must be used in conjunction with record types 4 and 50, or 54.

On fields 2 through 16 of record type 6, the user can code sensitivity parameters against which flow profiles are compared in order to determine if recalculation of flow profiles is warranted. Each entry on this record is associated with the entry and corresponding field of record type 4. Thus, the number of entries on this record should be the same as the number of entries on record type 4, otherwise a warning message will be printed, and the list may be either truncated or extended as appropriate.

As a general recommendation, larger sensitivity parameter values should be used for early hill-climb optimization stepsizes, and smaller values should be used as the hill-climb process converges toward the final solution. When record type 6 is not used, the program may apply one of two default lists, depending on what option of record type 5X is controlling the optimization run. The default lists applied by the program (i.e., by the use of record type 52, 53 or 54) are shown below for both "normal" and "quick" optimization runs, respectively, as well as the offsets-only and splits-only hill-climb stepsizes shown earlier:

#### 1. "Normal" optimization run -- record type 52:

Stepsizes:	15	40	-3	15	40	2	-2	2	
Parameters(%):		10	10	1	1	1	0.1	0.01	0.01
Values coded:	1000	1000	100	100	100	10	1	1	

#### 2. "Quick" optimization run -- record type 53 or 54:

Stepsizes:	15	40	15	2	-2	2	
Parameters(%):		20	20	10	1	0.1	0.1
Values coded:	2000	2000	1000	100	10	10	

#### 3. Offsets-only -- record type 50:

Stepsizes:	15	40	15	2	
Parameters(%):		10	1	0.1	0.01
Values coded:	1000	100	10	1	

4. Splits-only -- record type 50:

Stepsizes:	-2
Parameters(%):	0.01
Values coded:	1

## Cycle Length Optimization

Over the years, engineers have used a formula for the traditional minimum-delay cycle length (so-called Webster's method [43,44]), stated as follows:

$$C = \frac{(1.5 \cdot L) + 5}{1 - Y}$$

where C = cycle length in seconds,  
L = total lost time in seconds,  
Y = flow ratio, which is defined as,

$$Y = \sum_i \frac{v_i}{s_i}$$

and vi = critical lane volume (vph) during phase i, and  
si = saturation flow rate (vplphg) of the above lane.

To determine the minimum-delay cycle length according to Webster's method, the intersection-wide total lost time (L) and flow ratio (Y) could be obtained from HCS, possibly using one-touch HCS analysis from TRANSYT-7F. Simulation models such as TRANSYT and CORSIM do not directly provide L or Y as part of their output.

Since it cannot account for the effects of platoon progression and queue spillback, Webster's method is more effective for undersaturated, isolated intersections. The simulation-based cycle length optimization process within TRANSYT-7F accounts for not only the effects of volume and capacity at individual intersections, but also the effects of traffic flow patterns in the network, and spacing between signals.

Cycle length optimization may be requested in TRANSYT-7F by coding a minimum and maximum cycle length range. In addition, a cycle length search increment is necessary. The upper cycle length limit, lower cycle length limit, and cycle search increment may be specified on either the Edit > Analysis screen or the Edit > Optional > Global screen. Subsequently, TRANSYT-7F optimizes signal settings for every candidate cycle length between the minimum and maximum, at intervals of one increment. The "best" cycle length is the one that results in the best PI after all signal settings have been optimized.

High cycle lengths are helpful if they are needed to eliminate oversaturation (i.e., all queues served during one cycle), or if they are needed to serve oncoming platoons of vehicles in a progressive system. When volumes are low, low cycle lengths are better at minimizing delay because not only do they eliminate oversaturation, but the red times shown to each movement are shorter.

When volumes high enough that signal timing optimization is unable eliminate oversaturation, the percentage of green time allocated to each movement is often more important than the cycle length.

Changing the cycle length would have very little effect on vehicle delay, but changing the percentage of green time allocated to each movement would have a big effect on vehicle delay. In this type of a situation, the "optimum" cycle length reported at the end of an optimization run may actually be a somewhat random or arbitrary value that lies somewhere within the user's specified range of allowable candidate cycle lengths. (Consider Webster's minimum-delay cycle length formula above. This formula has "1-Y" in the denominator. When volumes are high enough that signal timing optimization is unable to eliminate oversaturation, Y is greater than 1.0, forcing the formula to generate a negative value. Thus, the minimum-delay cycle length is "undefined" when conditions are oversaturated.)

### Genetic Algorithm Cycle Length Optimization

If cycle length optimization has been requested, but split optimization has not been requested, initial splits for all candidate cycle lengths in the range are produced internally by the initial timing model. In general, allowing simultaneous optimization of cycle length and splits should allow the program to locate a better solution than it could if split optimization were prevented, although this process may take a long time on the computer. When split optimization is prevented, this allows the program to locate a local optimal solution much more quickly, but this will likely not be the global optimal solution.

When optimizing cycle lengths using the genetic algorithm, it is helpful for the cycle length range (between minimum and maximum) to be as narrow as possible. When the cycle length range is too wide, this can cause longer program running times for optimization, or in some cases may prevent the program from finding the global optimum cycle length. Forcing the program to evaluate ineffective cycle lengths slows down the genetic algorithm selection process. Preliminary cycle length evaluation runs can be helpful in determining a narrow range of effective cycle lengths for any given network. To save time, these preliminary cycle length evaluation runs could involve hill-climb optimization of cycle length-splits-offsets, or perhaps genetic algorithm optimization of cycle length-offsets (without splits).

When optimizing cycle lengths using the genetic algorithm, it is also helpful for the "Original Cycle Length" value (coded on the Edit > Analysis screen, or record type 5X) to be as close to the global optimum cycle length value as possible. This is sometimes a dilemma because it is not always possible to know the global optimum cycle length in advance. However, when the Original Cycle Length value is too far away from the global optimum cycle length value, this can cause longer program running times for optimization, or in some cases may prevent the program from finding the global optimum cycle length. This is because, in the first generation of the genetic algorithm, the initial timing model can only be applied for individual #1 who possesses the Original Cycle Length\*. All other "first generation" individuals must assume randomly generated cycle lengths and splits. The result is that the Original Cycle Length appears to be an especially good solution towards the beginning of the evolution process. If the global optimum cycle length is too far away from the Original Cycle Length, there might not be enough time to find that optimum solution.

*\*Note that if the initial timing flag has not been turned on by the user, it is still usually helpful for the Original Cycle Length to be as close to the global optimum cycle length as possible, because the user's coded initial timing plan is usually a reasonable and effective one. At least, it is usually much more effective than the timing plan that results from randomly generated splits.*

The Original Cycle Length affects the first individual in the first generation of optimization. The first individual contributes to the optimization process, and is also used to create "Initial" results for the output file. Typically the first individual in the first generation of optimization simply reflects the user-coded initial timing plan. However, if the initial timing flag is turned on, coordinated nodes on the optimization node list immediately adopt the Original Cycle Length. All other individuals in the first generation are unaffected by the Original Cycle Length. Subsequent individuals in subsequent generations aren't explicitly affected by the Original Cycle Length. However, they are usually implicitly affected by gene pool contributions from the first individual in the first generation.

If the user wishes to run genetic algorithm cycle length optimization without the initial timing flag selected, it is necessary to code an Original Cycle Length consistent with the coordinated nodes'

individual cycle lengths (on the Edit > Timing screen). If any local coordinated cycle length exceeds the Original Cycle Length, there may be error messages related to the sum of interval durations. If the Original Cycle Length exceeds any local coordinated cycle length, extra green time may be added to the final phase of the timing plan.

When running grouped node optimization, the Original Cycle Length affects the "difference of offset", which must be maintained between grouped nodes. This would affect both traditional TRANSYT optimization and direct CORSIM optimization. Refer to the section on Grouped Nodes for more details.

While cycle length optimization is running, the familiar TRANSYT banner screen will intermittently pop up in an adjacent "console" window that will be minimized and without focus. Each console window typically indicates a single simulation run, where each simulation run was used to evaluate an individual timing plan within the population.

The basis of determining the optimal cycle length, for both hill-climb and genetic algorithm optimization, is the Performance Index (PI) as specified within the TRANSYT-7F input (\*.TIN) file. Note that all PI's are meant to be maximized except for the Disutility Index (DI). The DI, which is a combination of vehicle delay, stops, fuel consumption, and queuing penalties, must be minimized.

TRANSYT-7F is also capable of optimizing uncoordinated operations. This means that for any uncoordinated intersections within the network, the splits and cycle length at these intersections can be optimized independently of the network background cycle length. For more details, refer to the documentation on Uncoordinated Optimization.

### Output Files

Genetic algorithm output files have a file name of "Genetic.txt". These output files present a summary listing of the original and final signal settings, plus a summary of the genetic algorithm input parameter values that were chosen. In order to obtain the typical detailed TRANSYT outputs (e.g., link performance, route performance, etc.), it is necessary to load the punch file containing the optimal timing plan, and perform a subsequent simulation-only run using this punch file. A sample output file is listed below.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                     Analysis Time Period:  
Jurisdiction:  
Run Title:   Four Node Sample File  
File Name:   FourNode.tin

Simulation Engine: TRANSYT

#### Genetic Algorithm Input Parameters

-----  
Crossover Probability (%):        30  
Mutation Probability (%):        1  
Convergence Threshold (%):       0.01  
Maximum Number of Allowable Generations: 250  
Actual Number of Optimization Generations: 250  
Population Size:                10  
Random Number Seed:            7781  
Elitist Method:                  True  
Lower/Original/Upper Cycle Length (sec): 60/100/100  
Initial Timing Flag:              Splits and Offsets  
Objective Function:               Disutility Index

## Optimization Results (Cycle Length, Splits, and Offsets)

-----

Node	Cycle		Offset		NS Phasing		EW Phasing	
	Init	Final	Init	Final	Init	Final	Init	Final
1	100	90	0	0	---	---	---	---
2	100	90	0	44	---	---	---	---
3	100	90	0	45	---	---	---	---
4	100	90	0	0	---	---	---	---

Node	Split #1		Split #2		Split #3		Split #4	
	Init	Final	Init	Final	Init	Final	Init	Final
1	10	10	47	35	10	10	33	35
2	10	10	47	35	10	10	33	35
3	10	10	47	35	10	10	33	35
4	10	10	47	36	10	10	33	34

	Control Delay (sec/veh)	Total Stops (%)	Fuel Consumption (gal/hr)	Travel Time (veh-hr/hr)	Performance Index
Initial	41.3	87	191	149	119.00
Final	27.2	92	171	118	98.60

The final solution originated in generation #250  
To obtain detailed output, process the punch file:  
FourNode.pun

## Hill-Climb Cycle Length Optimization

Hill-climb cycle length optimization in TRANSYT is relatively powerful due to the "brute force" treatment of the cycle lengths themselves. In other words, the cycle lengths are actually not hill-climbed. Instead, each candidate cycle length within the specified range is evaluated, and no candidate cycle length is skipped over. However, when evaluating each individual candidate cycle length, the splits and offsets are optimized using the hill-climb technique. Because of the automatic brute force treatment of cycle lengths, this sometimes allows hill-climb optimization to find a better solution than genetic algorithm optimization, particularly when the cycle length range is wide. On the other hand, if the user manually requested separate genetic algorithm optimization runs at each candidate cycle length, the best solution would be expected.

Regarding hill-climb optimization, initial timing plans for all candidate cycle lengths in the range are produced internally by the program. Since the initial timing model is applied automatically, results are not affected by whether the user has coded the initial timing flag or not. The optimum cycle length reported is the one with the best value of the PI. While cycle length optimization is running, the familiar TRANSYT banner screen will intermittently pop up in an adjacent "console" window that will be minimized and without focus. The user can switch focus to that window, if desired, to monitor the progress of the individual optimization runs for each candidate cycle length.

## Output Files

Hill-climb cycle length optimization output files have a file name of "t7fact.out". These output files present a summary listing of network performance at each candidate cycle length. In order to obtain the typical detailed TRANSYT outputs (e.g., link performance, route performance, etc.) following hill-climb cycle length optimization, it is necessary to load the punch file containing the optimal timing plan, and perform a subsequent simulation-only run using this punch file. A sample output file is listed below.

Analyst:                      Date Performed:  
Agency:                     Analysis Time Period:  
Jurisdiction:  
Run Title:   Four Node Sample File  
File Name:   FourNode.tin

Performance Index (PI): Disutility Index

---

#### CYCLE EVALUATION PERFORMANCE SUMMARY

---

Cycle Length (sec)	Average Delay (sec/veh)	Percent Stops (%)	Fuel Consumption (gal/hr)	Travel Time (veh-hr/hr)	Performance Index
60	41.5	121	195	149	138.20
70	30.6	98	175	125	107.40
80	28.5	93	172	121	101.30
90	27.4	92	171	118	99.05
100	32.4	87	177	129	104.60

TRANSYT-7F minimizes the Disutility Index  
The optimal cycle length is 90 seconds  
To obtain detailed output, process the punch file:  
FourNode.pun

## PROS Optimization

### PROS-Only Optimization

When optimizing PROS in a grid network, the default is to give the same weight to all nodes on all arteries; however, directional weightings can be employed to give priorities to specific arteries and/or directions. In a simple PROS optimization, the main objective is to maximize PROS for the arterial street through movements. Because only PROS are optimized, this is not suitable for optimizing networks where all intersections are not defined on a particular artery.

PROS-only optimization also suffers from the same disadvantages as the maximal bandwidth approach in that the actual traffic demand is not explicitly considered. For example, split optimization generally forces the green times of minor movements (at nodes where split optimization is permitted) to their minima. This is because PROS-only optimization does not provide criteria for setting green times for these movements. One of the following procedures can be used to avoid oversaturating minor movements and ensure an equitable distribution of green times when optimizing PROS:

1. For pre-timed controllers, initial splits that equalize the degrees of saturation can be requested. For traffic-actuated controllers, initial splits that achieve the target degree of saturation can be requested. All splits are then held constant in the first optimization step.
2. The splits can be coded by the user and fixed during the first optimization step. These splits should be calculated externally.
3. If the splits are requested to be optimized in the first optimization step, the minimum phase lengths for the minor movements have to be set high enough to avoid saturating them. This will likely require trial and error, although the user may also calculate the minimum effective green that will achieve a desired degree of saturation, according to the traditional actuated control model.

Another problem with PROS-only optimization is that it may cause the PROS to increase in one direction (not necessarily the critical direction) at the expense of PROS in the other direction. This problem can generally be avoided by coding proper directional weightings, or by coding initial timings from a maximal bandwidth optimization program.

### PROS & DI Optimization

This is an extension of the PROS-only optimization process described earlier. To deal with other movements, and to fine-tune the design, the procedure also tries to minimize the disutility index (DI) elsewhere in the system, while not reducing the arterial PROS. If this option is selected, TRANSYT performs two steps of the traditional hill-climbing procedure as follows:

1. The performance index (PI) is initially based purely on the PROSe on the arteries. A change in timing is retained if it increases the PROSe. When computing PROSe for a multi-arterial system, the PROS and CPROS are computed first for each artery. Then, the aggregate PROS and CPROS for the whole system are determined by summing up their values over all designated arteries. The PROSe for the system is determined based on these two values.
2. In pass 2, the hill-climbing procedure is employed to minimize the DI without reducing the PROSe value achieved in step 1. This allows for some adjustments of the offsets and splits on the arteries. In addition, timing for intersections not considered in pass 1 (because they are not on any "designated artery") are optimized entirely in pass 2.

Note that multiple steps or passes cannot be performed when genetic algorithm optimization is used. As such, PROS/DI is a better policy than PROS & DI for genetic algorithm optimization.

### PROS/DI Optimization

The purpose of the PROS/DI formulation is to combine the advantages of maximizing PROS with the advantages of minimizing DI. The procedure calculates the effective PROS on the arteries and the DI for the entire system after each timing shift. The shift is retained if it increases the PROS/DI ratio. Unlike simple PROS optimization, this policy considers both the PROS and DI at the same time; thus, it eliminates the need for the second optimization step from the previously described objective function.

Because the minor movements are accounted for in calculating the DI, splits can be optimized in addition to offsets. This policy tries to maximize progression, subject to maintaining sufficient green times for the minor movements. In addition, the policy attempts to find the set of offsets that clears the existing queue before a platoon arrives. Nodes not on the designated arteries are explicitly considered in the traditional DI calculation; thus, their offsets and splits are optimized in concert with the arterial progression.

By coding a specific weighting factor, the relative weighting of PROSe to the DI in the objective function can be varied. This allows for fine-tuning the relative importance of PROS on the arteries versus the DI for all links. The relative weighting of PROSe within the objective function can be varied by:

$$PI = \frac{(100 \bullet PROS_e)^{WP}}{DI}$$

where WP is the relative weight of PROS to the DI.

This allows for fine-tuning the relative importance of PROS on the specified arteries versus the DI for all links. Experience with the early version of the model indicated that, in some cases, optimizing splits based on the PROS/DI strategy without weighting tended to discriminate against minor turning movements. To correct this anomaly, research [18] from the University of Florida Transportation Research Center (TRC) suggests that a WP value of 50% would generally reduce the weight of PROS

relative to the DI. This increases the weight of minor movements in the optimization process and, generally, results in a better optimization of splits while maintaining good progression on the arteries. It should be noted, however, that this value of WP (50%) is not likely to be the ideal value for all networks. A trial and error process using different WP values may result in the best overall design.

### Directional Weighting

It may be desirable to favor one direction of travel over another, on an artery, such as during peak periods. In addition, it may be preferable to give different weights to different arteries. By weighting the arteries relative to each other and/or the two directions of travel along each artery, it may be possible to achieve a desired design or policy. Directional weighting factors can be coded in the Progression Route dialog screen (accessible from the Map View), or on the Edit > Optional > Route > Arterial Weighting screen.

Clearly, a 90:10 directional weighting would focus on maximizing progression in the forward direction, possibly at the expense of the reverse direction. Note that accepting a 50:50 directional weighting may have a substantially different effect on the optimization process, when compared to not using any directional weighting. Refer to the PROS section of the documentation for more details.

Inside the TIN file, Forward and Reverse Weightings for any given progression route are listed on record type 43, but only fields 1 through 4 are recognized by the Progression Route dialog. If directional weighting is not used, record type 43 will be omitted for the given route.

The weighting factors are proportional. For example, if '66' is coded for one route and '33' for another, the former will have twice the weight applied to PROS than the first. Although this is not a specifically required approach, the weighting factors can be considered as percentages. For example, to achieve 60% of progression in the forward direction, code '60' and '40' as the forward and reverse factors, respectively. If a 50:50 split on a second route is desired, but double the priority as the first route, code '100' in both fields as both the forward and reverse factor.

## **Optimization Under Congested Conditions**

In release 8, several new objective functions were added to TRANSYT. These new objective functions were designed to address severely oversaturated conditions at a time when multi-period optimization was not yet available. In release 10, multi-period optimization was added, also for the purpose of mitigating severely oversaturated conditions. This raises the question of which of the available techniques is most effective for optimizing oversaturated conditions.

To answer this question, it is helpful to define "severely" oversaturated conditions. A severely oversaturated network is one in which most links are oversaturated. In other words, most links in the network have a degree of saturation above 100% at the end of the peak 15-minute period. The traditional objective functions and single-period optimization techniques tend to be ineffective in these situations. For example, the disutility index (DI) is a less reliable indicator of network performance, because delay and stops cannot be measured accurately when links are oversaturated at the end of the time period. Regarding progression opportunities (PROS), vehicles have a tendency to stop at almost every intersection (when links are oversaturated at the end of the time period), such that there is nothing left to optimize.

The objective functions from release 8 focus on minimizing queuing, and maximizing the number of vehicle trips. Even when most links are oversaturated at the end of the time period, it is still possible to measure queue lengths and vehicle trips relatively accurately. Therefore, single-period optimization using these objective functions has a good chance of being more effective than single-period optimization using the traditional objective functions (DI, PROS). But what about multi-period optimization?



Ideally, the first time period would be chosen such that there were no residual queues at the very beginning, and the last time period would be chosen such that there were no residual queues at the very end. This allows delay to be measured accurately, even though the time period boundaries during the middle of the analysis experience residual queues. When delay can be measured accurately, it is logical to use the DI as the objective function for multi-period optimization.

The issue of whether applying the DI (with multi-period optimization) is always preferable to applying the release 8 objective functions (with single-period optimization) is uncertain. It is true that creating a multi-period input file is slightly more time-consuming and error-prone than creating a single-period input file. However, creating a multi-period file allows minimization of delay and stops, which may be more defensible than minimizing queuing, or maximizing vehicle trips. Refer to the Multi-Period Optimization section of the documentation for more details.

The objective functions from release 8, which require step-wise simulation, include:

1. Minimize the product of DI and queuing ratio (QR), i.e., QR\*DI, where queuing ratio is defined as:

$$QR = \sum_i \left( \frac{BOQ_i}{qc_i} + 1 \right)^{w_i}$$

where,

BOQi = average back of queue for link i  
 qci = queuing capacity of link i  
 wi = user-specified weight for link i

Regarding external links, queuing ratios are only computed for links with a queuing capacity of 8 vehicles or higher. Or, if no queuing capacity has been coded, queuing ratios are only computed for link lengths of 200 feet (61 meters) or higher. This measure is in place to prevent unimportant links from diluting the analysis.

The QR\*DI performance index (PI) value is always multiplied by '-1', resulting in a negative value. This treatment is in place so that all PI's (except for the disutility index) can be maximized during optimization.

2. Maximize the ratio of throughput (TR) over the DI, i.e., TR/DI, where throughput is defined as:

$$TR = \sum_i \left( \frac{\sum_j^s OUT_{ij}}{\sum_j^s IN_{ij}} \cdot 100 \right)^{w_i}$$

where,

OUTij = OUT-pattern for link i at step j  
 INij = IN-pattern for link i at step j  
 wi = user-specified weight for link i

3. Optimize throughput & DI, where throughput is initially maximized on selected links in the system. In the second stage, the DI of the entire system is minimized without reducing the throughput. This function is similar to the "PROS & DI" objective function, with throughput in the place of PROS.

4. Maximize throughput, with a penalty being imposed on the optimization if a maximum v/c ratio threshold is exceeded. Such a penalty will prevent minor links from being significantly shortchanged,

relative to the major movements.

5. Minimize the system DI, using high weights on the critical links. (Refer to the Weighting Screen documentation for more details.)

Within some oversaturated networks, the traffic volume can be so high that even a perfectly optimized signal timing plan could not prevent certain parts of the network from being oversaturated. According to conventional wisdom, if certain links within the network must be oversaturated, it is preferable that these links be "external" links instead of "internal" links. By definition, external links have no nearby upstream signal, and the traffic operations on these links cannot affect any upstream signal. Internal links have nearby upstream signals, and the traffic operations on these links potentially affect the upstream signal (through queue spillback), and are potentially affected by the upstream signal (through progression and metering effects).

Therefore, if certain links within the network are destined for failure, it may be preferable for these links to be external links, because queue spillback would not interfere with any upstream signals. On the other hand, queue spillback from internal links can potentially cause gridlock. The signal timing designer can sometimes take steps to ensure that internal links do not become oversaturated. These steps include increasing the minimum phase times on internal links, applying optimization weighting factors applied to internal links, etc.

At the end of an optimization run with severely oversaturated conditions, users will also sometimes observe unexpected optimum cycle lengths and offsets recommended by the program. This occurs because when traffic conditions are severely oversaturated, changes to the cycle lengths and offsets tend to have little or no impact on traffic operations or performance. Therefore, in this type of a situation, the "optimum" cycle lengths and offsets reported at the end of an optimization run may actually be somewhat random or arbitrary values.

## Grouped Nodes

If two or more intersections operate together and the user wishes to retain the offset relationship between them, these nodes can be defined as "grouped nodes". The most common applications of grouped nodes are:

- two closely spaced intersections, particularly if they are governed by a single controller
- diamond interchanges imported from HCS-Interchanges
- diamond interchanges timed with PASSER III [22]
- several intersections timed by another method that the user wishes to keep TRANSYT-7F from changing

Grouped nodes are specified by coding a negative number on the optimization node list. The optimization node list is located on the Edit > Optional > Global screen (or record type 2). If two or more nodes are grouped, the secondary node number(s) must be negative and must immediately precede the primary node number (which has a positive number) in the list. For example, if the list contains 2 -6 8 10, node 6 will be grouped with node 8 (note: do not refer to grouped nodes with negative numbers elsewhere in the data file). The offset difference between the grouped nodes will be held constant during the optimization run. Offsets and phase lengths (splits) for the nodes surrounding the group will be optimized normally. The user must provide the initial timing plan at grouped nodes. Grouped node optimization tends to be more effective when using the genetic algorithm.

### Genetic Algorithm Optimization with Grouped Nodes

The genetic algorithm can optimize cycle length, splits, and offsets at grouped nodes, but not phasing sequence. Phasing sequence optimization can still occur at non-grouped nodes while grouped

nodes are being optimized. The number of phases must be equal among intersections within each group. Cycle length and split times can be optimized at a grouped node, and these settings will be copied to other nodes within the group. Offsets can be optimized, but the relative difference of offset between the grouped nodes will be held constant. The Original Cycle Length (on the Edit > Analysis screen) is sometimes a factor in determining the difference of offset. If cycle length optimization or phasing sequence optimization are in effect, split optimization must be allowed at grouped nodes, because the initial timing model does not provide timing for grouped nodes. If split optimization is disallowed (on the Edit > Timing screen) at a grouped node, split optimization will automatically be disallowed for other nodes within the group. If offset optimization is disallowed at one grouped node, offset optimization will automatically be disallowed for other nodes within the group. There can be a maximum of four nodes per group.

On the optimization node list (Edit > Optional > Global > Optimization Node List, or record type 2), each set of grouped nodes has one or more negative node numbers, followed by a positive node number. The negative node numbers are considered the grouped "slaves", and the positive node number is considered the grouped "master". The grouped master node number actually has its timing optimized, so its initial timing plan may affect the optimization process. The grouped slave node numbers simply have their cycle length and green splits automatically set equal to that of the grouped master. Minimum phase times should be similar or equal at all grouped nodes, to avoid errors during the optimization process. The grouped slave node numbers have their offsets set equal to that of the grouped master, minus the original difference of offset, if any. Note that offsets are typically not optimized for the "master controller" node number, if one has been defined (Edit > Optional > Global > Network Parameters, or record type 10). The phasing sequence and number of phases are always held constant at grouped nodes.

On the Edit > Timing screen, grouped nodes should be marked as "coordinated signals", because they all experience platooned vehicle arrivals. If other coordinated signals in the traffic network have a cycle length that differs from the grouped node cycle length, then the grouped nodes should be analyzed or optimized in their own separate TRANSYT-7F input (\*.TIN) file. The TRANSYT model assumes one background system cycle length per data file, with certain intersections allowed to have different cycle lengths if they are marked as uncoordinated. If CORSIM is being used as the underlying model instead of TRANSYT, then it is acceptable to analyze or optimize multiple coordinated signal systems in one data file, having multiple background cycle lengths. However, beware that optimization tends to be more efficient and effective when it is allowed to focus on a small number of signalized intersections. If the engineer tries to optimize the entire city at once, the result will probably be an ineffective timing plan. Each cluster of closely spaced intersections tends to operate more efficiently from its own customized design.

The sample genetic algorithm optimization output file listed below illustrates an example of what could happen by grouping nodes 10, 11, 12, and 13. Notice that the optimized offsets at all four of these intersections are equal to their original offsets, and the optimized splits at these intersections are also equal (though different than the original splits). Offsets at grouped nodes will sometimes change during the optimization process, but the relative difference of offset between these nodes will remain constant. During a phasing sequence optimization run, the phasing sequence will be held constant at the grouped nodes, but may be optimized at other nodes (e.g., node #7 below).

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                     Analysis Time Period:  
Jurisdiction:  
Run Title: Benchmark Blvd  
File Name: Level2Perm.tin

Simulation Engine: TRANSYT

## Genetic Algorithm Input Parameters

```

Crossover Probability (%):      30
Mutation Probability (%):       1
Convergence Threshold (%):     0.01
Maximum Number of Allowable Generations: 40
Actual Number of Optimization Generations: 40
Population Size:               10
Random Number Seed:            7781
Elitist Method:                True
Initial Timing Flag:           Splits Only
Objective Function:             Disutility Index

```

## Optimization Results (Phasing Sequence, Splits, and Offsets)

```

Node Cycle   Offset NS Phasing EW Phasing
Init Final Init Final Init Final Init Final
6 100 100 37 37 LT LT LET LET
7 100 100 56 81 LNT LT LET LET
8 100 100 16 16 NS NS --- ---
9 100 100 35 47 --- --- WT WT
10 100 100 55 55 LST LST LET LET
11 100 100 14 14 LST LST LET LET
12 100 100 33 33 LST LST LET LET
13 100 100 53 53 LST LST LET LET
14 100 100 12 12 --- --- LET LET
15 100 100 32 25 --- --- WT WT

```

```

Node Split #1 Split #2 Split #3 Split #4 Split #5 Split #6
Init Final Init Final Init Final Init Final Init Final Init Final
6 9 10 14 14 40 40 7 7 30 29
7 9 9 7 8 62 54 7 7 3 22 12
8 72 67 14 18 14 15
9 7 7 69 69 24 24
10 22 14 3 7 30 33 12 11 3 3 30 32
11 22 14 3 7 30 33 12 11 3 3 30 32
12 22 14 3 7 30 33 12 11 3 3 30 32
13 22 14 3 7 30 33 12 11 3 3 30 32
14 7 7 3 3 64 41 26 49
15 7 10 75 69 18 21

```

	Control Delay (sec/veh)	Total Stops (%)	Fuel Consumption (gal/hr)	Travel Time (veh-hr/hr)	Performance Index
Initial	64.0	80	811	759	590.40
Final	24.6	73	564	414	329.50

The final solution originated in generation #37  
 To obtain detailed output, process the punch file:  
 Level2Perm.pun

## Hill-Climb Optimization

When running hill-climb optimization, TRANSYT treats grouped nodes as follows:

1. Their differences of offset are held fixed, although the group's offset can be optimized within the system.
2. Their splits may be optimized, or optionally held fixed.

In a hill-climb cycle length optimization run, all nodes should be included on the optimization node list, and grouped nodes are ignored.

---

# Genetic Algorithm Optimization

## Genetic Algorithm Optimization

Genetic algorithm (G.A.) optimization is a theoretical improvement over the traditional hill-climb optimization technique employed by TRANSYT in the 20th century. The genetic algorithm has the ability to avoid becoming trapped in a "local optimum" solution, and is mathematically best qualified to locate the "global optimum" solution. For a discussion on the meaning of local and global optimum solutions, refer to the documentation on Hill-Climb Optimization.

TRANSYT-7F is capable of performing G.A. optimization using either CORSIM or TRANSYT as the underlying simulation engine. When TRANSYT is used as the engine, all signal settings (cycle length, phasing sequence, splits, and offsets) can be optimized simultaneously, using the genetic algorithm. The following types of optimization require G.A., and cannot be accomplished via hill-climb:

1. Multi-period optimization
2. Direct CORSIM optimization
3. Phasing sequence optimization
4. Uncoordinated optimization

Optimization of all signal settings (cycle length, phasing sequence, splits, and offsets) simultaneously requires excessive program running times on the computer, and large amounts of computer memory. In some cases it is more practical and effective to perform "sequential" optimization of the individual signal settings, instead of "simultaneous" optimization of all signal settings.

In general, when signal settings must be optimized separately or sequentially, it is recommended to establish the cycle length and phasing sequence first. Subsequently, the cycle length and phasing sequence are held constant, and the splits and offsets are refined by an additional optimization run. Time permitting, it may be helpful to repeat the same steps using the original optimal solution as the starting point. If the program is unable to locate a better solution than the original optimal solution, the design process may be complete. This sequence of steps could be applied as follows:

1. Hill-climb cycle length optimization
2. Phasing sequence, offsets, and yield points, simultaneously (G.A.)
3. Splits and offsets (G.A. or hill-climb)

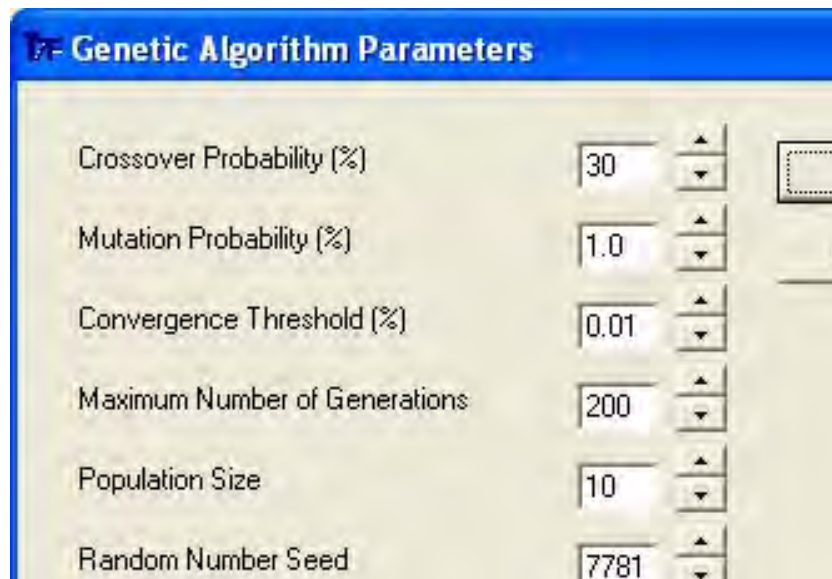
For a discussion of microscopic (CORSIM) optimization, refer to the Direct CORSIM Optimization section of the documentation. When using TRANSYT as the simulation engine, optimization is based on the user-selected performance index (PI), which is defined inside the standard TRANSYT-7F input (\*.TIN) data file. If the TRANSYT Disutility Index (DI) is selected as the PI, the program will attempt to minimize this value, whereas it would attempt to maximize any other PI value.

When using TRANSYT as the simulation engine, G.A. optimization supports four different types of simulation options. These include single-cycle link-wise simulation, single-cycle step-wise simulation, multi-cycle step-wise simulation, and multi-period step-wise simulation. As usual, link-wise simulation provides the fastest running times on the computer, but provides the least detailed simulation. Multi-cycle step-wise simulation may be preferable when oversaturated conditions prevail. Note that there is no available option for multi-cycle link-wise simulation.

If the initial timing flag (coded on the Edit > Analysis screen) has been set, a preliminary TRANSYT run is performed prior to G.A. optimization. Following the preliminary run, a punch file is generated, containing a complete initial timing plan. This initial timing plan is used as the starting point for G.A. optimization. Of course, if the initial timing flag has not been set, then the complete initial timing plan from the user's data file is used as the starting point for G.A. optimization.

### Genetic Algorithm Parameters

G.A. parameters can be customized by the user within the Edit > Analysis > Genetic Algorithm Parameters dialog screen (or record type 5X). These settings affect how much time on the computer will be required to locate the global optimum solution. All settings from this screen (illustrated below) are saved into the TRANSYT-7F input (\*.TIN) file.

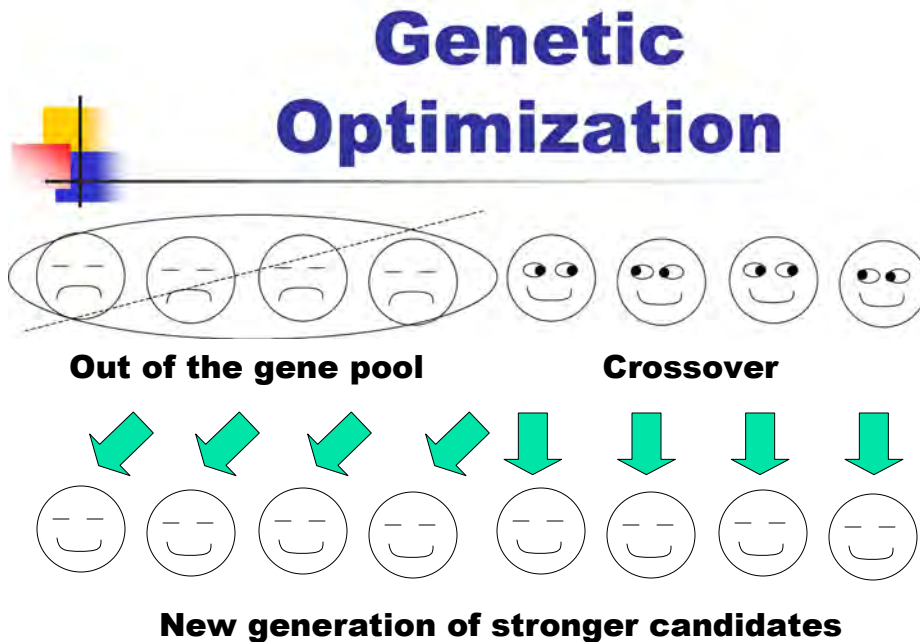


Parameter	Value
Crossover Probability (%)	30
Mutation Probability (%)	1.0
Convergence Threshold (%)	0.01
Maximum Number of Generations	200
Population Size	10
Random Number Seed	7781

Usually, most G.A. parameter values do not need to be modified by the user. For more thorough optimization and better timing plans, the Maximum Number of Generations must always be increased, although this will cause longer program running times on the computer. Refer to the Recommended G.A. Parameter Values section of the documentation for a listing and discussion of effective settings for specific networks. Theoretical aspects of the G.A. parameters are discussed in detail below.

### Crossover and Mutation

The genetic algorithm initially considers a trial population of signal timing plan candidates. Within this population a competition takes place where the weak candidates are discarded, as illustrated below, and the strong candidates are allowed to breed and perform crossover. Following crossover, a new generation of candidate solutions is born, and a mutation process is performed to ensure that fresh solutions are considered. Following mutation, another new competition takes place where the weak candidates are discarded and only the strong survive. This entire process takes place for numerous successive generations until convergence is achieved on the global optimal solution.



Traffic Signal Timing Optimization

Crossover is a mathematical breeding between strong candidate solutions. Following crossover, a new generation of candidate solutions is born. Low crossover probabilities result in younger generations of solutions that are more similar to the older generations, whereas high crossover probabilities result in younger generations that are less similar to the older generations.

Mutation is an arbitrary modification of randomly selected genes. The proportion of mutated genes is affected by the mutation probability. The mutation process prevents the gene pool from becoming stale and failing to recognize new possibilities. In addition, the mutation process allows the genetic algorithm to locate the global optimum solution instead of a local optimum solution. Beware that excessively high mutation probabilities may significantly increase the amount of time required for optimization, or may prevent optimization altogether.

If the optimization process is converging too quickly, or if the user wishes to examine a wider range of timing plan possibilities, then increasing the mutation probability may be helpful. If the optimization process is not converging quickly enough, or if the user wishes to acquire a reasonable and effective local optimum solution, then decreasing the mutation probability may be helpful.

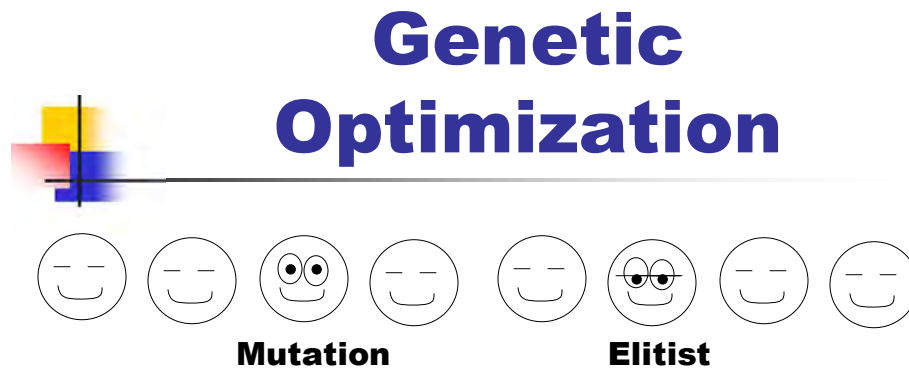
### Population Size

The population size is the number of individuals within each generation. For example, if 200 generations of optimization were performed with a population size of 10, this would result in a total of 2000 simulation runs. The default population size for brand-new data files is 10, but the software now allows the user to request population sizes between 6 and 99. Lower population sizes sometimes allow for faster program running times on the computer, but higher population sizes may facilitate genetic diversity and faster optimization of certain networks.

If the optimization process is converging too quickly, or if the user wishes to examine a wider range of timing plan possibilities, then increasing the population size may be helpful. If the optimization process is not converging quickly enough, or if the user wishes to acquire a reasonable and effective local optimum solution, then decreasing the population size may be helpful.

### Elitist Method

During each generation, the elitist individual (illustrated below) is chosen as the strongest of all individuals. If the elitist method has been selected, the elitist individual is then cloned and retained for subsequent generations, essentially enabling that individual to repeatedly perform crossover and mutation in subsequent generations. When one of the children demonstrates stronger characteristics than the original elitist individual, that child becomes the elitist individual, and the original elitist individual is finally allowed to perish. If the elitist method is not selected, only the children will be eligible for crossover and mutation, and the elitist individual will always perish in the next generation after crossover.



**Next generation has same size population. Now who is strongest?**

Traffic Signal Timing Optimization

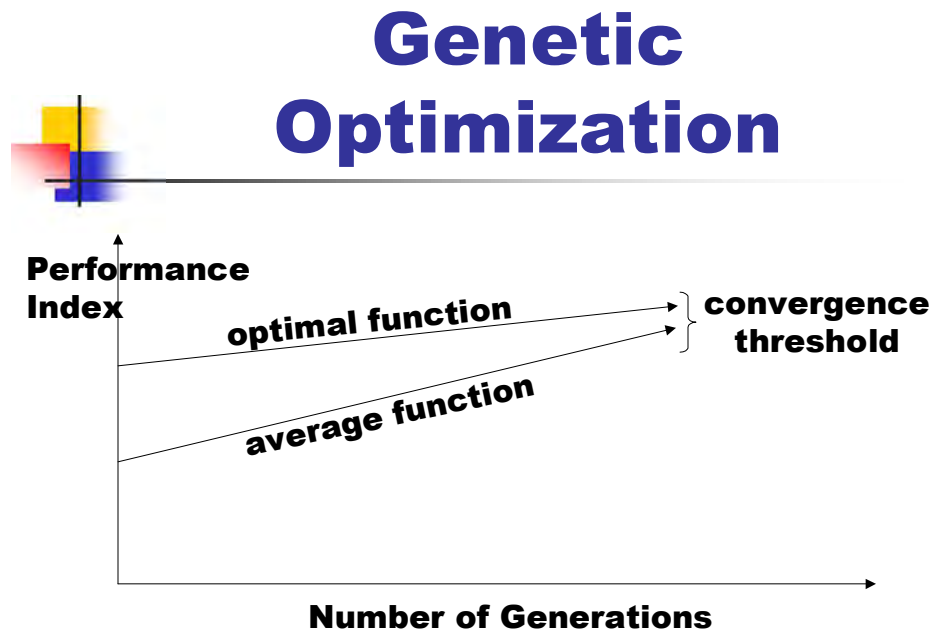
### Convergence Threshold and Maximum Number of Generations

While running G.A. optimization, the TRANSYT-7F interface continuously displays the optimal function value and the average function value. The optimal function value represents the best performance index value when considering all candidate timing plans in that particular generation. The average function value is the average performance index value from all candidate solutions in that particular generation.

During successive generations, the optimal function value and the average function value tend to improve due to the genetic selection process. The average function value can occasionally become worse due to a "bad mutation", but the optimal function value should always stay the same or improve (if the elitist method is used). Convergence is achieved when the average function value becomes sufficiently close to the optimal function value, as illustrated below. Increasing the maximum number of generations (or decreasing the convergence threshold) tends to produce longer program running times, but better timing plans. Of course, once the global optimum solution has been reached, additional generations and tighter convergence thresholds have no additional benefit. Complex traffic networks tend to require substantially more generations of candidate solutions before an optimal solution is



obtained.



Traffic Signal Timing Optimization

### Random Number Seed

Crossover, mutation, and the competition between individuals involve a certain degree of randomization. If the Random Number Seed value is left unchanged by the user, multiple runs with the same input data should always generate the same optimal design. If the random number seed is changed by the user, this may change the amount of time required to locate the optimal solution. Given the necessary G.A. parameter values (i.e., a high maximum number of iterations together with a low convergence threshold), the algorithm is designed to converge on the (one and only) global optimum solution. In order to obtain internal processes that are truly "mathematically random", it is recommended that the random number seed value should end in a '1', '3', '7', or '9'.

### Program Running Times

A potential drawback of the genetic algorithm is increased program running times on the computer; however, this potential drawback continues to be minimized by the ever-increasing processing speeds of today's computers. If the G.A. program running times are found to be excessively high on a particular computer, here is a short list of possible solutions:

- Specify a limited number of simulation cycles, or perhaps even single-cycle simulation, within the TRANSYT or CORSIM input file.
- Specify a higher convergence threshold.
- Specify a lower mutation probability.
- Specify a lower population size.
- Specify a lower maximum number of generations.
- Close other programs that may be running in the background.
- Use TRANSYT as the simulation engine, instead of CORSIM.
- Use a computer with a faster processing speed.

- Disable or exit any anti-virus programs that are running in the background.
- Deactivate the "Extended Run" and "Use Rest Stops" options within the Options > Genetic Algorithm screen.

Beware that changing the G.A. parameter values to reduce program running times might prevent the algorithm from locating the global optimum solution. A local optimum solution, such as what would be expected using hill-climb optimization, might be reached instead. Program running times are affected by the duration of simulation, and by the G.A. parameters. The duration of simulation is specified within the standard TRANSYT input (\*.TIN) data file, or the standard CORSIM input (\*.TRF) data file. In TRANSYT, step-wise simulation is more realistic, but runs more slowly than link-wise simulation.

### Output Files

Genetic algorithm optimization output files have a file name of "Genetic.txt". These output files present a summary listing of the original and final signal settings, plus a summary of the G.A. input parameter values that were chosen. In order to obtain the typical detailed TRANSYT outputs (e.g., link performance, system summary, route statistics, signal timing tables, etc.), it is necessary to load the punch file containing the optimal timing plan, and perform a subsequent simulation-only run using this punch file.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                      Analysis Time Period:  
Jurisdiction:  
Run Title: Benchmark Blvd  
File Name: Level2Clean.tin

Simulation Engine: TRANSYT

### Genetic Algorithm Input Parameters

```
-----
Crossover Probability (%):      30
Mutation Probability (%):       1
Convergence Threshold (%):     0.01
Maximum Number of Allowable Generations: 40
Actual Number of Optimization Generations: 40
Population Size:                20
Random Number Seed:            7783
Elitist Method:                 True
Lower/Original/Upper Cycle Length (sec): 70/90/90
Initial Timing Flag:            Splits and Offsets
Objective Function:             Disutility Index
```

### Optimization Results (Cycle Length, Phasing Sequence, Splits, and Offsets)

```
-----
Node Cycle   Offset  NS Phasing EW Phasing
   Init Final Init Final Init Final Init Final
  6  90  70   0  39  LT  NTS LET  LET
  7  90  70   0   0  LT  LT LET  LET
  8  90  70   0   0  NS  NS ---  ---
  9  90  70   0  18  ---  --- WT  WT
 10  90  70   0  22  LST LNT LET  LET
 11  90  70   0   5  LST LNT LET  LET
 12  90  70   0   0  LT  NTS LET  LET
 13  90  70   0   0  LT  LT LET  WTE
 14  90  70   0   0  ---  --- LET  LET
 15  90  70   0  44  ---  --- WT  WT
```

Node	Split #1	Split #2	Split #3	Split #4	Split #5	Split #6	
Init	Final	Init	Final	Init	Final	Init	Final
6	8	11	8	34	27	12	17
7	8	11	7	4	49	34	13
8	66	50	12	10	12	10	
9	9	8	59	45	22	17	
10	8	8	11	11	52	27	8
11	8	8	12	8	31	24	9
12	8	8	6	5	35	24	12
13	9	9	5	9	28	11	11
14	8	7	9	10	49	14	24
15	8	11	65	45	17	14	

	Control Delay (sec/veh)	Total Stops (%)	Fuel Consumption (gal/hr)	Travel Time (veh-hr/hr)	Performance Index
Initial	51.8	75	732	659	507.30
Final	23.0	80	560	398	332.10

The final solution originated in generation #40  
To obtain detailed output, process the punch file:  
Level2Clean.pun

## Recommended G.A. Parameter Values

Some trends have emerged regarding which G.A. parameter values appear to facilitate the optimization process, depending on network size, and depending on which signal settings (cycle length, phasing sequence, splits, offsets) are being optimized. These recommendations are considered approximate, and may not be suitable for all networks.

### Crossover Probability

It is recommended to leave this value alone, at the default value of 30%. In preliminary testing, crossover probability values between 30% and 70% did not significantly alter the efficiency of optimization. Values below 30% or above 70% appear to make the optimization process less efficient.

### Mutation Probability

For small networks with 4 intersections or fewer, a mutation probability of 4% is recommended. For larger networks with 5 intersections or more, a mutation probability of 1% is recommended.

### Convergence Threshold

It is recommended to leave the convergence threshold value alone, at the default value of 0.01%. This allows the Maximum Number of Generations to have primary control over the level of optimization.

### Maximum Number of Generations

For small networks and simple optimization problems, 50 generations should be sufficient for locating the optimal solution. For larger networks and complex optimization problems, 200 generations or more may be needed.

### Population Size

For simple optimization problems such as offsets only, splits only, or phasing sequence and offsets, a population size of 10 is recommended. For complex optimization problems such as phasing sequence and splits, a population size of 20 is recommended.

### Random Number Seed

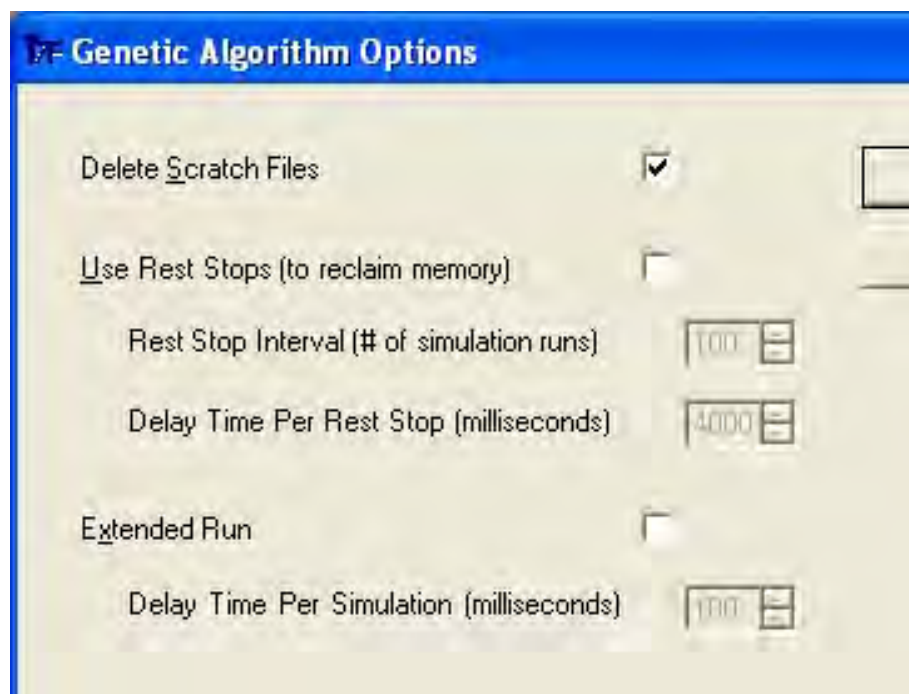
In most cases it is recommended to leave the random number seed value alone. If the other genetic algorithm parameters are chosen correctly, the optimization process should be successful. If the optimization process appears to be close to reaching the global optimum solution, but is getting stuck on a nearby local optimum solution, a second optimization run using a different random number seed might help.

### Elitist Method

Using the elitist method is recommended. The non-elitist method appears to make the optimization process less efficient in most cases.

## Genetic Algorithm Options

The Genetic Algorithm Options screen, which can be accessed in the GUI by selecting Options > Genetic Algorithm, is illustrated below.



### Delete Scratch Files

TRANSYT or CORSIM input files are generated for each candidate solution, and after processing, an output file will also be present for each input file. These files can be deleted following G.A. processing in order to clean up the hard drive, or they can be retained for further inspection. For example, if the cycle length and/or phasing sequence did not change during optimization, the scratch files

could be used to confirm that different alternatives were considered. In this situation it might be preferable to examine scratch files after only the second generation, because towards the end of optimization, most (if not all) data files have long since adopted the optimal characteristics.

This checkbox also affects whether scratch files will be automatically deleted for other types of runs (e.g., hill-climb cycle length optimization, traffic-actuated signal timing estimation, Spyglass, CORSIM processing). Note that scratch files are often located within the Windows TEMP folder.

The setting for Delete Scratch Files may be changed as long as TRANSYT-7F remains running, but is reset to the default (checked on) every time TRANSYT-7F is closed and re-opened.

### Rest Stops

The "rest stops" capability was implemented to prevent G.A. optimization runs from draining computer resources. Rest stops are sometimes needed when using relatively fast computers in conjunction with relatively small input files. In this situation, the execution of too many simulation runs in a short period of time prevents the computer from being able to efficiently reclaim memory. This causes the program to "freeze", and also interferes with other applications on the computer. Rest stops allow the computer to reclaim memory normally, and complete the optimization run. When using a slow computer, or a large input data file, rest stops will probably not be needed.

Rest stops are also not needed for direct CORSIM optimization, which tends to run much more slowly than macroscopic TRANSYT-based optimization. If the rest stops checkbox is turned on, it will be automatically ignored by direct CORSIM optimization.

By default, when rest stops are activated, the program will take a 4-second rest stop after every 100 simulation runs. During preliminary testing, these settings have allowed most optimization runs to complete comfortably on most computers. Reducing the frequency or duration of rest stops might allow the optimization runs to finish more quickly, but might also increase the probability of draining computer resources. Rest stop settings are saved into memory on any given computer, so if rest stops are frequently needed, these settings will be automatically loaded and applied during all future TRANSYT-7F runs.

### Extended Run

Preliminary testing has shown that "extended" genetic algorithm optimization runs, requiring several hundred or thousand generations, have the tendency to drain computer resources. Shorter genetic algorithm optimization runs, involving 200 generations or fewer, should not significantly drain the resources of today's computers. If a genetic algorithm optimization run appears to be draining computer resources, refer to the section on Genetic Algorithm Optimization for a separate list of possible solutions.

Powerful computers that have extensive resources are generally able to execute a higher number of optimization generations. Once the computer resources have been drained significantly, the optimization run tends to slow down or even abort processing. If the optimization run aborts, an error message is sometimes issued that says "Output file not found". This indicates that the computer does not have enough available resources to perform additional simulation runs and produce additional output files.

Many (or most) optimization runs may not require more than a few hundred generations to locate the global optimum solution, or to locate an acceptable local optimum solution. However, optimization of splits significantly increases the number of generations required. Because there are several splits at each intersection, and several intersections within the network, optimization of splits results in a much larger set of candidate timing plans. In general, simultaneous optimization of multiple signal settings (cycle length, phasing sequence, splits, and offsets) requires more generations than sequential optimization of individual signal settings. Similarly, optimization of large networks with many signalized intersections requires more generations than optimization of smaller networks with fewer signalized

intersections.

Each individual computer should have a specific maximum number of generations that can be executed, given the available memory and resources. If an optimization run is desired that runs additional generations beyond what the computer can handle, there are a few possible solutions to this:

1. More effective genetic algorithm optimization settings -- Settings such as the population size and mutation probability affect the efficiency of the genetic algorithm. The proper genetic algorithm settings depend on the network that is being optimized. By choosing more appropriate genetic algorithm settings for the network in question, this may allow the optimization process to locate the global optimum solution much more quickly.
2. Punch files -- At the end of any individual optimization run, an updated "punch" file is generated that contains the optimal signal settings from the previous run. This punch file could be used as the starting point for subsequent, additional optimization runs. For example, if a particular computer is limited to running 735 generations, it should be possible to perform two or three optimization runs (using punch files from the previous run) having 700 generations each. If this approach is chosen, it is necessary to close and reopen TRANSYT-7F before performing any subsequent optimization runs, in order to restore memory on the computer.
3. Extended run -- If this checkbox is selected, an alternative technique is used to launch the TRANSYT simulation engine (T7FWin2.exe) for analyzing each candidate timing plan. This alternative technique tends to require less computer resources, and allows for a higher number of generations in a single optimization run. Preliminary testing has shown that an extended run can allow for nearly triple the number of generations when compared to a normal run. Although this method is better at conserving memory, TRANSYT-7F does not wait for notification that the engine has finished running before it tries to read the output file. As such, a runtime error ("Permission Denied") sometimes occurs in the middle of the optimization run. The "Delay Time Per Simulation (milliseconds)" forces TRANSYT-7F to wait for a fixed amount of time before trying to read the T7FWin2 output files, but there is never any guarantee under this method that the runtime error can always be avoided. Lowering the Delay Time Per Simulation certainly reduces the overall G.A. optimization running time, but increases the probability of encountering the runtime error.

The Extended Run settings may be changed as long as TRANSYT-7F remains running, but are reset to their defaults every time TRANSYT-7F is closed and re-opened.

#### Upper limit on maximum phase times for fully-actuated controllers in CORSIM

This input parameter is only applicable to direct CORSIM optimization. The value of this input parameter is stored into memory on the computer, and is not stored within the TRANSYT-7F input (\*.TIN) file.

Although the Edit > Timing screen allows the user to specify minimum phase times for optimization, there are no inputs for maximum phase times. This is because the summation of green splits can usually be assumed to be equal to the background cycle length. However, CORSIM models fully-actuated signals without knowing their average green times, and without knowing their cycle lengths. CORSIM only knows the minimum and maximum green times for up to eight dual-ring phases per intersection. Therefore, when optimizing the maximum green times for all eight phases, an upper limit for these maximum green times must be assumed.

The "upper limit on maximum phase times" applies to all phases of all fully-actuated intersections being optimized. The maximum phase time is the summation of the maximum green time, the yellow interval time, and the all-red clearance time. Lower values of this setting may allow the program to locate the global optimum solution more quickly. The danger of a low value is that the global optimum solution may sometimes require a high maximum green time, especially when traffic volumes are high.

## Phasing Sequence Optimization

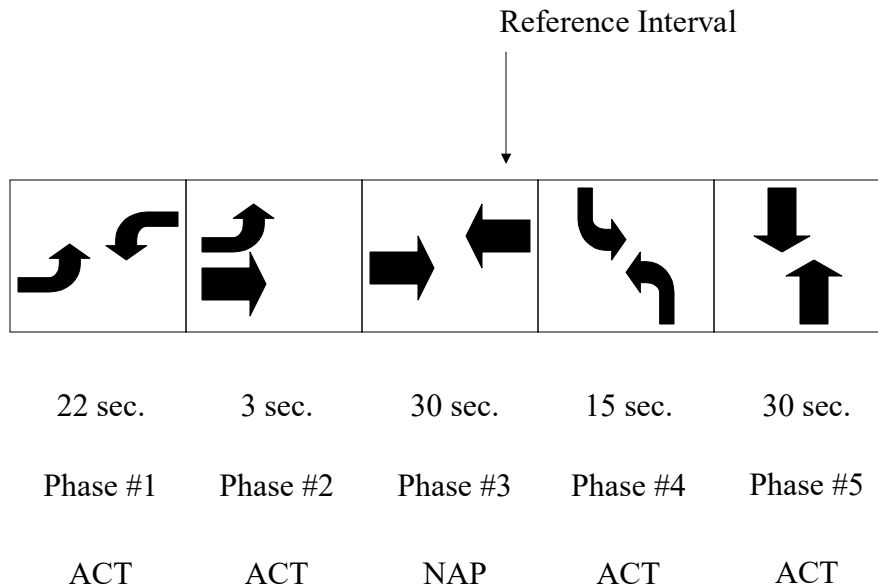
The purpose of phasing sequence optimization is usually to improve progression, although delay can sometimes be reduced as well. TRANSYT-7F performs phasing sequence optimization using the genetic algorithm. TRANSYT-7F is capable of optimizing the phasing sequence on both the major and minor streets. During optimization, TRANSYT-7F examines virtually all feasible phasing sequences including leading and lagging left-turns with and without overlap, lead-lag phasing, and split phasing. The program is designed to be equally effective in optimizing phasing sequences for networks with left-hand driving (e.g., having leading or lagging right-turns). The user may allow full optimization throughout the network, but can also specify restrictions on optimization at any approach of any intersection. On every approach, the user may specify whether they wish to allow optimization, overlap phasing, lead-lag phasing, leading through movements, or the "yellow trap".

### Determining where the phasing sequence can be optimized

The program uses the link numbering scheme (for the data file in question) to ascertain the first "North/South" phase and the first "East/West" phase in the timing plan. The program then uses the link numbering scheme to ascertain the total number of "North/South" phases and "East/West" phases in the timing plan. Exclusive pedestrian phases, and phases that primarily serve "extra" approaches 5 and higher, are preserved (not modified) during phasing sequence optimization.

Assuming right-hand driving, a phase is considered to be a North/South phase when the number of protected through and left-turn links in the North/South direction exceeds the number of protected through and left-turn links in the East/West direction. Similarly, a phase is considered to be an East/West phase when the number of protected through and left-turn links in the East/West direction exceeds the number of protected through and left-turn links in the North/South direction.

For example, the diagram below illustrates that phase #1 is the first East/West phase in the timing plan, and that phase #4 is the first North/South phase in the timing plan. There are 3 East/West phases and 2 North/South phases in the timing plan. The recommended practice is to define the reference interval as the major street through movement's yellow interval, providing consistency with the operation of coordinated semi-actuated controllers.



Note that if the data file does not explicitly request or define a link numbering scheme, the TRANSYT-7F standard scheme is assumed.

The program assumes that if there are no phases in the North/South direction, or only one phase in the North/South direction, the phasing sequence will not be optimized in the North/South direction. The same holds for the East/West direction. In addition, the program assumes that if there are four or more phases in the North/South direction, the phasing sequence will not be optimized in the North/South direction. The same holds for the East/West direction.

Unsignalized intersections, isolated intersections, and intersections absent from the optimization node list (Edit > Optional > Global > Optimization Node List, or record type 2) are considered ineligible for phasing sequence optimization. In this case, isolated intersections are indicated by unchecking the "Coordinated Signal" box on the Edit > Timing screen.

The following is a summarized list of things that can render entire intersection directions (North/South or East/West) ineligible for phasing sequence optimization:

1. Grouped nodes
2. Double cycled nodes
3. Unsignalized intersections
4. Isolated intersections
5. Intersections absent from the optimization node list
6. A number of phases in one direction (North/South or East/West) other than 2 or 3
7. None of the link numbers correspond with the link numbering scheme
8. Multiple green intervals have been coded for a single phase
9. Even though the number of phases in one direction (North/South or East/West) is 2 or 3, the phasing sequence is "non-standard", or illogical.

When a given direction (North/South or East/West) at a given intersection is deemed ineligible for phasing optimization, the characters "----" are written into the output file for that intersection and direction. If the program does not optimize the phasing sequence for any given approach, it simply simulates whatever the user has coded.



The current version of TRANSYT-7F does not support multiple green intervals used in conjunction with phasing sequence optimization. If multiple green intervals have been coded for a single phase, an error message will likely occur. Multiple green intervals are sometimes coded on the Edit > Timing screen, to indicate specific pedestrian or fixed green intervals that are separate from the variable interval, where the variable interval is subject to optimization. An alternative technique for enforcing specific pedestrian or fixed green intervals would be to code the appropriate minimum phase time, also on the Timing screen.

### Associated Links

To optimize the phasing sequence at complex intersections where more than 12 links have been defined (e.g., for a lane-by-lane analysis, or for five or more intersection approaches), it may be necessary to code "associated links". Associated links can be coded on the Edit > Optional > Global > Associated Links screen. Refer to the Global Screen section of the documentation (in chapter 3) for more details.

### Optimization Methodology

Phasing sequence optimization has been implemented in TRANSYT-7F through the use of "phase codes" as follows:

"T"	Both through movements
"L" or "R"	Both left-turn movements (or right-turn movements, if left-hand driving in effect)
"N"	Northbound movements only
"S"	Southbound movements only
"E"	Eastbound movements only
"W"	Westbound movements only

Assuming right-hand drive, the determination of which alternative phasing sequences can be examined (during the optimization process) depends primarily on the existing left-turn and through movement protection. Right-turns potentially affect this logic as well, because "one-directional" phase codes (N,S,E,W) cannot exist if there are any active movements on the opposing approach during the given phase. Although the phasing sequence may be optimized, the existing left-turn and through movement protection cannot be modified by the software.

The following examples involve the assumption of right-hand driving, although the concepts are similar for left-hand driving. First, take note that "split phasing" is defined as any of the following phasing sequences: NS, SN, EW, or WE. The following phasing sequences may be examined if the initial timing plan contains 2 phases in one direction and a) there are no protected left-turns present, or b) there are no protected through movements present, or c) the original timing plan specifies split phasing:

North/South	SN, NS
East/West	WE, EW

The following phasing sequences may be examined if a) the initial timing plan contains 2 or 3 phases in one direction, b) there are two protected left-turns present, and c) there are two protected through movements present:

North/South	TL, LST, SN, LT, STN, NS, LNT, NTS
East/West	TL, LWT, WE, LT, WTE, EW, LET, ETW

Note that the following phasing sequences are never examined, because they are considered inefficient and/or impractical:

North/South	TNL, TSL
-------------	----------

East/West      TEL, TWL

The following phasing sequences may be examined if a) the initial timing plan contains 2 phases in one direction, b) there are two protected through movements present, and c) there is only one protected left-turn present:

North/South    TN, NT, TS, ST  
East/West      TE, ET, TW, WT

The following phasing sequences may be examined if a) the initial timing plan contains 2 phases in one direction, b) there are two protected left-turns present, and c) there is only one protected through movement present:

North/South    LN, NL, LS, SL  
East/West      LE, EL, LW, WL

Note that secondary shared stopline movements cannot be considered protected movements in the logic above, even if they are through movements. This amendment to the logic prevents the separation of primary and secondary movements in any candidate phasing sequence to be examined by the program. Secondary shared stopline movements are coded on the Edit > Optional > Global > Shared Lanes screen (or record type 7). In addition, the phasing sequence optimization logic expects secondary shared turning movements to be represented by "cardinal" link numbers 1 through 12. The Lane Configuration screen accommodates this convention by automatically assigning "extra" link numbers (13 and higher) to any exclusive turn lane whose turning movement can also be performed from an adjacent shared lane.

In the logic above for determining phase codes, a phase code of "T" cannot be interpreted for "right-turn only" approaches (assuming right-hand drive). If the user wishes to optimize the phasing sequence on such approaches where no through movement exists, it might be preferable to define the turning movement as a through movement on the Edit > Traffic screen. By coding the movement this way, phasing sequence can still be optimized, and the movement can still be modeled as a turning movement by coding the proper saturation flow rate and feeding flow information on the Traffic and Feeders screens.

On the phasing sequence optimization screen illustrated below, the user may specify whether they wish to allow phasing optimization, overlap phasing, lead-lag phasing, leading through movements, and the "yellow trap". This screen is accessed from the Edit > Analysis screen. In the background, information specified on this screen is automatically coded onto record type 1X, field 16, whose codes can also be viewed on the Edit > Optional > Timing screen.

7F Phasing Sequence Optimization					
Switch	North/South	Allow Optimization	Allow Overlap	Allow Lead Lag	Allow Yellow
Copy to All	Default Policy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #13	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Copy Here	Intersection #14	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The "Allow Overlap" and "Allow Lead Lag" check boxes are primarily applicable when there are two protected left-turns and two protected through movements present. "Allow Overlap" must be checked in order to consider phasing sequences with overlapping movements, such as LST, STN, LNT, and NTS. Similarly, "Allow Lead Lag" must be checked in order to consider phasing sequences with leading and lagging left-turns, such as SN, STN, NS, and NTS. Finally, "Allow Lead Thrus" must be checked in order to consider phasing sequences with leading through movements, such as TL, TN, and TS.

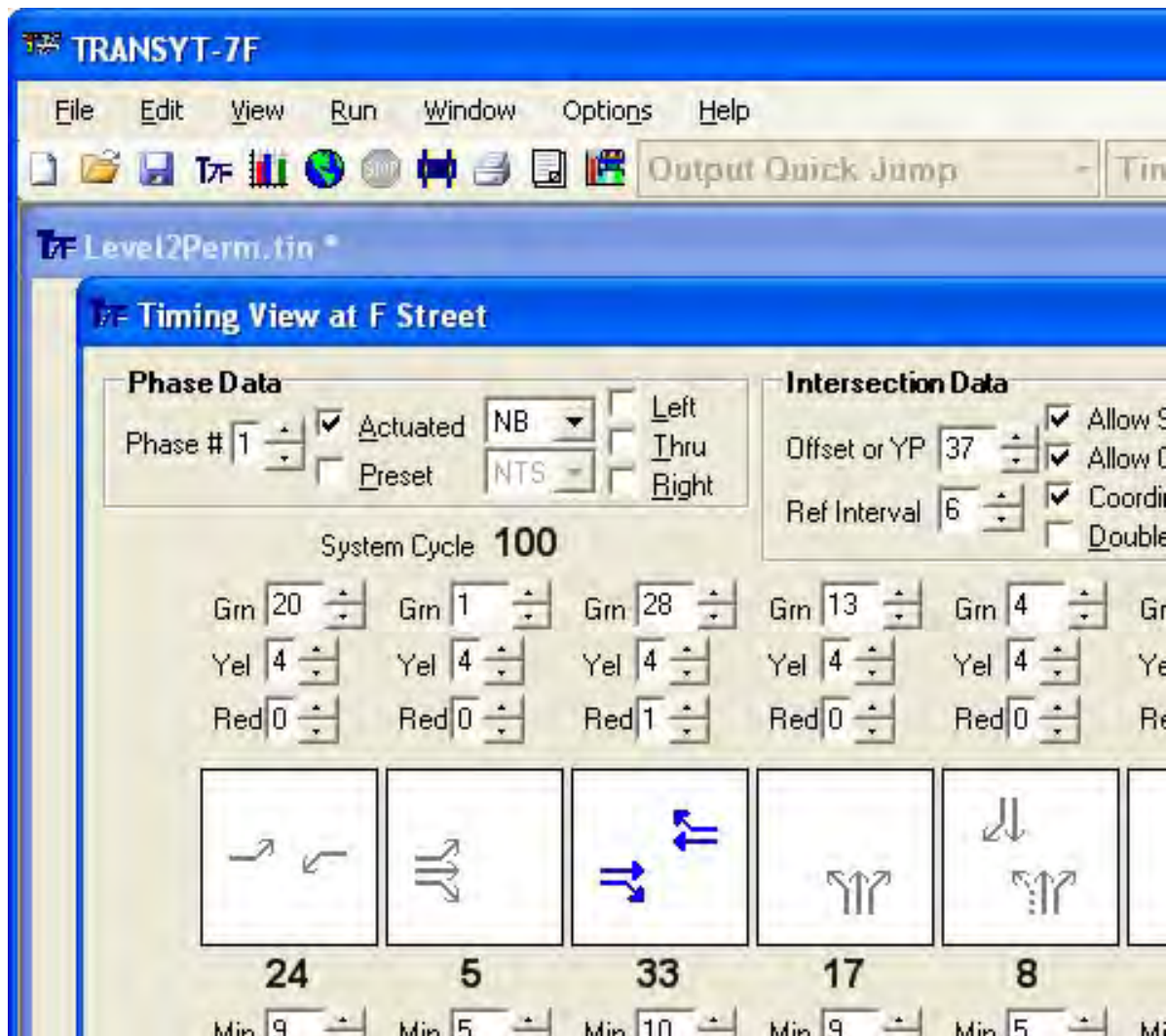
If split optimization has not been requested, TRANSYT-7F automatically sets the initial timing flag (Edit > Analysis screen or record type 1 field 11) to obtain new green splits for each phase. It uses a flag of '2', which retains either the original offsets or the optimized offsets. If simultaneous optimization of cycle length and phasing sequence (without split optimization) is requested, a flag of '1' must be used. Thus unless simultaneous optimization of splits and phasing sequence has been requested, green splits may be modified, but will not be optimized in a great level of detail.

#### Yellow Trap

The yellow trap potentially occurs in conjunction with permitted-protected phasing and lagging left-turns (assuming right-hand drive). For example, suppose there is lagging left-turn phasing in the westbound direction, corresponding with a phasing sequence of "TW". In the first phase, if there are permitted left-turns in both directions, the danger occurs at the end of the first phase. When the signal turns yellow, eastbound left-turning drivers may assume that westbound drivers are receiving the same yellow indication. This assumption encourages the eastbound left-turn drivers to perform dangerous turning movements during the yellow and all-red clearance intervals, while the opposing drivers continue to receive the green display.

The yellow trap also potentially occurs under lead-lag phasing (e.g., "ETW"). For example, the

phasing sequence ("NTS") illustrated in the screen image below contains a potential yellow trap in the north-south direction. Toward the end of phase #5, northbound permitted left-turners unfamiliar with local conditions may be tricked into believing that opposing traffic is about to stop. Because of this, they are tempted to "sneak" through the intersection during the yellow or all-red clearance intervals. However, clearly the southbound through-moving vehicles continue to receive the green ball indication during phase #6, so they would not stop or slow down during the phase #5 clearance interval. This is the source of the yellow trap collision hazard.



If phasing sequence optimization is requested, TRANSYT-7F attempts to identify such dangerous situations in both the initial phasing sequence supplied by the user, and in the optimal timing plan recommended by the program. If the yellow trap situation is identified, an asterisk will appear in the report or output file next to the phasing sequence in question, and a note at the bottom of the report will say: "\*" -- indicates a possible or potential yellow trap". If the user has unchecked the "Allow Yellow Trap" checkbox for a particular intersection, the program should be prevented from examining phasing sequences conducive to the yellow trap.

Because there are practical solutions in the literature for neutralizing the yellow trap, and because drivers in some jurisdictions have been trained to recognize the situation, it is sometimes perfectly

acceptable to allow the yellow trap during the optimization process. For example, in the case of lead-lag phasing, one feasible solution would involve switching to protected-only phasing for the left-turn movement opposite the lagging phase.

## Output

The genetic algorithm examines numerous generations of signal timing (and phasing sequence) candidates in order to determine the fittest individual. The output file listed below contains two columns related to phasing sequence optimization: "NS Phasing" and "EW Phasing". Following optimization, an updated "punch" data file containing the optimal phasing and timing will be available for further analysis.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                      Analysis Time Period:  
Jurisdiction:  
Run Title: Benchmark Blvd  
File Name: Level2Perm.tin

Simulation Engine: TRANSYT

### Genetic Algorithm Input Parameters

```
-----
Crossover Probability (%):      30
Mutation Probability (%):      4
Convergence Threshold (%):     0.01
Maximum Number of Allowable Generations: 20
Actual Number of Optimization Generations: 20
Population Size:               10
Random Number Seed:           7781
Elitist Method:                True
Initial Timing Flag:           Off
Objective Function:             Disutility Index
```

### Optimization Results (Phasing Sequence and Offsets)

```
-----
Node Cycle   Offset NS Phasing EW Phasing
  Init Final Init Final Init Final Init Final
 6 100 100 37 60 LT LNT LET WTE
 7 100 100 56 50 LNT LNT LET WE
 8 100 100 16 44 NS SN --- ---
 9 100 100 35 22 --- --- WT * TW
10 100 100 55 10 LST LNT LET LT
11 100 100 14 87 LST TL LET LWT
12 100 100 33 52 LST LT LET TL
13 100 100 53 74 LST LT LET LET
14 100 100 12 42 --- --- LET EW
15 100 100 32 11 --- --- WT WT
```

\* -- indicates a possible or potential yellow trap

	Control Delay (sec/veh)	Total Stops (%)	Fuel Consumption (gal/hr)	Travel Time (veh-hr/hr)	Performance Index
Initial	48.1	90	725	614	510.50
Final	29.4	66	571	456	338.90

The final solution originated in generation #20

To obtain detailed output, process the punch file:  
Level2Perm.pun

## Multi-Period Optimization

The purpose of multi-period optimization is to develop an effective signal timing plan for handling oversaturated conditions and residual queues. TRANSYT-7F performs multi-period optimization using the genetic algorithm. The program can optimize the splits and offsets of whatever time period is currently selected by the user, or all time periods simultaneously (Options > Time Period > Select All). If only one time period is selected by the user, timing plan transition will be the end result, after the selected time period is optimized independently. If either cycle length or phasing sequence optimization is selected, the program automatically optimizes one timing plan across all time periods. Within the TRANSYT-7F interface, most optimization input settings (e.g., node list, minimum phase times, etc.) are only coded for time period #1, but are applicable to all time periods.

The initial timing flag can be used for multi-period simulation runs, but cannot be used for multi-period optimization runs. Thus, if multi-period cycle length or phasing sequence optimization is requested, optimization of splits becomes mandatory.

When performing multi-period optimization, it is possible to optimize the individual time periods separately, but there are two disadvantages to doing this. First, when individual time periods are optimized separately, only splits and offsets can be optimized. Cycle lengths and phasing sequences cannot be optimized uniquely among time periods. Second, this strategy is likely to result in different sets of splits and offsets recommended for each time period. Frequently, it is preferable to implement one timing plan across all time periods.

A sample output report is listed below. Although the recommended optimal timing plan is sometimes applicable to all time periods, traffic volumes and feeding link flows are expected to fluctuate between time periods. The duration of each individual time period may also vary, although in practice this is less common.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                      Analysis Time Period:  
Jurisdiction:  
Run Title:   Four Node Sample File  
File Name:   FourNode TP.tin

Simulation Engine: TRANSYT

### Genetic Algorithm Input Parameters

-----  
Crossover Probability (%):                      30  
Mutation Probability (%):                      1  
Convergence Threshold (%):                      0.01  
Maximum Number of Allowable Generations:   100  
Actual Number of Optimization Generations:   100  
Population Size:                                  20  
Random Number Seed:                              7781  
Elitist Method:                                      True  
Lower/Original/Upper Cycle Length (sec):   100/100/140  
Objective Function:                                  Disutility Index

## Optimization Results (Cycle Length, Phasing Sequence, Splits, and Offsets)

Node	Cycle	Offset	NS	Phasing	EW	Phasing
	Init	Final	Init	Final	Init	Final
1	100	140	0	0	LT	LT
2	100	140	0	4	LT	LT
3	100	140	0	67	LT	LT
4	100	140	0	117	LT	LT

Node	Split #1	Split #2	Split #3	Split #4	Split #5
	Init	Final	Init	Final	Init
1	15	12	40	56	15
2	15	12	40	60	15
3	15	17	40	62	15
4	15	13	40	65	15

Control	Total	Fuel	Travel	Performance
Delay	Stops	Consumption	Time	Index
(sec/veh)	(%)	(gal/hr)	(veh-hr/hr)	

Time Period	Initial	Final	Initial	Final	Initial	Final
Time Period 1 (optimized)						
Initial	109	101	412	396	313.00	
Final	66.8	104	330	281	233.50	
Time Period 2 (optimized)						
Initial	164	103	521	548	423.70	
Final	76.1	106	351	311	257.60	
Time Period 3 (optimized)						
Initial	216	116	513	566	445.40	
Final	55.7	96	228	186	153.60	
Time Period 4 (optimized)						
Initial	159	115	413	431	345.80	
Final	42.2	88	195	152	121.40	
Time Horizon Summary						
Initial	162	109	464	485	381.97	
Final	60.2	98	276	232	191.53	

The final solution originated in generation #100  
 To obtain detailed output, process the punch file:  
 FourNode TP.pun

Note that the actuated estimation model cannot process multi-period data files. In the case of actuated control, it is expected that only one array of signal settings would be implemented in the field across all time periods, but that the fluctuating volumes in conjunction with actuation would naturally cause fluctuating average phase times. The actuated estimation and multi-period optimization procedures in TRANSYT-7F can be used to determine what phase times would occur in the field. For determining force-offs after split optimization, the time period resulting in the highest actuated degrees of saturation should be analyzed (in a single-period actuated estimation analysis, or using the analytical UC Berkeley procedure [26]), because average phase times are most sensitive to the force-offs under these conditions. Once determined, these same force-offs would likely also produce the lower phase times from the less congested time periods, because these phases would usually gap-out anyway. For more discussion on determining force-offs and maximum greens, refer to the Traffic-Actuated Control section of the documentation.

For a tutorial example of multi-period optimization, refer to the documentation section entitled "Multi-Period Analysis Examples".

## Uncoordinated Optimization

The Edit > Timing and Edit > Optional > Timing screens allow the user to specify "Allow Split Optimization" for each individual intersection, but there is no option to "Allow Cycle Length Optimization" for a subset of intersections. If cycle length optimization is requested, then cycle lengths must be optimized at all intersections on the optimization node list, regardless of whether or not they are coordinated. The cycle length range from the Edit > Analysis screen (or record type 1) applies to all signalized intersections on the optimization node list. Note that the initial timing flag cannot be used to generate initial timings for uncoordinated signals.

The nature of uncoordinated optimization depends on the following user inputs:

1. Whether or not cycle length optimization has been requested for the overall network
2. Whether or not split optimization has been requested for the overall network
3. Whether or not split optimization has been requested for a given intersection
4. Whether a given intersection is present on the optimization node list

Assuming that optimization of cycle length and splits has been requested for the overall network, the "truth table" below illustrates various types of uncoordinated optimization runs:

				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
absent from optimization node list				N	Y	N	N	Y	Y	N	Y	N	Y	N	N	Y	Y	N	Y
uncoordinated network				N	N	Y	N	Y	N	Y	Y	N	N	Y	N	Y	N	Y	Y
uncoordinated intersection				N	N	N	Y	N	Y	Y	Y	N	N	N	Y	N	Y	Y	Y
allow split optimization at this node				N	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
1	illegal input request; an error message will result																		
2	illegal input request; an error message will result																		
3	maintain the original timing plan at this intersection																		
4	maintain the original timing plan at this intersection																		
5	maintain the original timing plan at this intersection																		
6	maintain the original timing plan at this intersection																		
7	maintain the original timing plan at this intersection																		
8	maintain the original timing plan at this intersection																		
9	optimize the network cycle length; optimize splits locally																		
10	illegal input request; an error message will result																		
11	optimize cycle length and splits locally																		
12	optimize cycle length and splits locally																		
13	maintain the original timing plan at this intersection																		
14	maintain the original timing plan at this intersection																		
15	optimize cycle length and splits locally																		
16	maintain the original timing plan at this intersection																		

If optimization of cycle length without splits is requested for the overall network, the initial timing model is used to estimate splits for each candidate cycle length. For this scenario, the truth table below illustrates various types of uncoordinated optimization runs:



				1	2	3	4	5	6	7	8						
absent from optimization node list				N	Y	N	N	Y	Y	N	Y						
uncoordinated network				N	N	Y	N	Y	N	Y	Y						
uncoordinated intersection				N	N	N	Y	N	Y	Y	Y						
1	optimize the network cycle length; splits are estimated using the initial timing model																
2	illegal input request; an error message will result																
3	illegal input request; an error message will result																
4	maintain the original timing plan at this intersection																
5	illegal input request; an error message will result																
6	maintain the original timing plan at this intersection																
7	illegal input request; an error message will result																
8	illegal input request; an error message will result																

## Direct CORSIM Optimization

TRANSYT-7F offers "direct" CORSIM optimization (also known as "microscopic" optimization) of cycle length, splits, and offsets, using the genetic algorithm. With direct CORSIM optimization, TRANSYT-7F applies the genetic algorithm to supply timing plan candidates, and CORSIM evaluates them through its own simulation. The end result is a new copy of the original CORSIM input (\*.TRF) file containing the optimized timing plan.

Regarding macroscopic, TRANSYT-based optimization, TRANSYT-7F is also capable of "indirect" CORSIM optimization. With indirect CORSIM optimization, TRF files are first imported to create a TRANSYT-7F input (\*.TIN) file. Following macroscopic (TRANSYT-based) optimization, the timing is then exported back to the TRF file for subsequent CORSIM simulation. For more information on indirect CORSIM optimization, or timing export from TRANSYT to CORSIM, refer to the CORSIM Processing Methodology section of the documentation.

With other programs and processes, optimization is performed using a separate underlying analysis engine (different from CORSIM). Naturally, results tend to be different when the optimized timings must then be exported to CORSIM, or evaluated by CORSIM. With direct CORSIM optimization, there is no need to for the user to export timings to CORSIM, because the optimization is done directly. Thus the optimized results reported by TRANSYT-7F will be identical to any results obtained when simulating the updated TRF file within TSIS.

Because direct CORSIM optimization is accomplished via genetic algorithm, this ensures that the user's original timing plan will be improved upon, assuming that a better timing plan exists. In addition, direct CORSIM optimization can be easily applied by anyone, because there is no learning curve. Simply launch TRANSYT-7F, load the TRF file, select an optimization objective function, and click on "Run". This alone achieves a better CORSIM timing plan than any other program or process. Once the user gains experience with choosing non-default genetic algorithm parameters (e.g., mutation probability, population size, etc.), the optimization process may become even more efficient and effective.

All pre-timed and actuated intersections in CORSIM can be optimized (cycle length, splits, and offsets/yield points) directly. In the case of semi-actuated control, force-offs are explicitly optimized, and their associated permissive periods are handled automatically. Force-offs and permissive periods designed by the program are also compliant with any minimum phase times coded by the user. In the case of fully-actuated control, maximum green times are explicitly optimized. Multi-period CORSIM networks can also be optimized as a function of "cumulative NETSIM results" in the CORSIM legacy report.

The current version of TRANSYT-7F does not support direct CORSIM optimization of phasing sequence, or freeway ramp meters. Phasing sequence optimization could be accomplished by

macroscopic (TRANSYT-based) optimization, and then one-touch CORSIM animation could be used to evaluate the optimized phasing sequence in CORSIM. If an optimized phasing sequence is coded or exported into a TRF file, the overall timing plan could then be further improved via direct CORSIM optimization of cycle length, splits, and offsets.

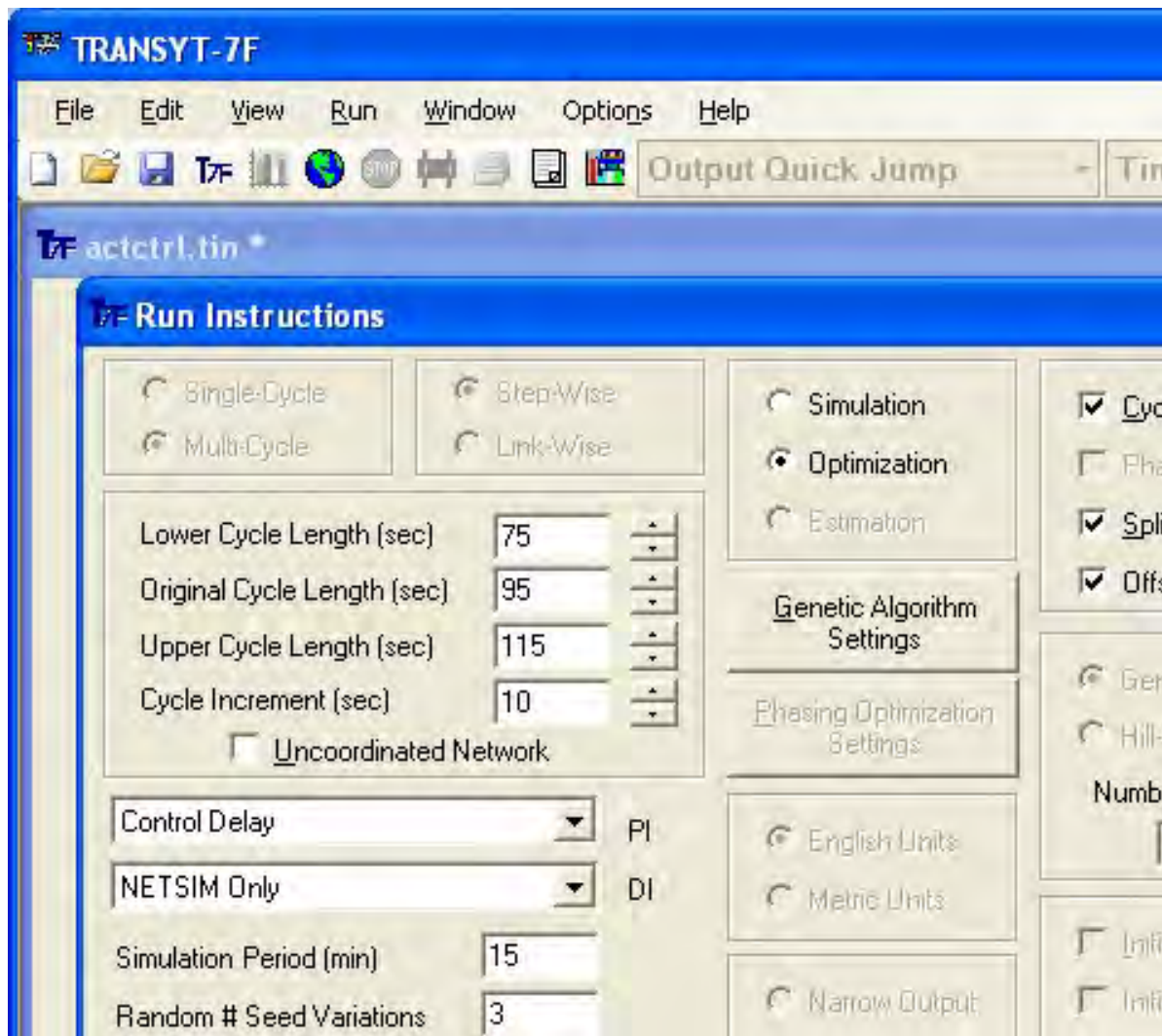
When a TRF file is initially opened in TRANSYT-7F, a TIN file is automatically generated in order to store the associated processing and optimization settings. The processing settings that are generated (i.e., duration of simulation, initialization time, random number seed variations, volume multiplier) are discussed in the CORSIM Processing Methodology section. However, these processing settings also affect optimization runs, whereas the optimization settings do not affect processing-only runs without optimization. The following optimization settings affect direct CORSIM optimization:

- 2) ● optimization node list (Edit > Optional > Global > Optimization Node List, or record type 2)
- genetic algorithm parameters (Edit > Analysis screen, or record type 5X)
- cycle length range (Edit > Analysis screen, or record type 1)
- optimization objective function (Edit > Analysis screen, or record type 5X)
- signal settings to be optimized globally (Edit > Analysis screen, or record type 5X)
- minimum phase times (Edit > Timing screen, or record type 2Y)
- allow split optimization at this node (Edit > Timing screen, or record type 1X)
- allow offset optimization at this node (Edit > Timing screen, or record type 1X)
- coordinated signal (Edit > Timing screen, or record type 1X)

For fully-actuated signals, there are a few additional input settings that affect direct CORSIM optimization. Regarding minimum phase times, it becomes necessary to provide this information for up to eight dual-ring phases. Because of this, the offset and reference interval are used to represent the phase time and minimum phase time, respectively, for phase #8. The summary text report for direct CORSIM optimization also uses the offset and reference interval to represent the phase time and minimum phase time. On the Options > Genetic Algorithm screen, there is also an "upper limit on maximum phase times" setting, which can be customized by the user. Refer to the Genetic Algorithm Options section of the documentation for more details. To hold timing constant at fully-actuated intersections during optimization, the user could either omit such intersections from the optimization node list, or choose (Options > Preferences) to not import fully-actuated intersections from CORSIM.

For grouped nodes, the Original Cycle Length affects the "difference of offset", which must be maintained between grouped nodes. Refer to the section on Grouped Nodes for more details.

The screen image below illustrates some of the direct CORSIM optimization settings mentioned above.



Other data such as traffic volume and feeding link information are also generated within the TIN file, but have no effect on direct CORSIM optimization. This extra data can be reviewed within the TRANSYT-7F Edit screens if desired, but only becomes relevant if the user elects to perform a companion macroscopic analysis. For more information on conversion to macroscopic analysis, refer to the CORSIM Processing Methodology section of the documentation.

Optimization objective functions are contained in the Performance Index (PI) combo box. Objective functions can either be selected for the surface street network (NETSIM Only), or for the entire CORSIM network (Network Wide). No initial timing model is available for NETSIM, so the initial timing flag checkboxes are disabled.

On the Edit > Timing screen, the Allow Split Optimization and Allow Offset Optimization checkboxes are somewhat self-explanatory, but the Coordinated Signal checkbox should only be deactivated for isolated intersections. For more details on the effect this would have on the genetic optimization process, refer to the Uncoordinated Optimization section. Minimum phase times are also shown on the Edit > Timing screen. By default, when the TIN file is initially generated, minimum phase times are set equal to the yellow and all-red clearance times plus 5 seconds. For overlap phases at a pre-timed signal, minimum phase times are set equal to the yellow and all-red clearance times plus 1 second. The user may elect to increase these minimum phase times to either speed up the optimization

process, or to accommodate pedestrian requirements.

Pre-timed interval durations in CORSIM are not allowed to exceed 120 seconds. Although uncommon, this limitation is sometimes circumvented by coding two consecutive phases with identical movements, or by defining an actuated controller with "max recall" (to act like a pre-timed controller). When performing pre-timed controller optimization with cycle lengths above 120 seconds, minimum phase times must be coded carefully. For example, consider a four-phase signal with minimum phase times of 10 seconds apiece, and an upper cycle length of 160 seconds. If the first three phases were assigned durations of 10 seconds, the fourth phase would become 130 seconds, which would abort the optimization process (producing "fatal error 2626" in CORSIM). In this case, minimum phase times of 15 seconds apiece would have avoided this problem. Internally, for two-phase pre-timed signals, low minimum phase times will be automatically reset to a value of (UpperCycle - 120).

Following direct CORSIM optimization, an updated "punch" TRF file will be available for further analysis. This punch file will be called "Genetic.trf", and will contain the optimized signal settings.

For a tutorial example of direct CORSIM optimization, refer to the documentation section entitled "CORSIM Processing and Optimization Examples".

A sample output listing from direct CORSIM optimization is illustrated below.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                     Analysis Time Period:  
Jurisdiction:  
Run Title:   Run Title  
File Name:   actctrl.tin

Simulation Engine: CORSIM 6

Random # Seed Variations:            1  
Volume Multiplier:                    1  
Simulation Period (sec):              900  
Initialization Period (sec):          900

#### Genetic Algorithm Input Parameters

Crossover Probability (%):            30  
Mutation Probability (%):            1  
Convergence Threshold (%):           0.01  
Maximum Number of Allowable Generations: 10  
Actual Number of Optimization Generations: 10  
Population Size:                      10  
Random Number Seed:                  7781  
Elitist Method:                        True  
Upper Limit on Maximum Phase Times (sec): 70  
Objective Function:                    Control Delay (NETSIM only)

#### Optimization Results (Splits and Offsets)

```
-----  
Node Cycle    Offset  NS Phasing EW Phasing  
Init Final Init Final Init Final Init Final  
190 95 90    0 61 --- --- --- ---  
350 --- --- --- --- --- --- --- ---  
151 90 90    18 62 --- --- --- ---  
152 90 90    85 12 --- --- --- ---  
255 90 90    0 56 --- --- --- ---
```

450 70 56      --- --- --- ---

Node	Split #1	Split #2	Split #3	Split #4	Split #5	Split #6	Split #7
	Init	Final	Init	Final	Init	Final	Init
190	36	66	45	13	14	11	
350	90	--					
151	40	21	35	48	15	21	
152	35	57	40	18	15	15	
255	30	31	45	25	15	34	
450	50	34	20	22			

	<-----NETSIM Approach Links----->								Network-Wide	
	Control	Total	Stop	Queue	Vehicle	Percent			Delay	
	Delay	Delay	Delay	Delay	Trips	Stops	Speed		Time	Speed
	------(sec/veh)-----				(%)	(mph)	(veh-hr)		(mph)	
Initial	8.8	10.7	7.4	7.8	4094	37.5	15.7		13.1	17.7
Final	7.8	9.8	6.6	7.0	4117	36.7	15.7		12.1	18.4

The final solution originated in generation #7  
 To obtain detailed output, process the punch file:  
 Genetic.trf



# Chapter 6: Tutorial Examples

## Creating a Brand-New Input Data File

These two examples show how to create brand-new input data files for a traditional (TRANSYT-based) macroscopic analysis. The primary discussion will assume English units, and driving on the right-hand side of the road, but it is also possible to run through these examples using metric units and left-hand driving. To double-check the input data after these exercises, refer to the FourNode.tin and Sample #3.tin files, which are available in the TRANSYT-7F Example folder.

These examples assume that HCS and CORSIM input files are not being imported to create the new dataset. For tutorial examples of HCS and CORSIM file import, in which TRANSYT-7F input files are generated or expanded automatically to reflect the imported data, refer to the following documentation sections: Creating a New Input Data File by Importing HCS Files, and CORSIM Processing and Optimization Examples.

### Four Node Example

The hypothetical conditions will involve four signalized intersections in a square "box" formation. To simplify the process, this example will assume that each intersection approach contains an exclusive left-turn pocket, and a shared through and right-turn lane. (For left-hand drive, each intersection approach should contain an exclusive right-turn pocket, and a shared through and left-turn lane.) The turn pockets are capable of storing up to 12 vehicles. Further, each turning movement serves 45 vehicles per hour, and each through movement serves 360 vehicles per hour (vph), with a peak hour factor (PHF) of 0.90.

To create the new file, simply go to File > New (or click on the New File icon) in the TRANSYT-7F graphical user interface (GUI). Because the data will be mostly the same at all four intersections, the easiest way to create the network is to code traffic and timing data at one intersection, and then make three copies of that intersection. The only difference between the four intersections will be which approach directions have an upstream signal, but all link connections will be coded after the traffic and timing data have been completed. The screen image below illustrates some appropriate settings for beginning the new file. For more information on the screen below, see the New TRANSYT Input File section of the documentation.

Field	Value	Field	Value
File Name	Sample.tin	Default Traffic Volume (vph)	
Run Title	Chapter 6 Sample File	Default Peak Hour Factor	
Link Number Scheme	TRANSYT-7F Standard	Default Saturation Flow Rate (vph)	
Number of Intersections	1	Default Link Length (ft or m)	
Number of Routes	0	Default Free-Flow Speed (mph or km/h)	
Cycle Length (sec)	100	Default Number of Phases	
Duration of Analysis (min)	15	Default Yellow Time (sec)	
Number of Time Periods	1	Default All-Red Time (sec)	
<input checked="" type="radio"/> English Units		Default Gap Setting (sec)	
<input checked="" type="radio"/> Right-Hand Drive		Default Detector Length (ft or m)	

Once the "OK" button has been clicked to create the new data file, it is now possible to code the remaining data. First, it is helpful to set the "intersection name". By going to the Edit > Optional > Intersection > Names screen, it is possible to type in the cross street names associated with each intersection. Subsequently, these intersection names will show up on the edit screens, and sometimes in the output file reports. This part of the data entry process is optional.

To begin with, it is best to code the traffic volume and lane configuration data. It is important to do this now so that later, when the link connections are coded, "automatic flow balancing" can be performed. To make it easier to view all four approaches simultaneously, it is preferable to select the highest possible screen resolution (i.e., right-click on the Windows desktop and select Properties > Settings > Screen resolution > More). Then, proceed to "Edit > Lanes > Intersection View". This loads the Lane Configuration screen. Notice that the intersection name is displayed on the top of the screen below. Since the default traffic volume of 45 vph was coded in the original New File screen, it is only necessary to code the through movement volume of 360 vph, and the turn pocket storage of 12 vehicles, at all four approaches.



**7F Lane Configuration at University and 13th**

103

104

Lane #   ☒ Lane Group  
☐ Lane-by-Lane

Distance (ft)

	Left	Thru	Right
Volume (vph)	<input type="text" value="45"/>	<input type="text" value="360"/>	<input type="text" value="45"/>
Peak Hour Factor	<input type="text" value="0.90"/>	<input type="text" value="0.90"/>	<input type="text" value="0.90"/>
Turn Bay Storage (veh)	<input type="text" value="12"/>		

106

105

Lane #   ☒ Lane Group  
☐ Lane-by-Lane

Distance (ft)

	Left	Thru	Right
Volume (vph)	<input type="text" value="45"/>	<input type="text" value="360"/>	<input type="text" value="45"/>
Peak Hour Factor	<input type="text" value="0.90"/>	<input type="text" value="0.90"/>	<input type="text" value="0.90"/>
Turn Bay Storage (veh)	<input type="text" value="12"/>		

Lane #   ☒ Lane Group  
☐ Lane-by-Lane

Distance (ft)

	Left	Thru
Volume (vph)	<input type="text" value="45"/>	<input type="text" value="360"/>
Peak Hour Factor	<input type="text" value="0.90"/>	<input type="text" value="0.90"/>
Turn Bay Storage (veh)	<input type="text" value="12"/>	

Node

Lane #   ☒ Lane Group  
☐ Lane-by-Lane

Distance (ft)

	Left	Thru	Right
Volume (vph)	<input type="text" value="45"/>	<input type="text" value="360"/>	<input type="text" value="45"/>

Add...

This would be a good time to click on File > Save, to prevent any data from being lost by accident. The next step is to go to the Edit > Traffic screen, to review volumes and other traffic-related data. For this example, the current values for link length, mid-block source volume, start-up lost time, and extension of effective green time do not need to be modified. Therefore, the Traffic screen can be bypassed. Notice that volumes on the Traffic screen are listed as 50 vph for turns and 400 vph for thru movements, even though they were coded as 45 vph and 360 vph on the Lanes screen. This is because the Lanes screen contains the "unadjusted" volumes, and the Traffic screen contains the "adjusted" volumes. In this example, even though each thru movement serves 360 vehicles in one hour, the peak hour factor of 0.90 implies that a flow rate of 400 vph exists during the peak 15-minute period.

**TRANSYT-7F**

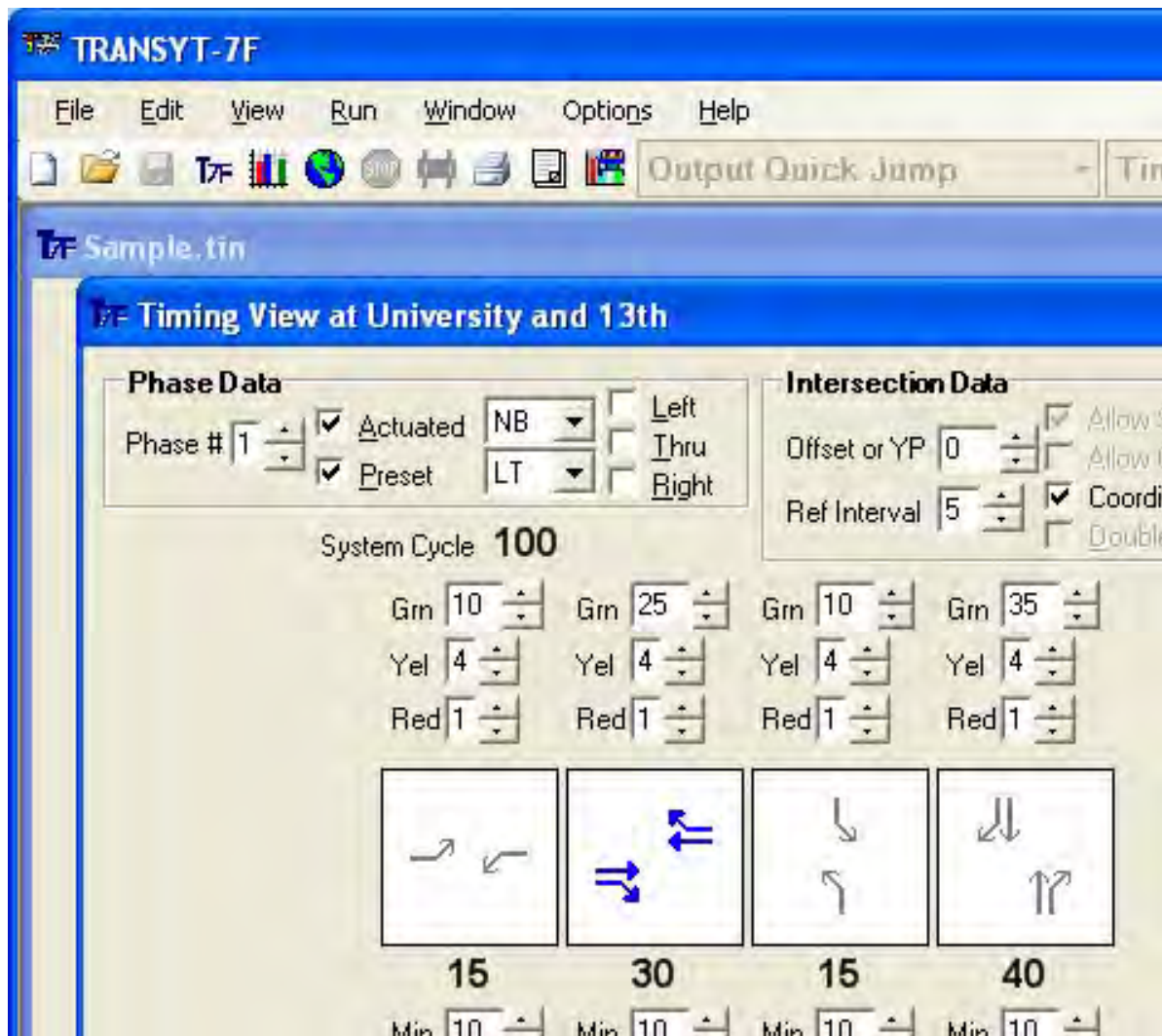
File Edit View Run Window Options Help

Output Quick Jump - Tim

**Traffic at University and 13th**

Eastbound			Westbound			Northbound			Southbound	
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru
106	105	111	108	107	112	102	101	109	104	103
Total Volume (vph)										
50	400	50	50	400	50	50	400	50	50	400
Saturation Flow Rate (vphg)										
1900	1900	0	1900	1900	0	1900	1900	0	1900	1900
Link Length (ft)										
1320	1320	1320	1320	1320	1320	1320	1320	1320	1320	1320
Mid Block Source Volume (vph)										
0	0	0	0	0	0	0	0	0	0	0
Start-Up Lost Time (sec)										
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Extension of Effective Green Time (sec)										

After finishing the traffic volume and lane configuration data, it is then time to specify timing data on the Edit > Timing screen. The screen image below illustrates the timing plan that will be coded for this example. The timing plan is "semi-actuated", meaning that only the major street through movement phase is coded as non-actuated. The phasing sequence specifies leading turn arrows in both (NS and EW) directions, with no overlaps. Most phasing sequences can be coded fairly quickly by using the "preset phasing" feature. Simply click on the "Preset" checkbox, and then select the desired phasing sequence in the adjacent combo box to the right. This checkbox is deactivated by default, because changing the preset phasing by mistake can quickly disrupt the coding of any existing timing plan. Phasing sequences available for selection will be a function of the lane groups previously defined on the Lane Configuration screen.



Other settings within the "Phase Data" frame also affect the preset phasing feature. For example, the first phase of the preset phasing sequence is indicated by the "Phase #" field, the approach direction is indicated by the "NB/SB/EB/WB" combo box, and the actuation is indicated by the "Actuated" checkbox. For this example, the phase codes "LT" are used to designate leading left-turns with no overlaps. (For left-hand drive, the phase codes "RT" are used to designate leading right-turns with no overlaps.) Refer to the Phasing Sequence Optimization section of the documentation for a description of the available phase codes.

A few other settings on this screen are also worth mentioning. First, the "Allow Offset Optimization" checkbox is automatically turned off for the master controller, if one has been defined (Edit > Optional > Global > Network Parameters). This is because it is typically desirable for the offset to remain zero at the master controller, for ease of reference. Second, the "Reference Interval" is coded as interval #5, because this is the major street through movement yellow interval. It is typically desirable for the reference interval to be consistent with the "yield point" of actuated controllers, such that the optimized offsets will not require translation prior to field implementation. Third, the minor street through movement has a split of 40 seconds, which is greater than the major street through movement split of 30 seconds. This type of green time allocation is fairly unusual, because usually the major street gets most of the green time. Finally, the minimum phase times have been set to 10 seconds apiece. As a rule of thumb, the minimum phase time for non-overlap phases should be at least five seconds greater than the

sum of yellow and all-red clearance times, just to satisfy driver expectation. Beyond that, the minimum phase times could be optionally increased to support pedestrian crossing requirements, or to facilitate more efficient signal timing optimization.

Note that when the initial timing flag is activated on the Edit > Analysis screen, any user-coded green times are ignored, and the program generates its own green times for analysis. It is even acceptable to leave the green times blank (coded as zero seconds) when the initial timing flag is used. However, for this example, it is assumed that the user wishes to either evaluate the manually coded timing plan, or wishes to use the manually coded timing plan as a starting point within the optimization process.

At this stage, saturation flow rates for each movement could be coded on the "Satflow View" screen (Edit > Satflow). In order to view the full contents of this screen, the screen resolution must be set to at least 1024x768 or higher. Thru movement saturation flow rates are already set correctly at 1900 vehicles per lane per hour of green (vplphg), but the left-turn saturation flow rates still need to be modified. Right-turn saturation flow rates cannot be coded because they are shared with the thru movement. For left-hand drive, the right-turn saturation flow rates still require modification, and left-turn saturation flow rates cannot be coded because they are shared with the thru movement. Therefore, click the "Apply" button to automatically code adjusted saturation flow rates for each lane group.

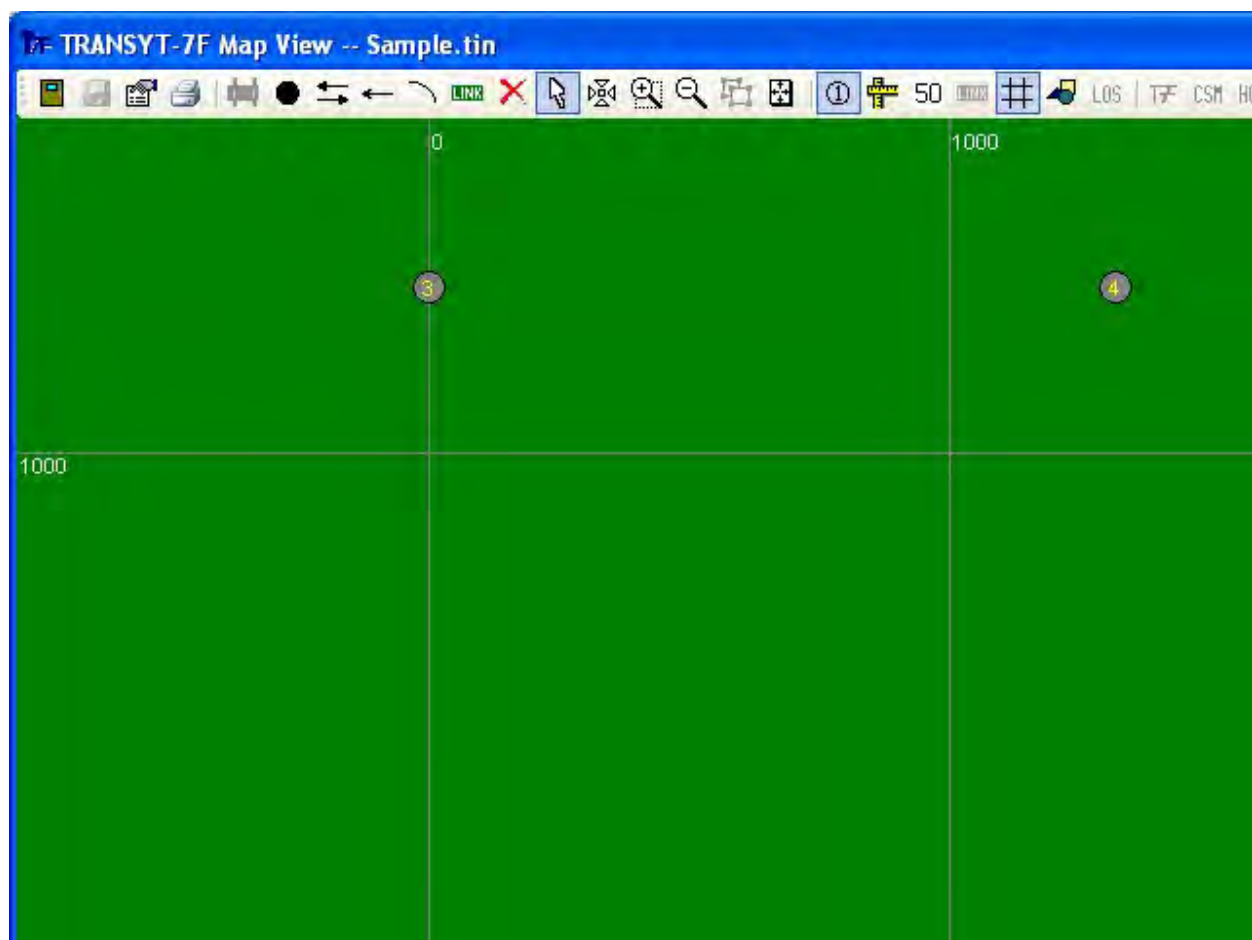


**7F Satflow View at University and 13th**

Eastbound			Westbound			Northbound			Southbound	
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru
106	105	111	108	107	112	102	101	109	104	103
Base Saturation Flow Rate (vplphg)										
1900	1900		1900	1900		1900	1900		1900	1900
Number of Lanes										
1	1		1	1		1	1		1	1
Lane Width (ft)										
12	12		12	12		12	12		12	12
Heavy Vehicles (%)										
0	0		0	0		0	0		0	0
Grade (%)										
0	0		0	0		0	0		0	0
Parking Prohibited										
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Parking Maneuvers Per Hour										
0	0		0	0		0	0		0	0
Number of Buses Stopping										
0	0		0	0		0	0		0	0
Area Type										
Other										
Lane Utilization										
Egl	Egl	Egl	Egl	Egl	Egl	Egl	Egl	Egl	Egl	Egl
Type of Turning Movement										
LT	TH	RT	LT	TH	RT	LT	TH	RT	LT	TH
Ped-Bike influence on Left-Turns										
Noi	Noi	Noi	Noi	Noi	Noi	Noi	Noi	Noi	Noi	Noi
Ped-Bike influence on Right-Turns										
Noi	Noi	Noi	Noi	Noi	Noi	Noi	Noi	Noi	Noi	Noi

Now that all of the intersection-specific data have been coded, it is possible to make copies of this intersection inside the Map View. The Map View is launched by either clicking on the "earth" icon, or by pressing F3, or by selecting Edit > Map. Because no map has yet been drawn, the node appears in the middle of the screen. To make copies of this intersection, simply right-click on the node, and select "Copy

and Paste". The user can then create and/or drag the new nodes to any desired location on the screen. Since the link lengths will be 1320 feet for this simple example, it is best to drop nodes at coordinates such as (0,0), (0,1320), (1320,0), and (1320,1320), which might require zooming out with the magnifying glass. For metric units, since the link lengths will be 402 meters for this example, it is best to drop nodes at coordinates such as (0,0), (0,402), (402,0), and (402,402). The screen image below illustrates the configuration of the four-node network.

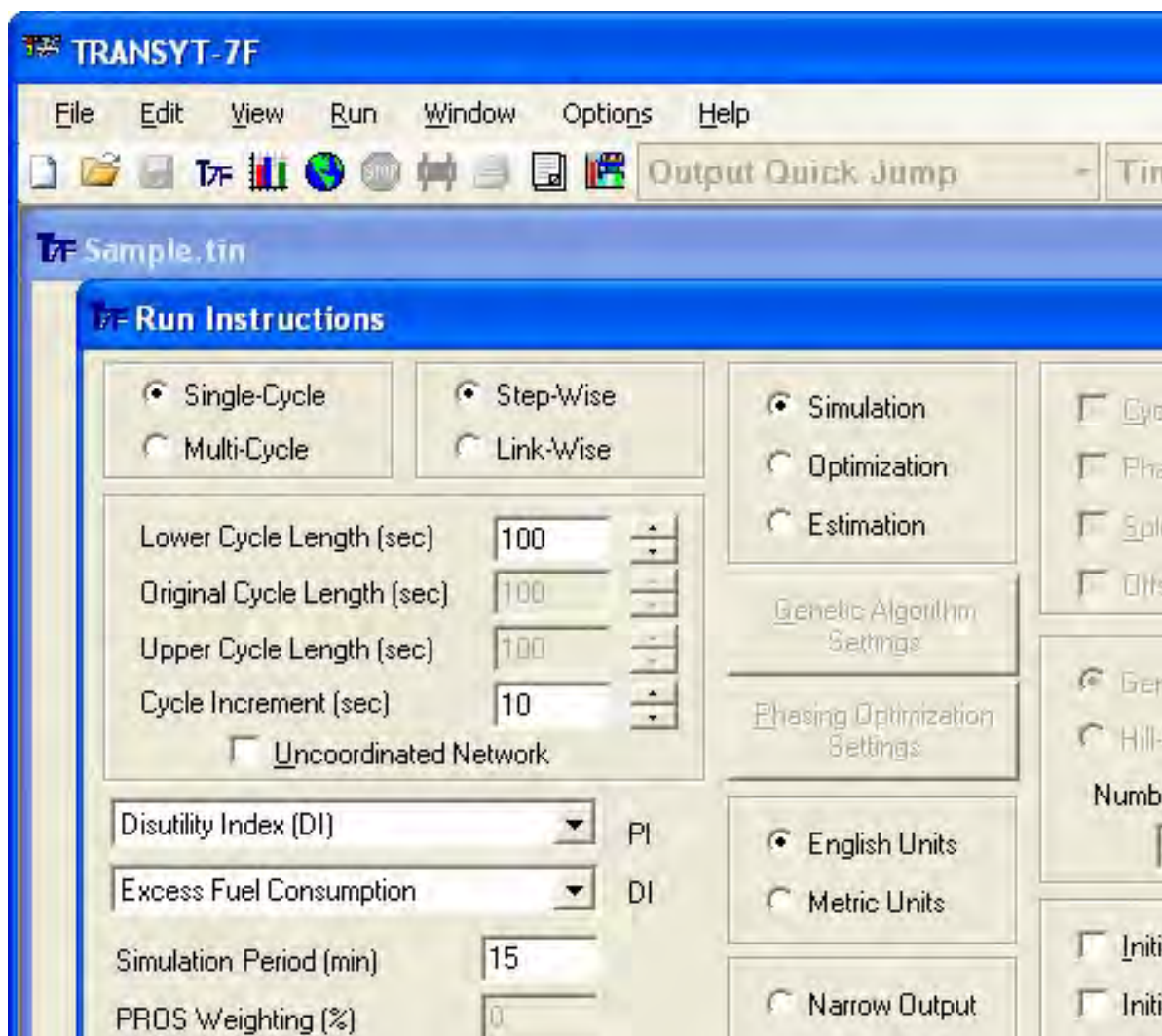


For this exercise, the user should click on the "Properties" icon, to confirm that the map orientation is "North". This will indicate that the northbound direction travels straight up the computer screen. Before dragging the nodes to their desired location, it is sometimes helpful to import a bitmap background image using the Properties icon, and scale the bitmap using the "Bitmap Scaling" icon. A sample bitmap file ("Sample.bmp") is available within the Example folder, so this could be loaded as an extra exercise.

Clearly, it is difficult to drag and drop nodes at the exact coordinates (0,0), (0,1320), (1320,0), and (1320,1320). Since these node coordinates do not affect the link lengths, and do not affect TRANSYT model results, it is not important for the node coordinates to be precise. However, for a perfectly scaled link-node diagram, the user can "hardcode" the node coordinates in one of two ways. By drawing a mouse "rubber band" around one or more nodes, the Properties icon can be used to code the exact X and Y coordinates of the selected node(s). The Properties dialog screen can also be loaded by pressing the Z key. Otherwise the record type editor could be used to code the node coordinates on record type 70.

Now that the node locations have been established, link connections can be drawn in the Map View. Prior to drawing link connections, confirm that the default cruise speed is set at 35 mph (56 kmph) in the Options > Cruise Speed screen. Then click on the "Two-Way Link" icon, and draw four links between the four nodes. This will invoke automatic flow balancing on the Feeders screen. Refer to the Automatic Flow Balancing section of the documentation for more information on its associated assumptions and methodology. If desired, the user could load the Feeders screen at this time, to review or modify the feeding flow data. Also at this time, the user could code link curvature and/or link names on the map, if desired.

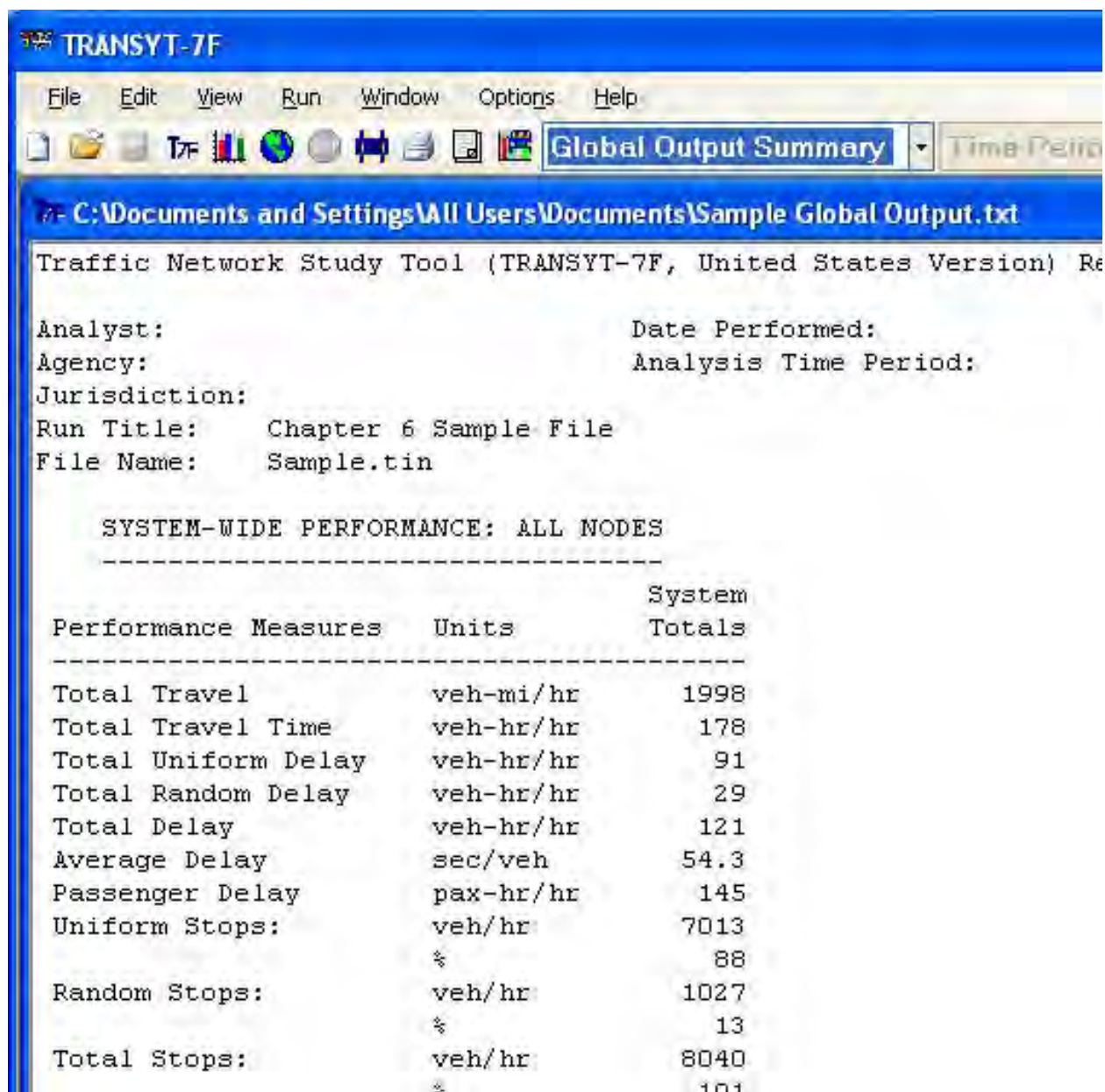
At this stage, it is possible to proceed to the Edit > Analysis screen (illustrated below), to specify run instructions. The screen below indicates single-cycle step-wise simulation, with the typical analysis period of 15 minutes. The initial timing flags should be deactivated so that the coded timing plan can be explicitly simulated. The disutility index is selected as the objective function for any upcoming optimization runs. Disutility index values are also reported for simulation-only runs, but this measure of effectiveness is more useful and meaningful in the context of optimization.



At this time, after saving the data on the Analysis screen, it should be possible to run TRANSYT



without getting any fatal errors. One can run the model either by clicking on the blue "T7F" icon, or by pressing <Ctrl>-R. Numerical results from the model are illustrated in the screen image below. Due to round-off errors caused by metric unit conversion, results illustrated below will be slightly different for metric data files (e.g., 54.4 seconds of delay per vehicle, instead of 54.3). Any error messages encountered at this stage are most likely due to coding errors or omissions on the Lanes, Traffic, Timing, or Feeders screens. Users who can't figure out how to eliminate a program error message can e-mail their data file to McTrans technical support. Refer to the Technical Support section of the documentation for more details.



The screen image above displays "Global Output Summary" in the Output Quick Jump combo box, at the top of the GUI. The Output Quick Jump can also be used to jump to other parts of the legacy report, such as the link performance output tables. Certain parts of the output file, such as the signal timing tables, route statistics, and text-based diagrams, will only be available on the Output Quick Jump



list if they were requested by the input file. Certain other "optimization" output files, created by genetic algorithm optimization, hill-climb cycle length optimization, and actuated estimation, do not support (and usually do not require) Output Quick Jump capabilities.

To search more slowly within a particular section of the output file, it is best to place the mouse cursor directly within the output file window, and navigate the report using the keyboard. It is also possible to switch focus from the Output Quick Jump to the report text by using the <ENTER> key. The Time Period Quick Jump, directly to the right of the Output Quick Jump, is only activated when the input data file contains multiple time periods. The Time Period Quick Jump controls the input data and output data that will be displayed by the GUI.

A sample output summary report for node #1 of the FourNode network is displayed below. Several similar input and output summary text reports are available on the Output Quick Jump. Note that the appearance and content of these summary reports are affected by the chosen format (narrow or wide) of the legacy report, and that overall shared lane results are only listed for the primary (through) link.

**TRANSYT-7F**

File Edit View Run Window Options Help

Node #1 Time Period

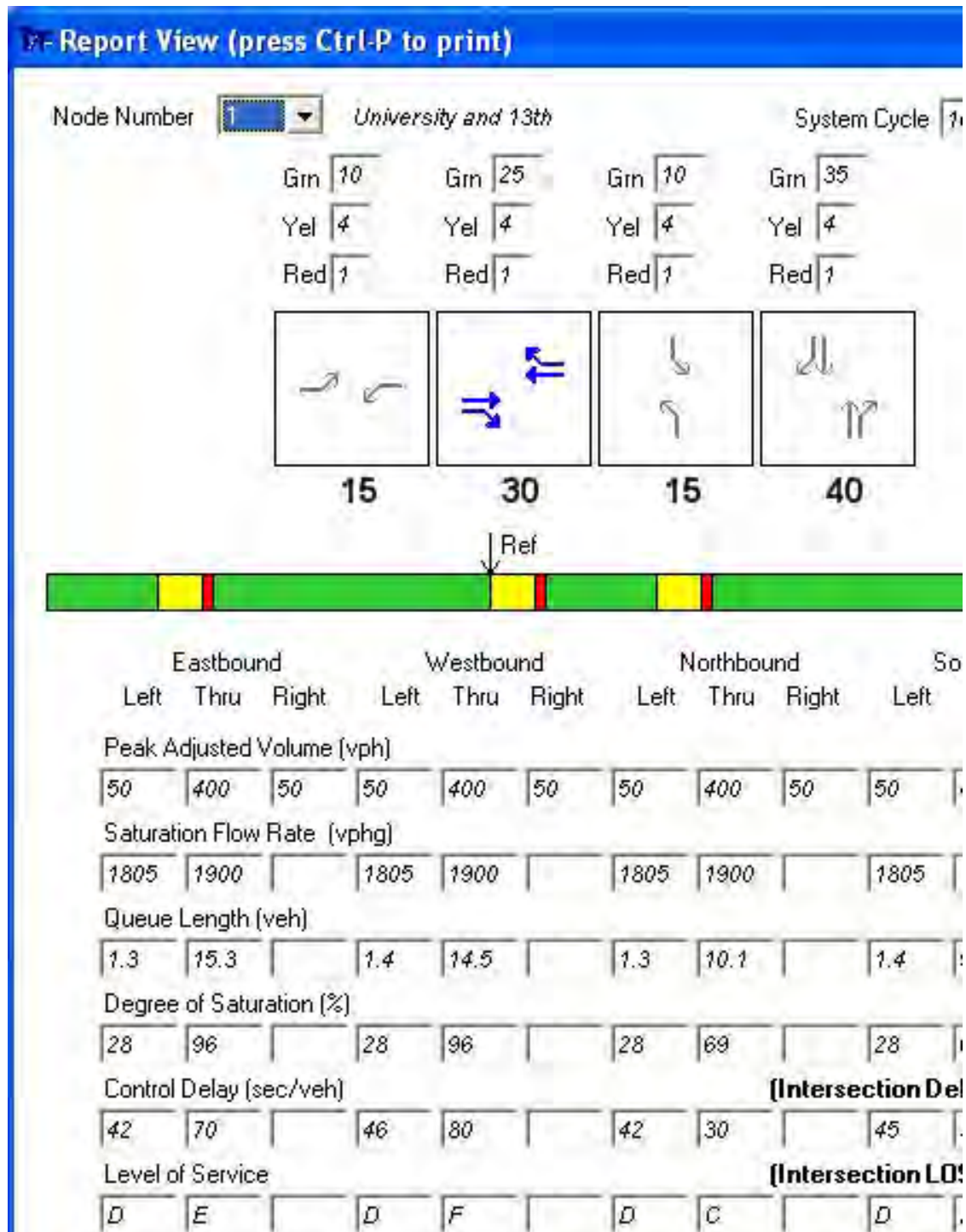
C:\Documents and Settings\All Users\Documents\Sample Node Output.txt

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release

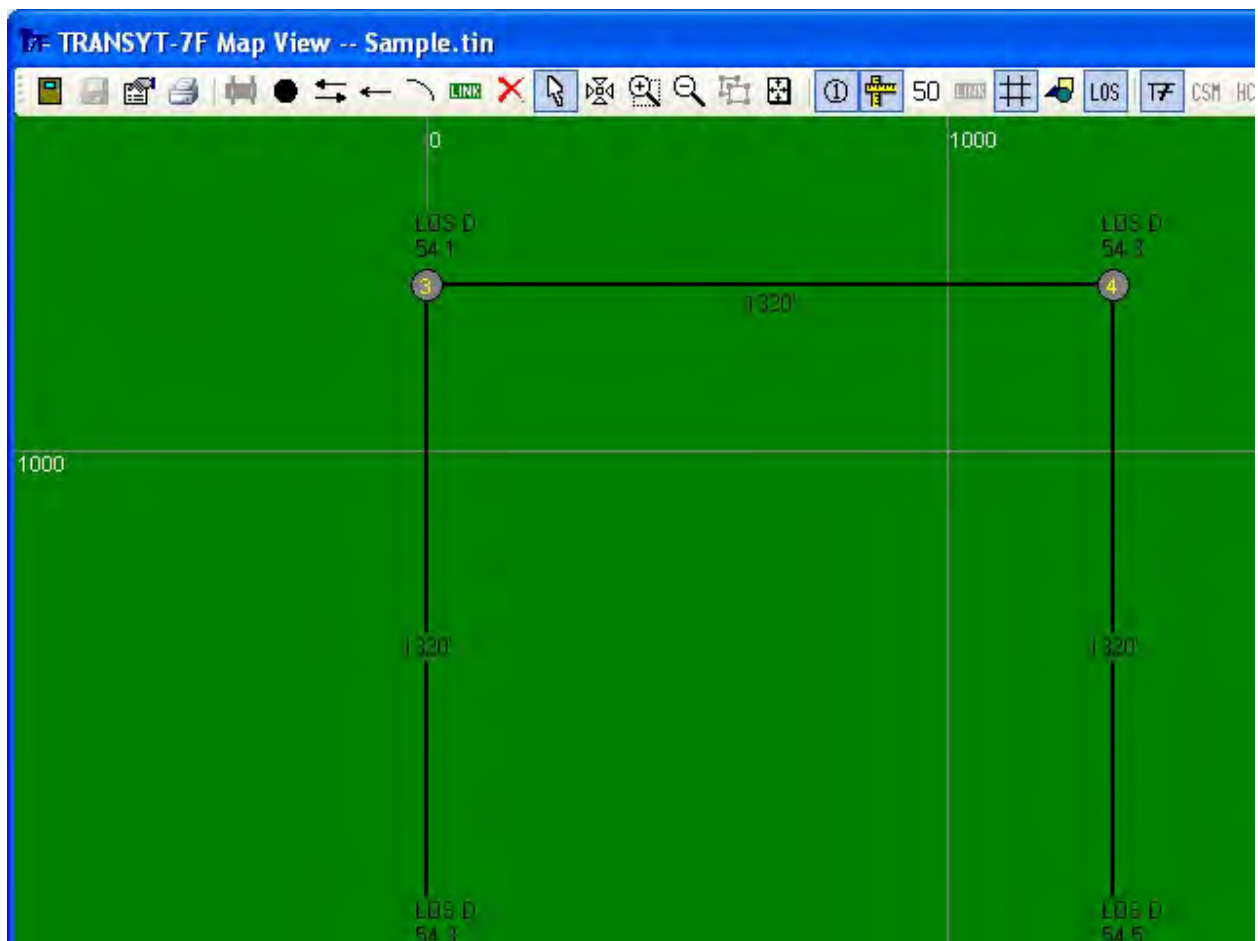
Analyst: Date Performed:  
 Agency: Analysis Time Period:  
 Jurisdiction:  
 Run Title: Chapter 6 Sample File  
 File Name: Sample.tin  
 Node Number: 1  
 Intersection: University and 13th

	Eastbound			Westbound			Northbound		
	LT	TH	RT	LT	TH	RT	LT	TH	RT
Output Flow (vph)	50	450		50	449		50	450	
Degree of Sat. (%)	28	96		28	96		28	69	
Tot. Travel (veh-mi)	12	112		12	112		12	112	
Tot. TravTime (veh-h)	0.9	12.0		1.0	13.3		0.9	7.0	0
Avg. TravTime (sec/v)	68	96		72	106		68	56	
Unif. Delay (veh-h)	0.5	4.6		0.6	7.6		0.5	3.4	0
Rand. Delay (veh-h)	0.0	4.2		0.0	2.4		0.0	0.3	0
Total Delay (veh-h)	0.5	8.8		0.6	10.0		0.5	3.8	0
Avg. Delay (sec/v)	42	70		46	80		42	30	
Unif. Stops (vph)	44	427		50	439		44	368	
Unif. Stops (%)	90	95		101	98		90	82	1
Rand. Stops (vph)	1	140		0	82		1	18	
Rand. Stops (%)	3	32		1	19		3	4	
Total Stops (vph)	46	568		50	521		46	386	

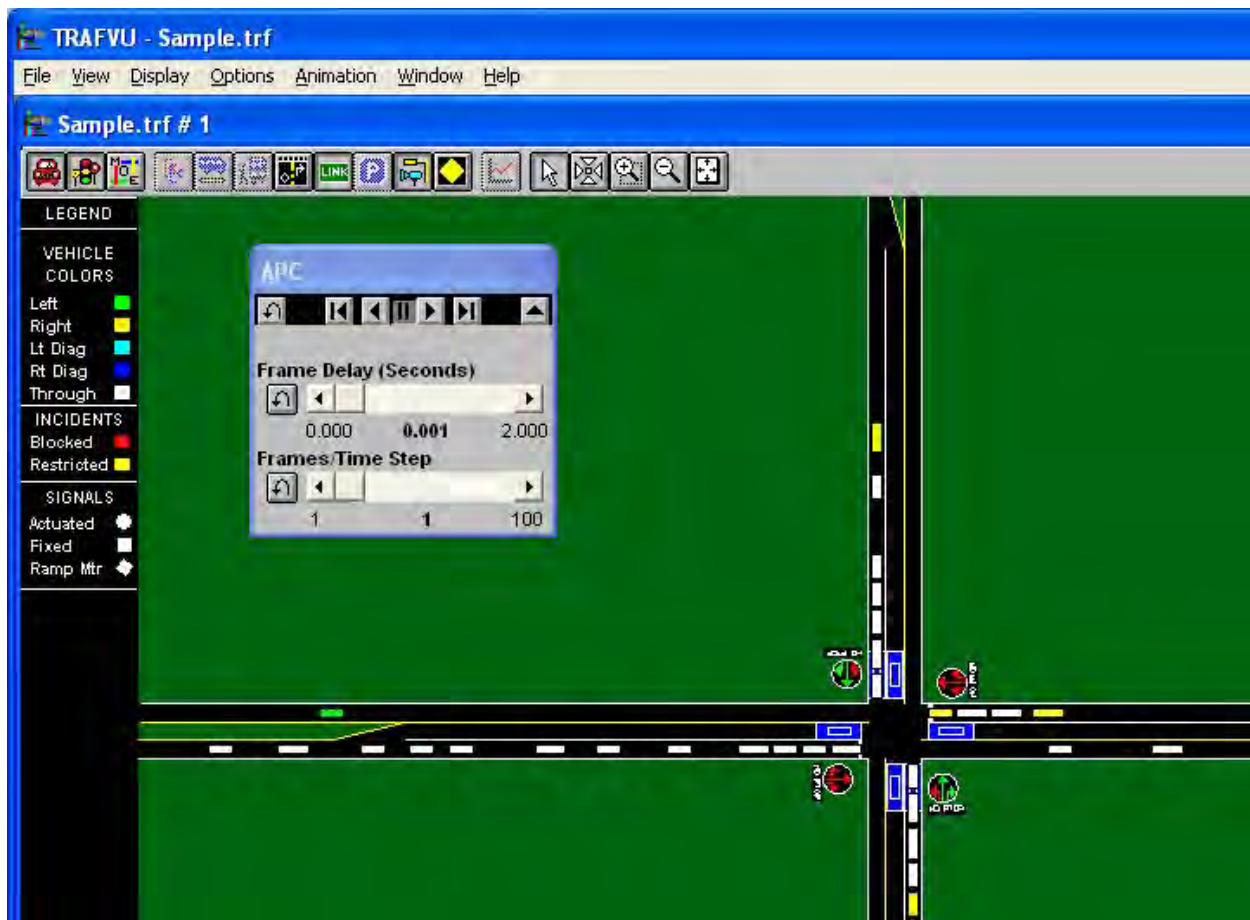
The one-page formatted report illustrated below can be obtained by selecting View > Report, or by clicking on the "View Formatted Report" icon. Screen resolution should be set to at least 1024x768 to view and print the entire report.



The screen image below illustrates delay and level of service superimposed on the Map View.

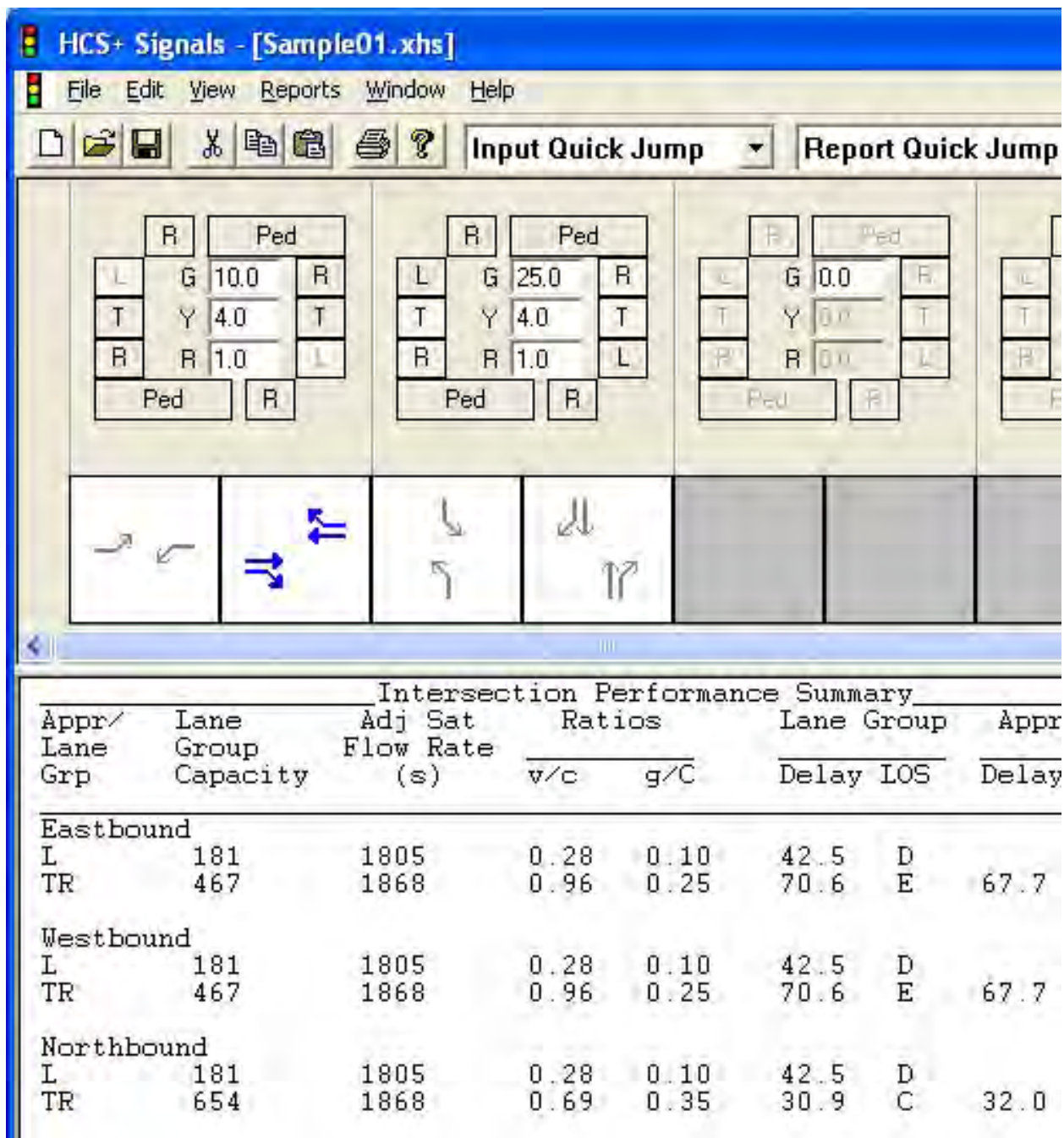


The screen image below illustrates one-touch CORSIM animation of the four-node network. This feature can be accessed by selecting "File > CORSIM > View animation for this network", or by clicking on the "CORSIM Animation" icon. In TRAFVU, the user can zoom in and zoom out to view different aspects of the network. The traffic and timing data should look similar to those within TRANSYT-7F.



The screen image below illustrates one-touch HCS analysis of node #1. This feature can be accessed in the Map View by right-clicking on a node and selecting "HCS > View HCS results for this node". The traffic and timing data should look similar to those within TRANSYT-7F.



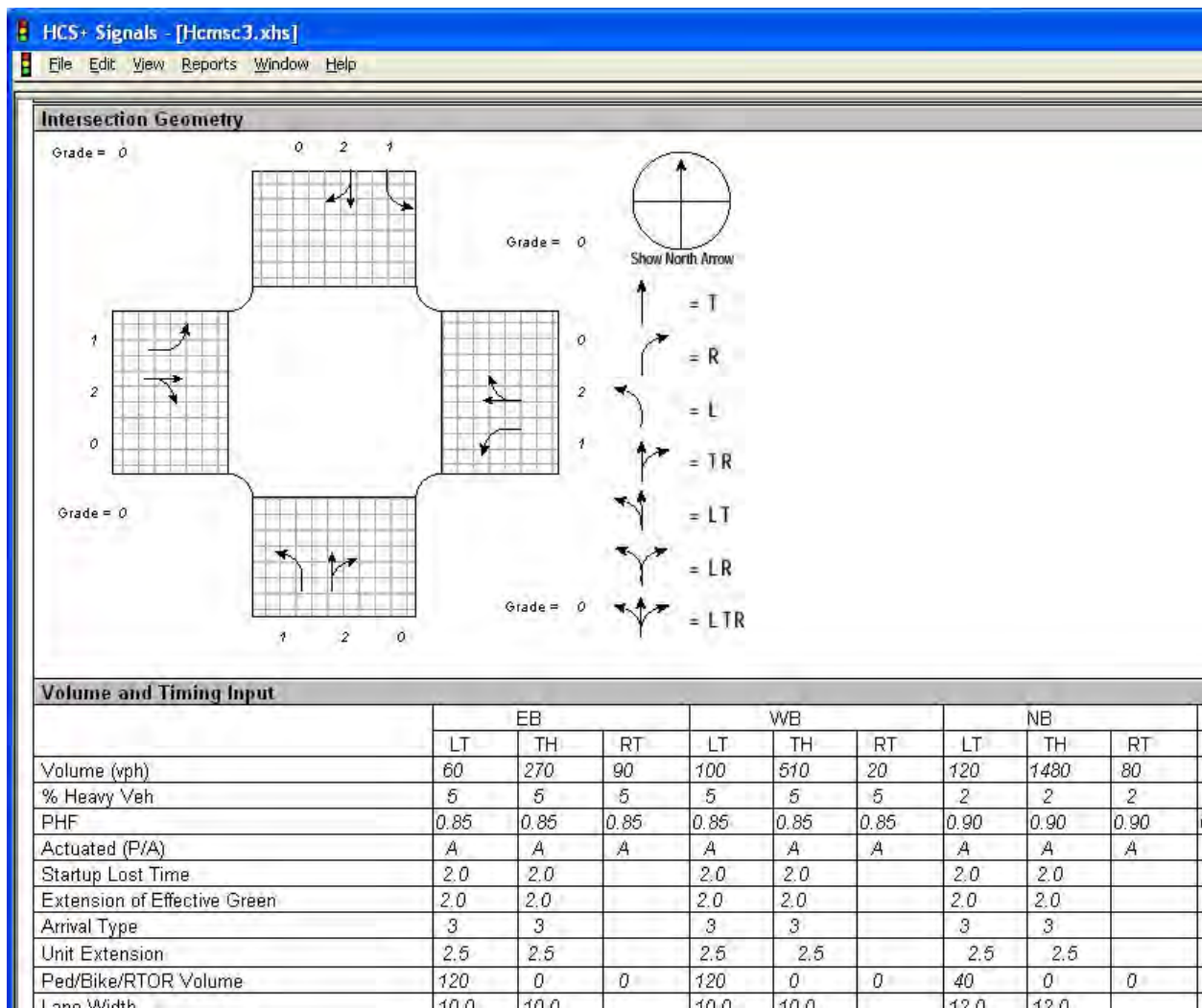


### HCM Sample Calculation #3 Example

This example involves re-creation of the HCM sample calculation #3 conditions within TRANSYT-7F. In this example, the intersection of Fifth Avenue (NB/SB) and Twelfth Street (EB/WB) is a major central business district (CBD) junction of two urban streets. If desired, this intersection could be added to the Four Node example network described earlier, but the discussion below will describe how to create a brand-new file to analyze this intersection as isolated. Also keep in mind that this TRANSYT-7F file could be generated automatically by loading the appropriate HCS example file. Conditions for this intersection are listed within chapter 16 of the Highway Capacity Manual [12], and are summarized below.

- Twelfth Street HV = 5 percent, Fifth Avenue HV = 2 percent
- Twelfth Street PHF = 0.85, Twelfth Street parking, 5 maneuvers/h
- Fifth Avenue PHF = 0.90, Twelfth Street pedestrian volume = 120 p/h
- Actuated signal, Fifth Avenue pedestrian volume = 40 p/h
- Yellow = 4.0 s, Movement lost time = 4 s
- Level terrain, Arrival Type 3
- 3.0-m lane widths for EB/WB, No bicycles
- Pedestrian signals exist, No buses
- 12-ft lane widths for NB/SB
- Assume base saturation flow rate = 1,900 pc/h/ln
- No overlaps in signal phasing
- 90.0-s cycle length, with green times given
- Assume a unit extension of 2.5 s for all phases

Network geometry, volume, and timing inputs are illustrated by the HCS formatted HTML report below. Each approach contains two through lanes with the rightmost lane shared, and an exclusive left-turn lane. The phasing sequence consists of "green ball" phasing in both directions, but the north/south direction has leading left-turn ("green arrow") protection.



For assistance with determining the adjusted saturation flow rates, an HCS screen image is provided below. In this screen, saturation flow rates must be computed for the primary phase (protected left-turn phase) as well as the secondary phase (permitted left-turn phase).

**HCS+ Signals - [Hcmsg3.xhs]**

File Edit View Reports Window Help

Eastbound			Westbound			Northbound			Southbound		
L	TR		L	TR		L	TR		L	TR	
Ideal Saturation Flow Rate (pcphgpl)											
1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Lane Width (ft), All 12.0											
10.0	10.0	12.0	10.0	10.0	12.0	12.0	12.0	12.0	12.0	12.0	12.0
Percent Heavy Vehicles (%), All 0											
5	5	5	5	5	5	2	2	2	2	2	2
Percent Grade (%)											
0			0			0			0		
Parking Maneuvers per hour											
<input type="checkbox"/> Yes 20 <input checked="" type="checkbox"/> Yes 5			<input type="checkbox"/> Yes 20 <input checked="" type="checkbox"/> Yes 5			<input type="checkbox"/> Yes 20 <input type="checkbox"/> Yes 20			<input type="checkbox"/> Yes 20		
Bus Stops per hour											
0			0			0			0		
Highest Single Lane Volume in Lane Group (vph). <input type="checkbox"/> Field data available.											
Conflicting Bikes and Pedestrians per hour											
Bikes 0 Peds 120			Bikes 0 Peds 120			Bikes 0 Peds 40			Bikes 0		
Percent Turns Using Protected Phase											
0.0			0.0			0.0			0.0		
Adjustment Factors											
$f_W$ 0.933 0.933			0.933 0.933			1.000 1.000			1.000		
$f_{HV}$ 0.952 0.952			0.952 0.952			0.980 0.980			0.980		
$f_g$ 1.000			1.000			1.000			1.000		
$f_p$ 1.000 0.938			1.000 0.938			1.000 1.000			1.000		
$f_{bb}$ 1.000 1.000			1.000 1.000			1.000 1.000			1.000		
$f_a$ 0.900			0.900			0.900			0.900		
$f_{LU}$ 1.000 0.952			1.000 0.952			1.000 0.952			1.000		
$f_{LT}$ 0.208 1.000 Pri.			0.347 1.000 Pri.			0.950 1.000 Pri.			0.950		
$f_{LT}$ Sec.			Sec.			0.199 Sec.			0.073		

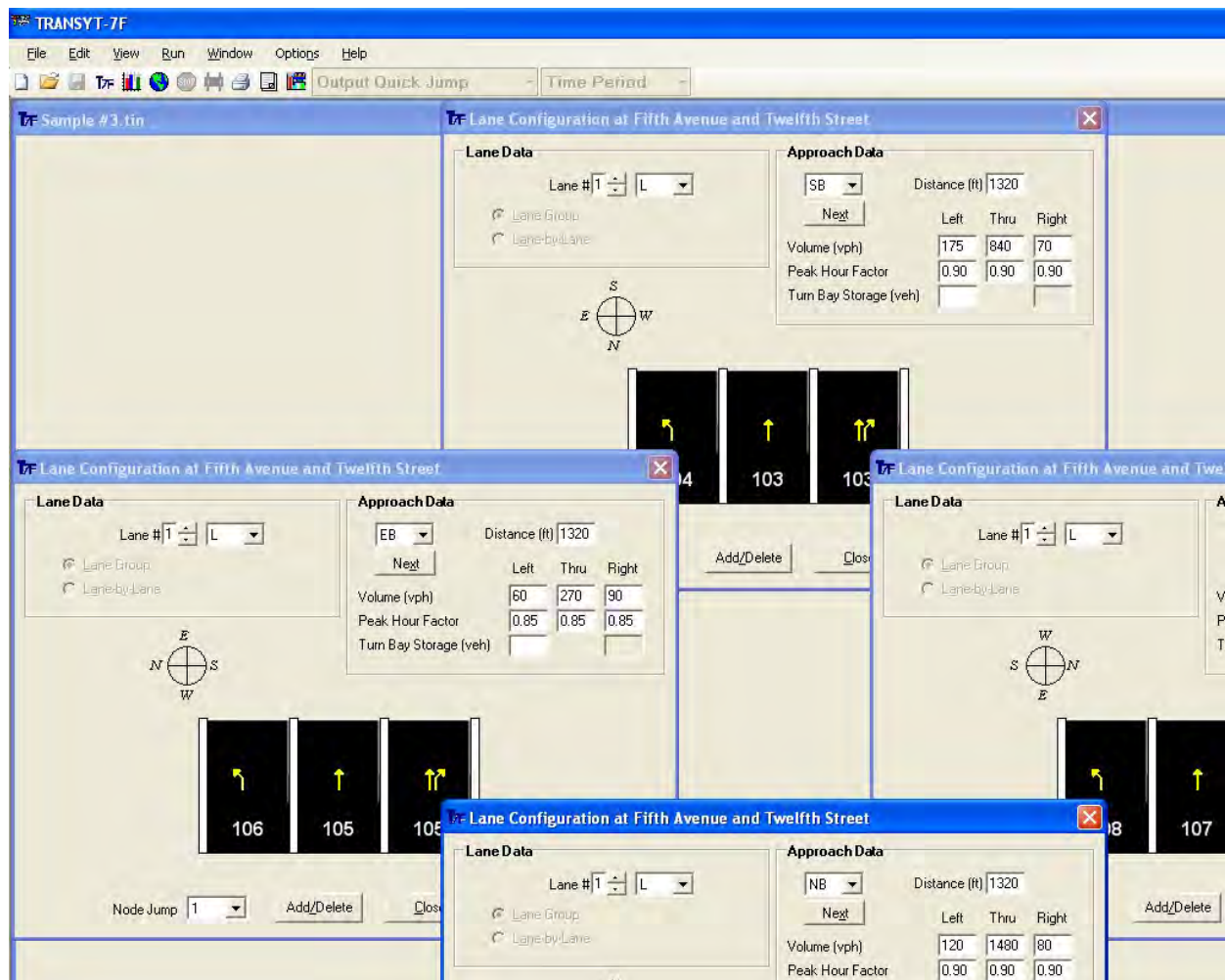


With these available inputs, the TRANSYT-7F input file can now be coded. To create the new file, go to File > New (or click on the New File icon) in the TRANSYT-7F GUI. The screen image below illustrates some appropriate settings for beginning the new file.

**New TRANSYT Input File**

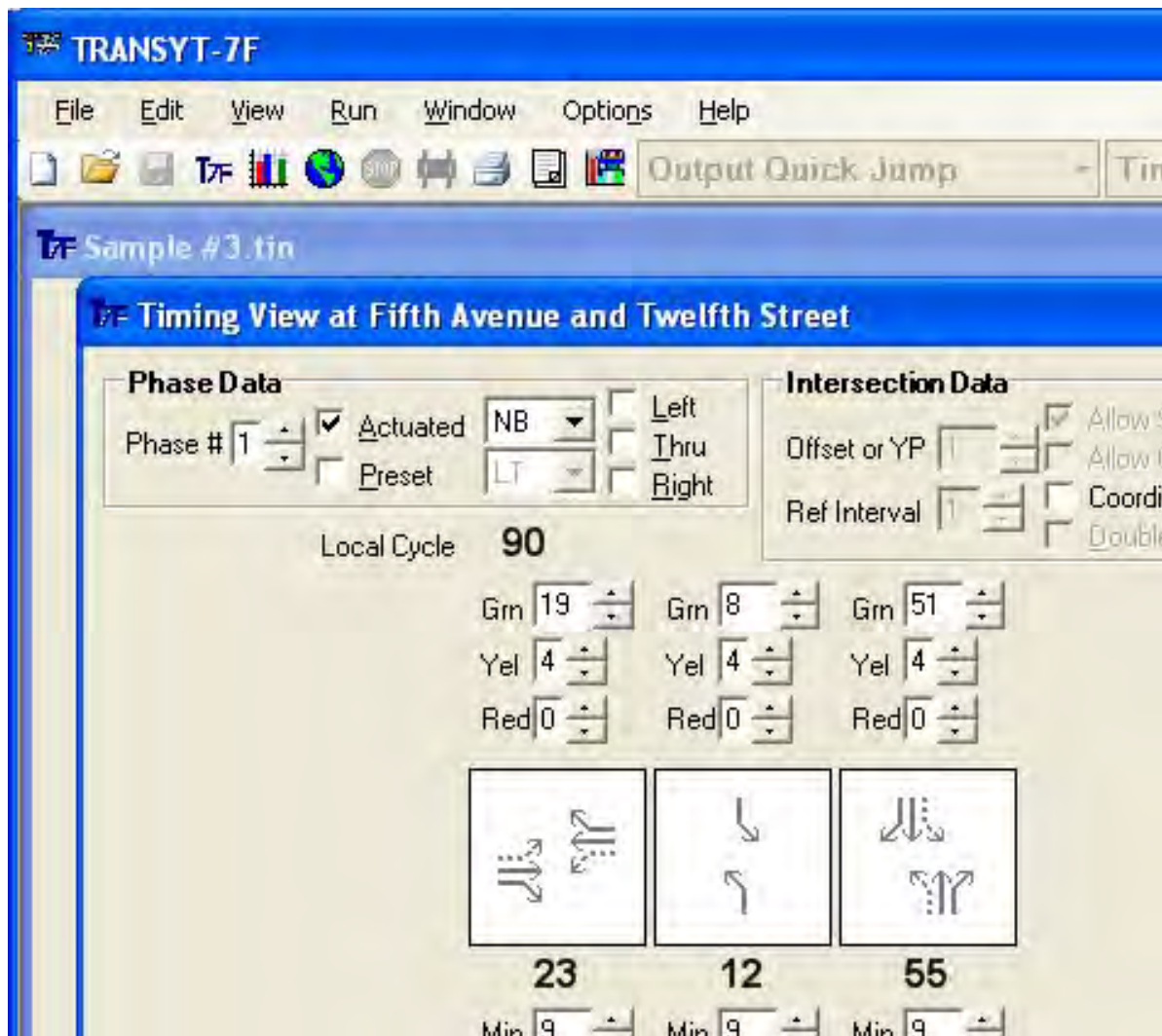
File Name	Sample #3.tin	Default Traffic Volume (vph)
Run Title	HCM Sample Calculation #3	Default Peak Hour Factor
Link Number Scheme	TRANSYT-7F Standard	Default Saturation Flow Rate (vph)
Number of Intersections	1	Default Link Length (ft or m)
Number of Routes	0	Default Free-Flow Speed (mph or km/h)
Cycle Length (sec)	90	Default Number of Phases
Duration of Analysis (min)	15	Default Yellow Time (sec)
Number of Time Periods	1	Default All-Red Time (sec)
<input checked="" type="radio"/> English Units	<input checked="" type="radio"/> Right-Hand Drive	Default Gap Setting (sec)
<input type="radio"/> Metric Units	<input type="radio"/> Left-Hand Drive	Default Detector Length (ft or m)

After clicking on "OK" to generate the new file, the user could optionally begin by coding the intersection name (Fifth Avenue and Twelfth Street) on the Edit > Optional > Intersection > Names screen. Then, the Lane Configuration screen (Edit > Lanes) is used to specify roadway geometry, as shown in the screen image below. Unadjusted hourly volumes, which would be counted in the field across one hour, are coded here. Later on, after coding the peak hour factors, peak adjusted volumes can be viewed or modified on the Edit > Traffic screen.



The Edit > Traffic screen could be viewed next to check the adjusted volumes and saturation flow rates, but for this example, no further changes are needed to the data on this screen. Link lengths have no effect on the overall delay and level of service results for an isolated intersection, although the fuel consumption and travel time results would be affected.

Next, the timing plan should be coded on the Timing View screen (Edit > Timing), as shown in the screen image below. For an isolated intersection, the offset and reference interval values are not applicable, and would have no effect on results.



The next step of the coding process is to specify the adjusted saturation flow rates. Selecting Edit > Satflow loads the Satflow View screen. In order to view the full contents of this screen, the screen resolution must be set to at least 1024x768 or higher. For this example, there are several mitigating factors that can adjust the saturation flow rates on each approach. Once finished, each value will be saved to the data file by clicking on the Apply button. Saturation flow rates for permitted-only links (106 and 108) and secondary shared links (109-112) do not need to be coded, because these flow rates are handled internally by the simulation model. After coding the input data as shown below, click the "Apply" button to automatically code adjusted saturation flow rates for each lane group. The six calculated saturation flow rates shown below should differ only slightly from the ones shown earlier in the HCS screen image.



### 7F Satflow View at Fifth Avenue and Twelfth Street

Eastbound			Westbound			Northbound			Southbound	
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru
106	105	111	108	107	112	102	101	109	104	103
Base Saturation Flow Rate (vplphg)										
<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>	<input type="text" value="1900"/>
Number of Lanes										
<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="2"/>
Lane Width (ft)										
<input type="text" value="10"/>	<input type="text" value="10"/>	<input type="text" value="10"/>	<input type="text" value="10"/>	<input type="text" value="10"/>	<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="text" value="12"/>	<input type="text" value="12"/>
Heavy Vehicles (%)										
<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="2"/>	<input type="text" value="2"/>
Grade (%)										
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Parking Prohibited										
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Parking Maneuvers Per Hour										
<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Number of Buses Stopping										
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Area Type										
<input type="text" value="CBD or similar"/>										
Lane Utilization										
<input type="text" value="Egl"/>	<input type="text" value="Slic"/>	<input type="text" value="Egl"/>	<input type="text" value="Egl"/>	<input type="text" value="Slic"/>	<input type="text" value="Egl"/>	<input type="text" value="Egl"/>	<input type="text" value="Slic"/>	<input type="text" value="Egl"/>	<input type="text" value="Egl"/>	<input type="text" value="Slic"/>
Type of Turning Movement										
<input type="text" value="LT"/>	<input type="text" value="TH"/>	<input type="text" value="RT"/>	<input type="text" value="LT"/>	<input type="text" value="TH"/>	<input type="text" value="RT"/>	<input type="text" value="LT"/>	<input type="text" value="TH"/>	<input type="text" value="RT"/>	<input type="text" value="LT"/>	<input type="text" value="TH"/>
Ped-Bike influence on Left-Turns										
<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>
Ped-Bike influence on Right-Turns										
<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>	<input type="text" value="Noi"/>

It should now be possible to run the TRANSYT model without errors. Although the HCS software reports an intersection-wide average delay of 44.2 seconds per vehicle, TRANSYT initially reports an intersection-wide average delay of 51.3 seconds per vehicle. However, if the user calibrates the model by coding 2.0 sneakers per cycle on the Edit > Optional > Permitted screen, TRANSYT reports an

intersection-wide average delay of 42.7 seconds per vehicle. Since TRANSYT and the HCM have different permitted movement models, their results will sometimes differ in the presence of permitted movements, unless some amount of calibration is performed.

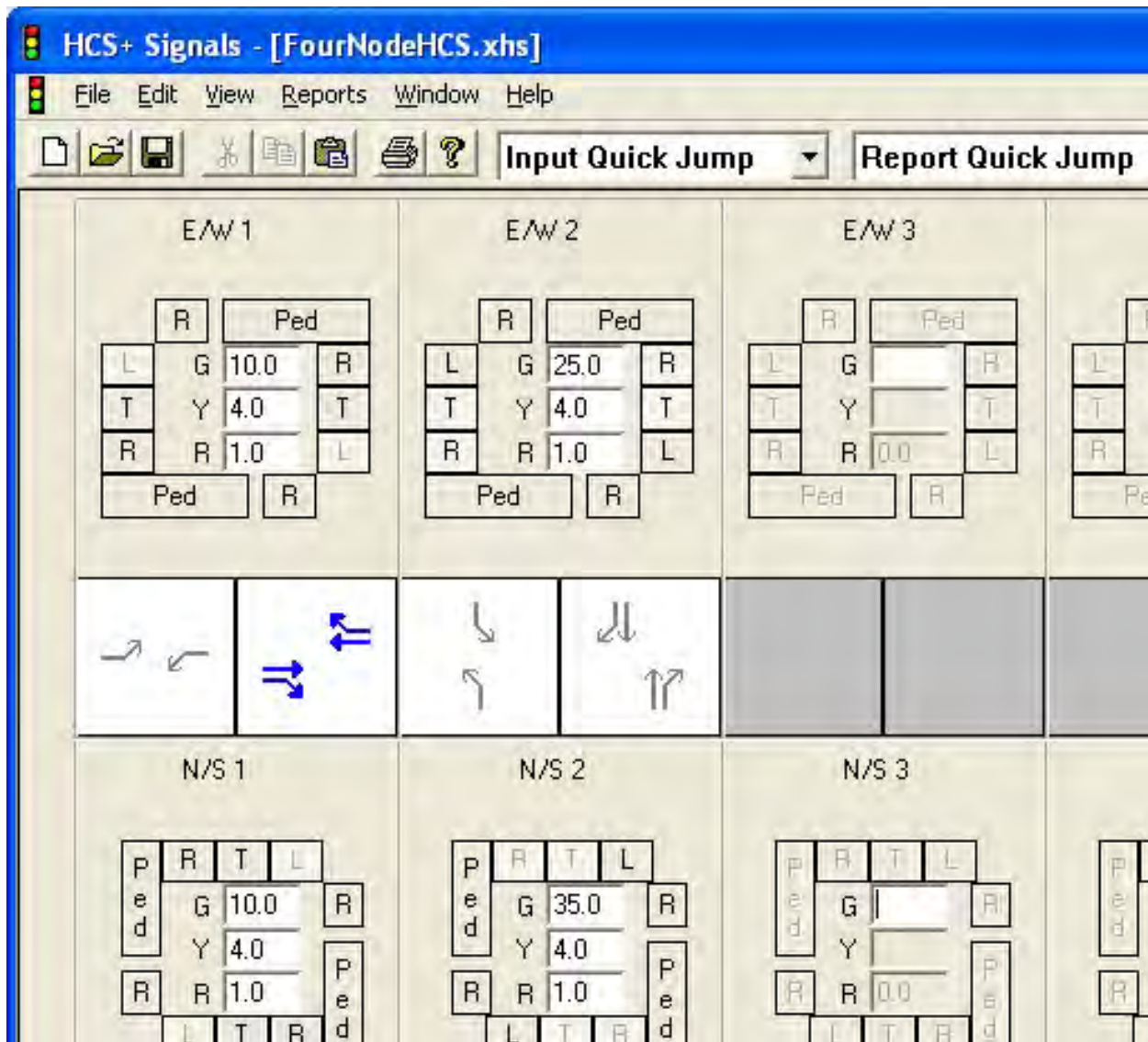
## Creating a New Input Data File by Importing HCS Files

The previous examples showed how to generate TRANSYT-7F input files from scratch, using the graphical user interface (GUI) input screens. It is also possible to generate the exact same network from scratch by importing individual HCS-Signals data files. From the GUI, proceed to File > Open, select "\*.XHS" as the file type, and browse to open the desired data file. This causes an isolated intersection TIN file to be generated automatically. Next, select File > HCS > Append an HCS-Signals intersection to this network. This will prompt the user to specify a node number for the new intersection to be imported.

Sample data files HCMSC1.xhs, HCMSC2.xhs, HCMSC3.xhs, and FourNodeHCS.xhs should be used for these exercises. For example, the isolated intersection (Sample #3.tin) from the previous exercise is easily reproduced simply by importing HCMSC3.xhs.

### FourNode Example

FourNodeHCS.xhs can be imported four times in a row to reproduce the four-node network (FourNode.tin) from the previous exercise (i.e., the FourNode Example from Creating a Brand-New Input Data File, at the beginning of Chapter 6). The screen image below illustrates the timing plan as seen from the HCS-Signals GUI.



After importing this HCS file once and then appending this HCS file three times, the following additional coding steps would be necessary to match the numerical results from the previous example:

- Switch from "Optimization" to "Simulation" (Analysis screen)
- Code all reference intervals as 5 (Timing screen)
- Code a default cruise speed of 35 mph (56 kmph) (Options > Cruise Speed)
- Drag-and-drop nodes to form a box with 1320-foot (402-meter) intersection spacing, as in the previous example (Map View)
- Code two-way link connections as in the previous example (Map View)
- Code an external link speed of 35 mph (56 kmph) (Global screen)
- Code a master controller node number of 1 (Global screen)

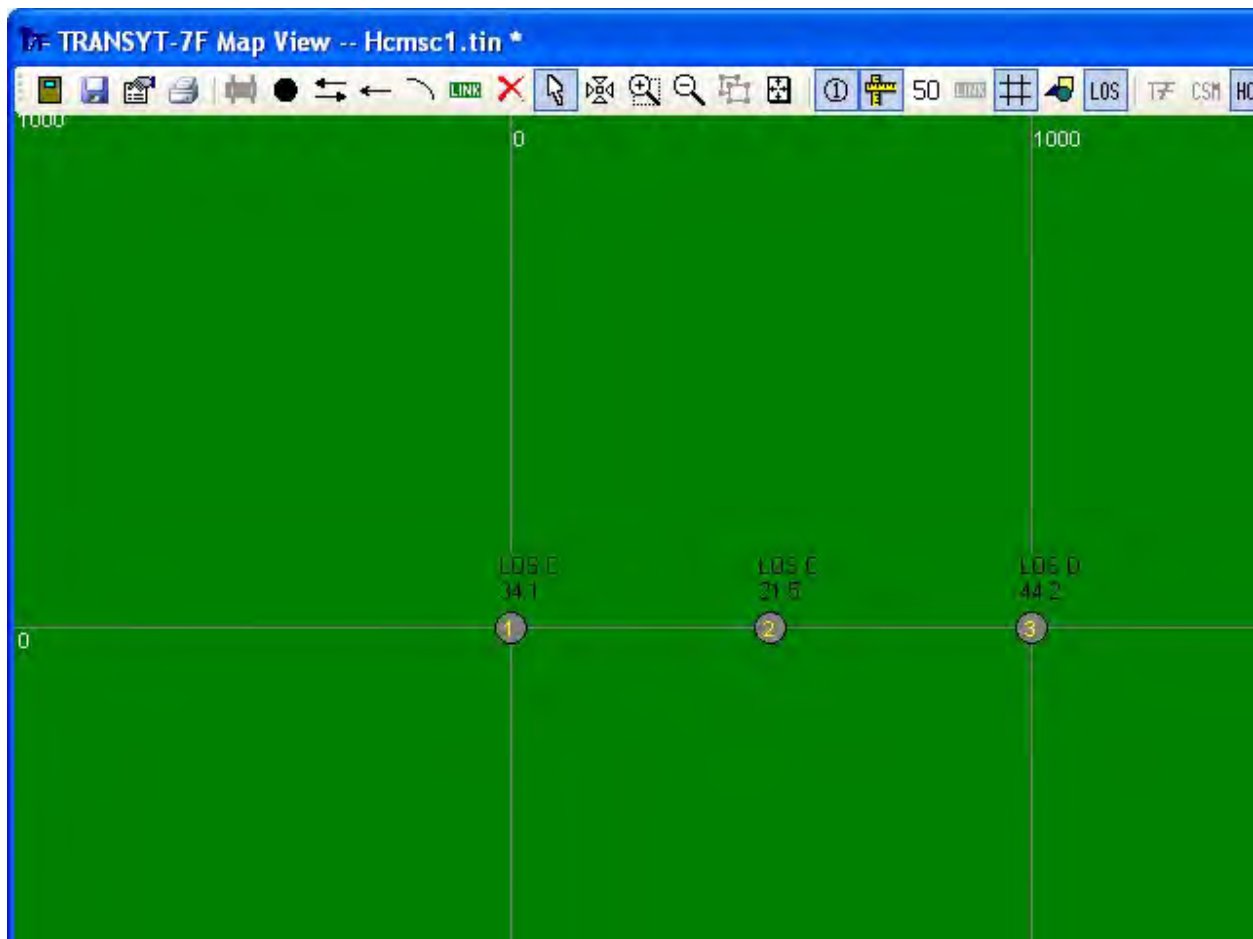
When this particular HCS file is imported, TRANSYT-7F link lengths and queuing capacities are estimated using the "Available Queue Storage" values from HCS-Signals.

#### Arterial Street Example

In this example, sample data files HCMSC1.xhs, HCMSC2.xhs, HCMSC3.xhs are all imported to create a three-intersection arterial street. Prior to importing the HCS files, specify a free-flow speed of 35 mph (56 kmph) on the Options > Cruise Speed screen. The first data file to be imported should be HCMSC1.xhs, which automatically creates a TRANSYT input file called HCMSC1.tin. Subsequently, click on File > HCS > Append an HCS-Signals intersection to this network, to import HCMSC2.xhs and HCMSC3.xhs.

At this stage it is possible to run TRANSYT optimization by clicking on the blue "T7F" icon. For this tutorial exercise, HCS model results will be compared with TRANSYT model results, but for existing conditions only. In order to model existing conditions, it is necessary to switch from optimization to simulation. Therefore, open the Edit > Analysis screen, and click on "Simulation". In addition, since these three intersections do not all have the same cycle length, turn on the "Uncoordinated Network" checkbox. After this, perform another TRANSYT run by clicking on the blue "T7F" icon.

In order to quickly compare results between HCS and TRANSYT, it is best to open the Map View. Try to place the nodes 500 feet apart from one another, to create an arterial street with 500-foot spacing. Then, click on the "LOS" icon, to superimpose intersection delays and LOS on the map. Then, by clicking between the "T7F" and "HCS" icons, a quick comparison is possible. The sample Map View is illustrated below.



Notice that the average intersection delay is significantly different between HCS and TRANSYT at all three nodes. This is due to an inappropriate application of single-cycle simulation in TRANSYT.

Because this arterial street currently has two intersections operating on a 70-second cycle, and one intersection operating on a 90-second cycle, it was necessary to simulate 90 seconds. However, this prevented TRANSYT from computing "average" results for the two intersections operating on a 70-second cycle. Proceed to the Edit > Analysis screen, and select "Multi-Cycle" simulation.

After re-running the TRANSYT model, the Map View shows that results are now consistent between HCS and TRANSYT at nodes 2 and 3, but there is still a significant difference at node #1. The results at node #1 can be made somewhat more consistent by going to the Edit > Optional > Permitted screen, coding opposing right-turn links (112, 111, 110, and 109) for the left-turn links (106, 108, 102, and 104), and eliminating the 2.0 left-turn sneakers (click on "Next Data") per cycle for each left-turn link.

To create an arterial street with platoon progression, draw some two-way link connections between nodes 1-2 and nodes 2-3 on the Map View. Then, define a progression route by selecting all three nodes, and clicking on the Progression Route icon. Go ahead and accept all default values on the Progression Route dialog screen, including an eastbound Forward Direction, and no Directional Weighting. Then, proceed to the Edit > Analysis screen, and turn off the "Uncoordinated Network" checkbox. Since the intersections have different cycle lengths, it is also necessary to turn on "Initial Timing Flag 1". Next, turn on the "Coordinated Signal" checkbox for every intersection on the Edit > Timing screen. Finally, at node #3, turn off the Actuated checkbox for phase #1, and code interval #2 as the Reference Interval. At this stage, it would be possible to optimize progression, and view time-space diagrams. These exercises will be demonstrated within the Progression Analysis Examples tutorial section.

## Editing Input Data (Sensitivity Analysis)

This section of the documentation demonstrates how the TRANSYT model reacts to basic changes in a variety of input data. The "base case" or "base scenario" network is the one generated from scratch in the first exercise of the tutorial chapter, i.e., the four-node "box" shaped network. Alternatively, this file ("FourNode.tin" or "FourNode Lefty.tin") can also be obtained from the TRANSYT-7F Example folder.

The exercise begins by examining the TRANSYT link performance summary tables at intersection #1. Although the average delay for the entire intersection is 54.3 seconds per vehicle, average delay for the eastbound through movement is reported as 70.6 seconds per vehicle. Further, degree of saturation for the eastbound through movement is reported as 96%. This output listing is illustrated in the screen image below.

(For left-hand drive, the exercise begins by examining the TRANSYT link performance summary tables at intersection #2. Although the average delay for the entire intersection is 54.2 seconds per vehicle, average delay for the westbound through movement is reported as 70.6 seconds per vehicle. Further, degree of saturation for the westbound through movement is reported as 96%.)



**TRANSYT-7F**

File Edit View Run Window Options Help

Output Quick Jump Time Period

C:\Documents and Settings\All Users\Documents\Sample.tof

<PERFORMANCE WITH INITIAL SETTINGS>

Movement/ Node No.	Flow vph	Sat Flow vphg	Degree of Sat %	Total Travel veh-mi	Travel Time Total veh-h	Time Avg sec/v	----- Uniform	Random	Delay - Tot
NB THRU	450	1900P	69	112.41	7.09	56.7	3.48	0.38	3.8
LEFT	50	1805	28	12.49	0.95	68.4	0.58	0.01	0.5
SB THRU	450	1900P	69	112.41	8.16	65.2	4.66	0.27	4.9
LEFT	50	1805	28	12.49	0.99	71.5	0.63	0.01	0.6
EB THRU	450	1900P	96*	112.41	12.06	96.5	4.63	4.20	8.8
LEFT	50	1805	28	12.49	0.95	68.4	0.58	0.01	0.5
WB THRU	449	1900P	96*	112.41	13.33	106.8	7.65	2.44	10.0
LEFT	50	1805	28	12.49	1.01	72.4	0.65	0.00	0.6
1 :	1999	MAX =	96*	499.58	44.53		22.85	7.32	30.1

Movement/ Node No.	----- Uniform	Max Back of Queue	----- Random	----- Total	Queuing Capacity	Time Full Link	Critical
-----------------------	------------------	-------------------	-----------------	----------------	---------------------	-------------------	----------

### Sensitivity Analysis of Traffic Data

After running the model to observe these initial results, it is best to close the legacy output file report by selecting File > Close. Then, proceed to the Edit > Traffic screen, and reduce the eastbound through traffic volume at node #1 from 400 vph to 350 vph. (For left-hand drive, reduce the westbound through traffic volume at node #2 from 400 vph to 350 vph.) After changing the traffic volume, note that the file name at the top of GUI now has an asterisk next to it, to the right. This is called the "dirty file flag", and it indicates that the input data has been modified since the file was last saved. When data are saved within the individual input screens, this saves the "temporary file", but does not save the "permanent file" that usually resides on the user's local hard drive. In order to save the permanent file, it is necessary to manually select File > Save, press Ctrl-S, or click on the floppy disk icon at the top of the GUI.

Now run the TRANSYT model one more time (blue T7F icon, Run > TRANSYT-7F, or Ctrl-R) with the new traffic volume of 350 vph. This time, average delay for the corresponding through movement is reported as 54.1 seconds per vehicle, and the degree of saturation is reported as 86%. This reduction in delay, from 70.6 to 54.1 seconds per vehicle, is fairly significant, and reflects the high degree of saturation for this movement. At lower degrees of saturation, a similar volume reduction would be unlikely to reduce delay as much. Note that, for internal links having a nearby upstream signal, any change to the traffic volume should usually be followed by a modification of the feeding link flows. Feeding link flows can be revised on the Edit > Feeders screen, or by deleting and re-drawing links on the Map View.

For the next experiment, reset the through movement traffic volume to the original value of 400 vph, and then increase its saturation flow rate from 1900 vphg to 2000 vphg. After running the model, average delay for the through movement is now 60.8 seconds per vehicle. Delay is lower than it was in the base scenario, because higher saturation flow rates cause the queue to be served more quickly.

Finally, reset the through movement saturation flow rate to the original value of 1900 vphg, and then decrease its start-up lost time from 2.0 seconds to 1.0 seconds. After running the model, average delay for the through movement is now 62.2 seconds per vehicle. Again, delay is lower than it was in the base scenario, because lower start-up lost times allow the queue to be served more quickly.

### Sensitivity Analysis of Timing Data

The following tutorial exercises will demonstrate the effect of changing signal timing-related data. To begin, it is necessary to restore the base scenario from the previous example. This time, instead of examining individual link performance, the network-wide summary results are examined instead. In the base scenario, network-wide average delay is reported as 54.3 seconds per vehicle for right-hand drive, and 54.4 seconds per vehicle for left-hand drive.

Next, proceed to the Edit > Analysis screen. Note that the original analysis period duration was coded as 15 minutes, which is the recommended duration for most analyses. This time, try coding an analysis period duration of 60 minutes, save the data, and run the model. Note that the network-wide average delay is now reported as 61.5 seconds per vehicle for right-hand drive, and 61.6 seconds per vehicle for left-hand drive. The increase in delay is caused by the random (or "incremental") delay model, which stipulates that sustained periods of near-saturation or oversaturation will result in increased opportunities for phase failures. If the majority of the network had been undersaturated, changing the analysis period duration would not have affected the results very much.

Return to the Edit > Analysis screen, and restore the original analysis period duration of 15 minutes, in order to restore the base scenario conditions. Next, turn on the checkbox that says "Ignore Actuated Flags". This checkbox allows the network to be quickly evaluated as "pre-timed", without having to recode the intersection-specific timing data. After saving the data and running the model, the network-wide average delay is now reported as 55.6 seconds per vehicle, which is slightly higher than the base scenario delay of 54.3 seconds per vehicle. Average delay is always expected to be lower under actuated control, although the degree of delay reduction is a function of network-wide degrees of saturation, and certain actuated controller settings. Average phase times are also usually lower for actuated control, relative to pre-timed control. Under the right conditions, actuated control can offer dramatic reductions in delay.

Return to the Edit > Analysis screen, and turn off the checkbox that says "Ignore Actuated Flags", in order to restore the base scenario conditions. Next, turn on the checkbox that says "Initial Timing Flag 1". This checkbox causes TRANSYT to generate initial green times to be simulated at each intersection, such that any previously user-coded green times will be ignored. Note that "Initial Timing Flag 2" behaves the same way as Initial Timing Flag 1 with regard to the green times, but the offsets are handled differently. Refer to the Initial Timing Model section of the documentation for more details. After saving the data and running the model, the network-wide average delay is now reported as 41.3 seconds per vehicle, which is a significant improvement over the base case. This indicates that the original user-coded green times were not very effective.

Return to the Edit > Analysis screen, and turn on the radio button selection that says "Optimization". Next, turn on the checkboxes "Splits" and "Offsets", and select the radio button for "Hill-Climb". These selections request hill-climb optimization of splits and offsets. After saving the data and running optimization, network-wide average delay is now reported as 32.4 seconds per vehicle, which is a significant improvement over the previous runs. Note that "Disutility Index (DI)" was the optimization objective function for this run, and average delay is a significant component of the DI. A network-wide delay of 40.0 would be obtained by deactivating the initial timing flag prior to optimization, which demonstrates that the process is sometimes sensitive to its starting point.

Return to the Edit > Analysis screen, and select the radio button for "Genetic Algorithm". Next, click on the button that says "Genetic Algorithm Settings". Code a "Mutation Probability" of 4.0%, and a "Maximum Number of Generations" of 350. For this simple example network, the genetic optimization result is only slightly superior to the hill-climb optimization result. More detailed networks tend to receive a significantly better design from the genetic algorithm, when applied properly. By doing a sample genetic algorithm optimization run, it is also possible to observe that the optimization process takes longer (on the computer) than hill-climb, and that it is possible to abort the optimization run by clicking on the Stop sign icon at the top of the GUI. Refer to the Recommended G.A. Parameter Values section of the documentation for more information on the appropriate genetic algorithm parameter values for certain networks.

### Sensitivity Analysis of Turning Movement Protection

The following tutorial exercises will demonstrate the effect of changing major street left-turn protection. For driving on the left-hand side of the road, this exercise (using "FourNode Lefty.tin") would involve changing the major street right-turn protection. To begin, it is necessary to restore the base scenario from the previous example. This time, instead of examining network-wide summary results, the northbound left-turn degree of saturation at node #1 is examined instead. (For left-hand drive, the northbound right-turn degree of saturation at node #2 is examined.) In the base scenario, the northbound turning movement operates in "protected-only" mode, and has a degree of saturation of 28%.

Next, proceed to the Edit > Timing screen, and switch to the appropriate node number mentioned above. Remove the northbound protected turn from phase #3 and add a northbound permitted turn to phase #4. After saving the new "permitted-only" turn protection and running the model, the northbound turning movement degree of saturation remains unchanged. To introduce a difference into the permitted-only results, code the opposing shared turning movement as "Opposing Link #2" on the Edit > Optional > Permitted screen. For right-hand drive, the northbound permitted left-turn should yield to 100% of the flow of southbound right-turn link 110. For left-hand drive, the northbound permitted right-turn should yield to 100% of the flow of southbound left-turn link 204. After re-running the model, degree of saturation for the northbound permitted movement is now reported as 43%, because it must now yield to additional opposing flow. To restore the base scenario conditions, Opposing Link #2 should now be deleted within the Edit > Optional > Permitted screen.

Next, return to the Edit > Timing screen. Restore the northbound turn protection to phase #3, but do not remove the northbound permitted turn in phase #4. After saving the new "protected-permitted" or "compound" turn protection and running the model, the northbound turning movement degree of saturation is reported as 11%. Congestion has decreased for the turning movement, because it can now move in both the protected and permitted phases. Although this type of protection always minimizes delay for the turning movement, the protected portion of the phase takes green time away from the through movements, and the permitted portion of the phase is occasionally a safety hazard for the turning drivers.

Finally, return to the Edit > Timing screen for node #1. Remove the northbound left-turn protection in phase #3. Then, proceed to the Edit > Lanes screen. Make the permitted left-turn a shared left-turn by coding a lane use of "LTR" for lane #2. Then, delete lane #1 by either right-clicking on lane #1 and selecting "Delete", or by using the "Add/Delete" pushbutton. After saving the new "permitted-shared" turn protection and running the model, the northbound thru degree of saturation is reported as 85%, which exceeds the original value of 69%. Congestion has significantly increased for the northbound approach, because the shared lane now experiences additional congestion and queuing. Intersection average delay has also increased from 54.3 seconds per vehicle to 55.6 seconds per vehicle.

(For left-hand drive, return to the Edit > Timing screen for node #2. Remove the northbound right-turn protection in phase #3. Then, proceed to the Edit > Lanes screen. Make the permitted right-turn a shared right-turn by coding a lane use of "LTR" for lane #1. Then, delete lane #2 by either right-clicking on lane #2 and selecting "Delete", or by using the "Add/Delete" pushbutton. After saving the new

"permitted-shared" turn protection and running the model, the northbound thru degree of saturation is reported as 85%, which exceeds the original value of 69%. Congestion has significantly increased for the northbound approach, because the shared lane now experiences additional congestion and queuing. Intersection average delay has also increased from 54.2 seconds per vehicle to 55.5 seconds per vehicle.)

## Progression Analysis Examples

This section of the documentation demonstrates the coding and analysis of progression-related input data within TRANSYT. The "base case" or "base scenario" network is the one generated from scratch in the first exercise of the tutorial chapter, i.e., the four-node "box" shaped network. Alternatively, this file ("FourNode.tin" or "FourNode Lefty.tin") can also be obtained from the TRANSYT-7F Example folder. Note that the results from FourNode.tin throughout this exercise are sometimes slightly different than the ones from FourNode Lefty.tin, due to round-off errors caused by metric unit conversion.

The second sample network is the three-intersection arterial street, created earlier in the HCS import tutorial section. In this example, HCMSC1.xhs, HCMSC2.xhs, and HCMSC3.xhs were all imported to create an arterial street within HCMSC1.tin. Progression analysis of this arterial street will take place after the initial set of FourNode exercises.

The initial set of program runs will be used to demonstrate model sensitivity to progression-related input data. Next, the coding of progression routes, and interpretation of route-specific outputs, will be demonstrated. Finally, the Time-Space Diagram and Profile View screens will be used to illustrate progression results graphically, instead of numerically.

### Progression Sensitivity Analysis

The exercise begins by examining the TRANSYT link performance summary tables (specifically the westbound through movement) at intersection #1. (For left-hand drive, the eastbound through movement at intersection #2 is examined.) Although the average delay for the entire intersection is 54.3 seconds per vehicle, average delay for the through movement is reported as 80.9 seconds per vehicle. Further, degree of saturation for the through movement is reported as 96%. This base scenario output listing is illustrated in the screen image from the prior "Editing Input Data (Sensitivity Analysis)" tutorial exercise. On the second table of link-specific results, arrival type is reported as 1 for the through movement, which denotes poor progression.

After running the model to observe these initial results, close the report by selecting File > Close. Then, proceed to the Edit > Traffic screen for the appropriate node number mentioned above, and reduce the through movement link length from 1320 feet (402 m) to 1000 feet (305 m). Depending on the graphical user interface (GUI) settings (refer to the documentation for Preferences), coding one of the link lengths on an approach will sometimes automatically update every link length on the approach. It is usually preferable for each link length on an approach to be equal, because this allows for proper simulation of platoon dispersion. Shorter turn pocket lengths can be optionally modeled by explicitly coding the appropriate queuing capacities, but when conditions are undersaturated (no queue spillover), this may have little impact on the results.

After running the model, average delay for the through movement is now 75.7 seconds per vehicle. Delay is lower than it was in the base scenario, because the lower link length allows for better progression (i.e., dense platoons arriving on green). Although it is often not possible to change link lengths in the field, this exercise demonstrates the impact of progression on the delay model. Note that lower link lengths will not always result in lower delays and better progression, because progression is also affected by the offsets and phasing sequence. Generally, lower link lengths facilitate the best progression, provided that the timing plan is optimized.

For the next experiment, reset the through movement link length to the original value of 1320 feet (402 m). Then, proceed to the Edit > Optional > Global > Network Parameters screen, and code a "Network-Wide Platoon Dispersion Factor" of 50, which represents a PDF value of 0.50. After running the model, average delay for the through movement is now 78.8 seconds per vehicle, which is also lower than the base case delay of 80.9 seconds per vehicle. This result is slightly unusual, because increasing the PDF tends to decrease platoon intensities, which tends to increase delays. This implies that with a link length of 1320 feet (402 m), and the prevailing timing plan, the platoon centroid is arriving on red.

Next, return to the Edit > Optional > Global > Network Parameters screen, and code a Network-Wide PDF of 25, which represents a PDF value of 0.25. Generally, lower PDF values are used to simulate more intense platooning. After running the model, average delay for the through movement is now 81.7 seconds per vehicle, which is slightly higher than the base case delay of 80.9 seconds per vehicle. This further implies that the platoon centroid is arriving on red. After this, the Network-Wide PDF value should be restored to its original value of 0.35, to facilitate the upcoming exercises.

### Coding of Progression Routes and Interpretation of Route-Specific Outputs

Progression route coding is necessary to access some of the corridor analysis features within TRANSYT-7F. These features include progression-based optimization, route summary statistics, and time-space diagramming. In a grid network system, the optimization process needs to know which major arteries are preferred for progression. Progression would be optimized to some degree without defining arterial routes, because the disutility index (DI) accounts for stops, and because good progression tends to reduce delay. However, DI optimization is unlikely to optimize progression as effectively or efficiently as PROS (or PROS/DI) optimization, which explicitly considers the arterial routes. Moreover, progression optimization may even lead to a lower disutility index than DI-only based optimization, because the optimization process is able to focus on a certain subset of logical timing plans.

There are multiple techniques for coding progression routes. Numerical link pairs along the progression route may be coded directly on the Edit > Optional > Route screen (or record type 42). In this example, progression routes will be coded graphically on the Map View. Following the graphical coding, the numerical link pairs will be automatically written to the data file, and could be viewed on the Edit > Optional > Route screen at any time.

The first step toward graphical coding of progression routes is to launch the Map View by clicking on the "earth" icon, or pressing F3. Next, drag a mouse "rubber band" around node numbers 1 and 2, to select these nodes for the progression route. The nodes become highlighted after being selected. It is also possible to use Ctrl-<Right Click> to select multiple nodes, if it is not possible to drag a rubber band around the desired group of nodes.

Next, click on the Progression Route icon, or press the X key. On the Progression Route dialog screen, select an eastbound Forward Direction, and no Directional Weighting. (For left-hand drive, select westbound as the forward direction.) After clicking the "OK" button, the Map View should now illustrate the progression route using bold lines for emphasis. To complete the progression route coding, define a second route to include nodes 3 and 4.

Now that the progression routes have been defined, it is possible to obtain route-oriented outputs and timing plans from the model. First, a "simulation-only" run is performed, and the results are displayed in the screen image below. It is possible to use the Output Quick Jump to go directly to the Route Performance summary tables. Route performance results are provided for both the forward and reverse arterial directions. For example, in the eastbound direction, results are provided for links "105" and "205", in accordance with the link numbering scheme.

<ROUTE SUMMARY REPORT>

Route #1

Node No.	Link No.	Flow vph	Sat Flow vphg	Degree of Sat %	Total Travel veh-mi	Travel Time Total veh-h	Travel Time Avg sec/v	Delay Uniform veh-h	Delay Random veh-h
1	105	450	1900P	96*	112.41	12.06	96.5	4.63	4.20
2	205	449	1900P	96*	112.41	13.35	107.0	7.69	2.43
Forward :		899	MAX =	96	224.81	25.41		12.32	6.62
1	107	449	1900P	96*	112.41	13.33	106.8	7.65	2.44
2	207	450	1900P	96*	112.41	12.06	96.5	4.63	4.20
Reverse :		899	MAX =	96	224.81	25.38		12.28	6.64

Now, select "Progression" from the Output Quick Jump list. This output table provides additional artery-wide measures of effectiveness for each route, including "efficiency", "attainability", "interference", and "PROS", which are defined in Chapter 4. Note that fairly long routes, which include numerous signalized intersections, can sometimes have efficiencies and attainabilities of zero. This is because the longer a route gets, the more physically difficult it becomes to achieve a non-zero progression band throughout the artery.

Other progression-related output sections available on the Output Quick Jump list include the time space diagrams, and the flow profile diagrams. The "percentage of vehicles arriving on green (PVG)", "platooning index", and "platooning moment" output parameters are reported among the flow profile diagrams. For example, by using the Output Quick Jump to go to the flow profile diagrams, the percentage of vehicles arriving on green (PVG) for the original base scenario is observed as only 5% for the westbound through movement (link #107), which indicates very poor progression. (For left-hand drive, the percentage of vehicles arriving on green for the original base scenario is observed as only 6% for the eastbound through movement, link #205.) Other progression-related outputs that do not command their own report sections include "arrival type" and "stops".

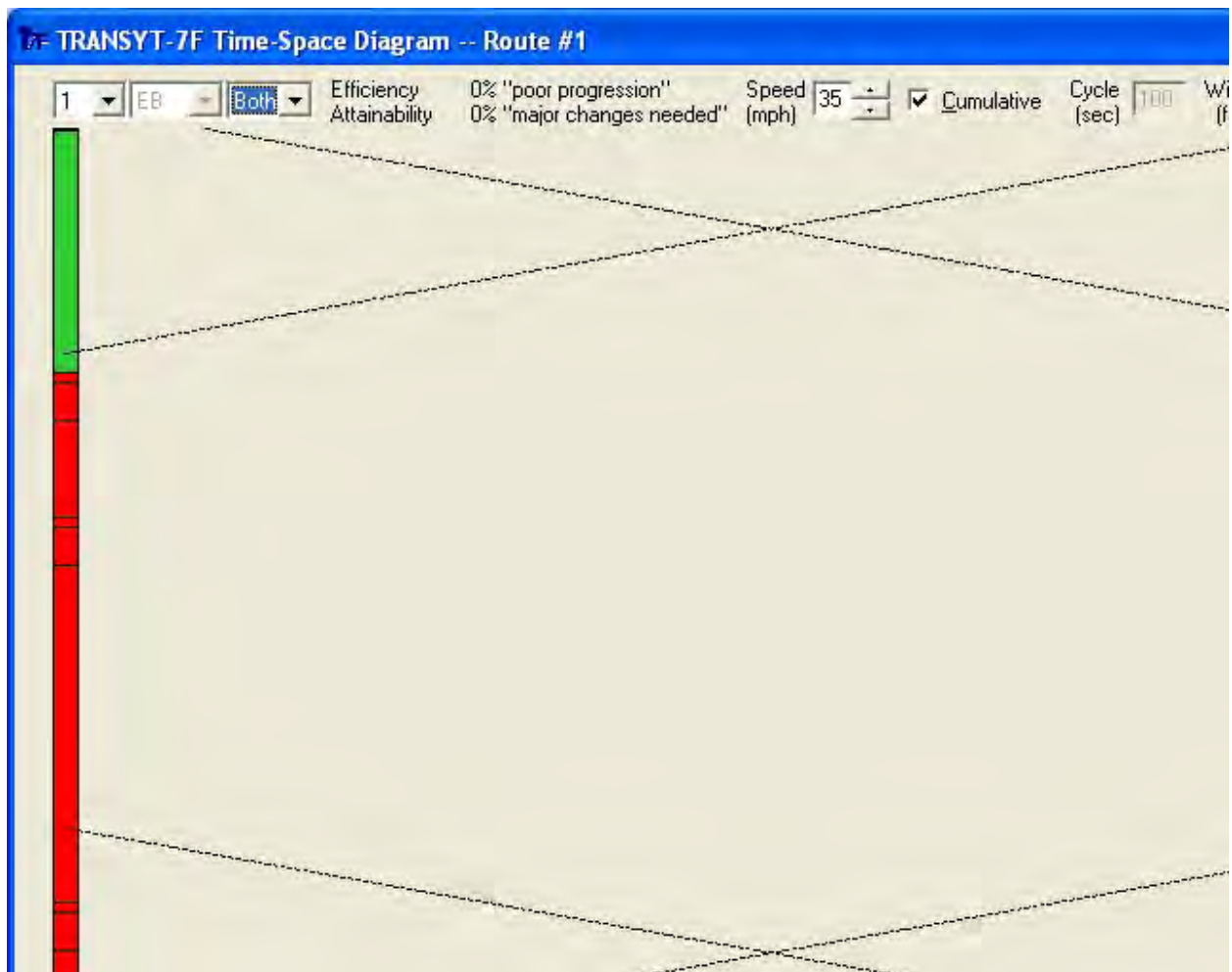
Note that the coding of progression routes has no effect on the optimization process, unless one of the progression-related objective functions (e.g., PROS-only, PROS/DI) has been selected. If the user switches to one of the progression-related objective functions, this may result in a different timing plan being recommended as optimal. For example, higher cycle lengths tend to maximize progression, whereas lower cycle lengths tend to minimize the disutility index (DI).

### Graphical Progression Analysis



Analyzing progression results graphically can provide additional information beyond the standard numerical outputs. For example, suppose that the analyst is uncertain about whether the optimization process has successfully maximized progression. In this case, the Time-Space Diagram screen could be used to visually determine if there are sufficient progression opportunities. For another example, suppose that the analyst wishes to minimize red light running at an intersection known for its high accident rates. In this case, the Profile View screen can be used to determine if there are dense platoons arriving towards the end of the green window. The Profile View screen can also be used to determine whether a platoon is arriving on green, by comparing the "input flow rate" profile with the "saturation flow rate" profile, because the saturation flow profile indicates the green portion of the cycle for any particular movement.

Route #1 is illustrated in the screen image below. To paint progression bands in both directions simultaneously, it is also necessary to select "Both" in the directional combo box, as shown at the top of the screen image below. Next, confirm the arterial cruise speed by going to Options > Cruise Speed. The default cruise speed to generate the diagram below should be 35 mph (56 kmph), to be consistent with data from the Edit > Feeders screen. This particular progression diagram illustrates the "simulation-only" results (no optimization). The Time-Space Diagram screen can be loaded by clicking on the blue "arterials" icon (originally from HCS-Arterials) at the top of the TRANSYT-7F GUI, or by selecting View > Progression Diagram. Note that when the input file does not contain any progression routes, the Progression Diagram icon is automatically disabled.



For both routes, in both the forward and reverse directions, the screen shows that the base scenario has probably the worst possible progression. Not only does the progression band arrive within the red window downstream, but in each case it arrives toward the beginning of the red window, which implies the longest possible uniform delay time until the upcoming beginning of green. Better progression would be expected from an optimized timing plan.

Notice that Efficiency and Attainability are both reported as 0% on the Progression Diagram screen. However, in the legacy report, Efficiency was reported as 4%, whereas Attainability was reported as 13%. This difference occurs because the Progression Diagram screen uses a more modern definition of effective green time. Refer to the Efficiency and Attainability sections of the documentation for more details.

The second arterial route within the data file, containing nodes #3 and #4, can also be examined by switching the route number combo box from "1" to "2" in the upper left-hand corner of the Progression Diagram screen. For the base scenario, the progression diagram will be the same for both routes, because the timing plans (including the offsets) are identical at all four intersections.

Next, change the arterial cruise speed at the top of the Time-Space Diagram screen. The default cruise speed that generated the diagram above should have been 35 mph (56 kmph), to be consistent with data from the Edit > Feeders screen. Change the speed to 45 mph (72 kmph). The progression bands should change slope in response to the new data. Notice now that the higher travel speed appears to provide a minimal bandwidth for the initial vehicles departing node #3 to arrive at the end of green downstream at node #4, and vice versa from node #4 to node #3. Restore the original cruise speed of 35 mph (56 kmph), to facilitate the remainder of the exercise.

At this time, the user could also experiment with changes to the offsets and reference intervals on the Edit > Timing screen. For example, since it takes 25.7 seconds to travel 1320 feet (402 m) at a speed of 35 mph (56 kmph), coding an offset of 26 seconds at the downstream node should produce an ideal bandwidth in the forward direction. The Time-Space Diagram screen is designed to update itself whenever the screen is resized. Restore the base scenario by coding an offset of zero seconds. Notice that the offsets and progression bands can also now be adjusted graphically, using the mouse, within the interactive Time-Space Diagram screen.

For the next experiment, go to the Edit > Analysis screen, and select hill-climb optimization of splits and offsets, along with initial timing flag 1. Notice that the default performance index (PI) is coded as the disutility index (DI). Now select "Run TRANSYT-7F" to initiate the optimization process. This hill-climb optimization run was essentially identical to the one from the "Editing Input Data (Sensitivity Analysis)" exercise. The System Summary output table again shows an average delay of 32.4 seconds per vehicle for the network. If the Output Quick Jump is used to jump to the "Progression" section of the legacy report, it can be seen that there are an average of 19 progression opportunities (PROS) for both directions of both routes.

Go to the Edit > Analysis screen, and select "PROS/DI" as the new performance index for optimization. Then re-run the optimization process by selecting "Run TRANSYT-7F". As expected, the System Summary output table indicates that this optimized timing plan has a higher disutility index (DI) than the previous run involving DI-only optimization (i.e., 105.7 for this run vs. 104.6 previously). However, the network-wide average delay per vehicle is lower (i.e., 31.2 seconds per vehicle for this run vs. 32.4 previously), which does not always happen with progression-based optimization. If the Output Quick Jump is used to jump to the "Progression" section of the legacy report, it can be seen that there are an average of 23 PROS for both directions of both routes.

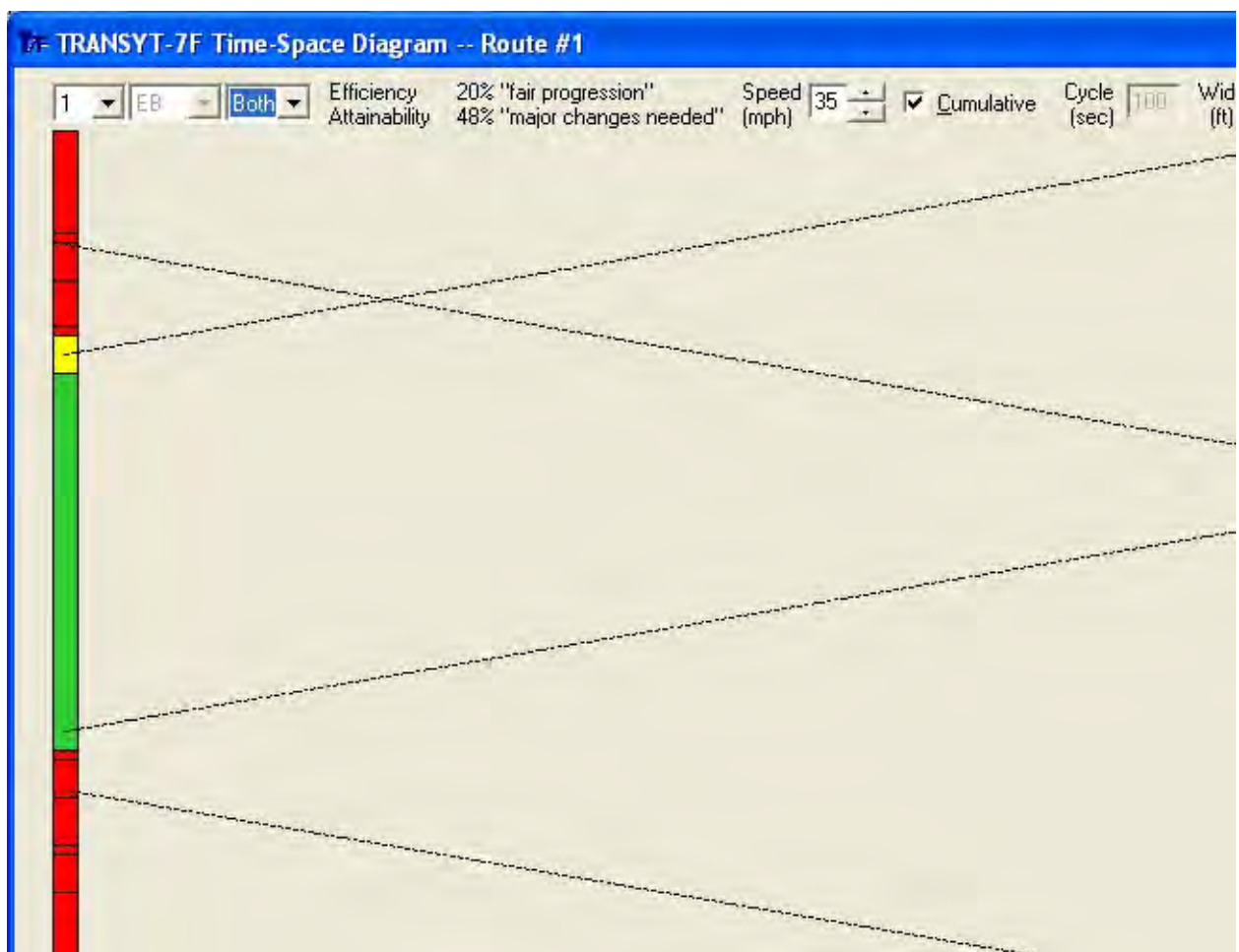
Next, load the Map View, select route #1 (nodes 1 and 2), and click the Progression Route icon. Turn on the "Use Directional Weighting" checkbox to code Forward and Reverse Weightings of '99' and '1'. After re-running the optimization process, if the Output Quick Jump is used to jump to the "Progression" section of the legacy report, it can be seen that there are 45 PROS for the forward direction



of route number 1.

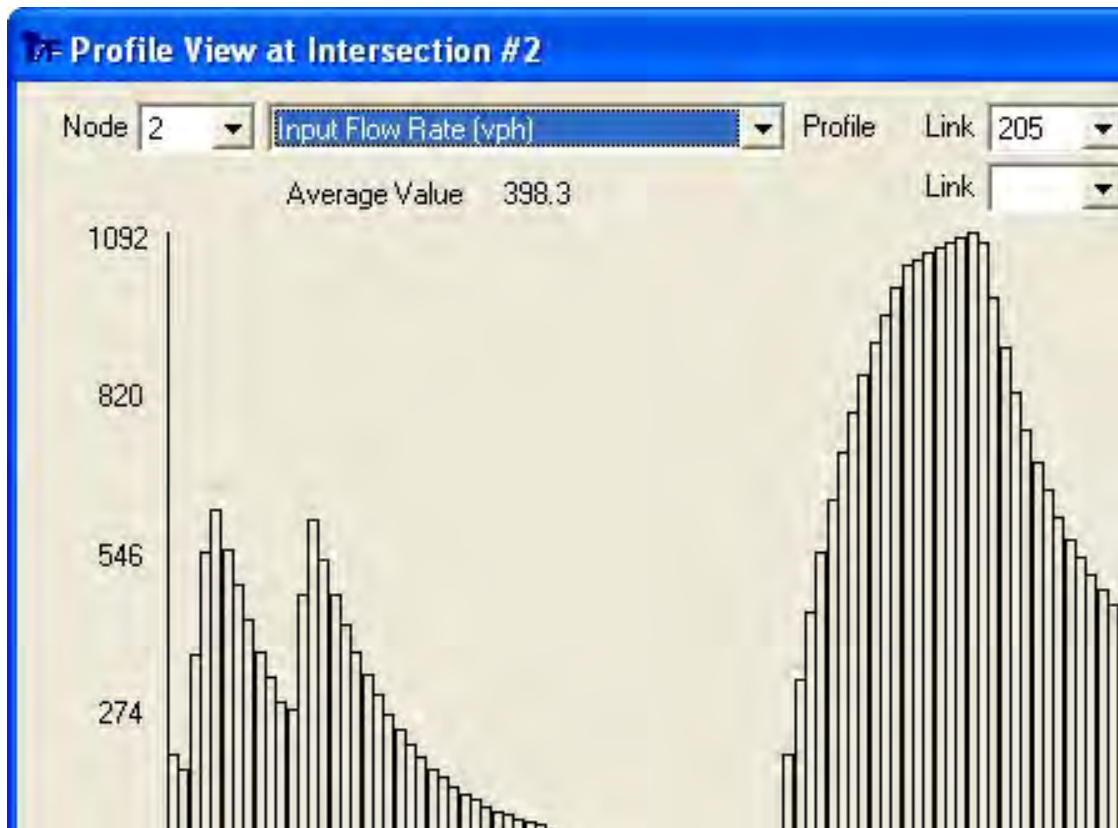
The punch file (a duplicate copy of the input file containing the optimized timing plan) should now be loaded, in order to view the progression diagram. Select File > Open, and move the file type selection from "\*.TIN" to "\*.PUN". It should now be possible to locate and open the punch file "Sample.pun". Note that if any further optimization runs were desired, it would be helpful to use File > Save As in order to restore the \*.TIN extension, such that there would be no danger of the punch file overwriting the input file (or vice versa).

Next, load the progression diagram (shown below) by clicking on the blue arterials icon, or by selecting View > Progression Diagram. Progression is almost perfect in the forward direction, due to the selection of PROS/DI as the optimization objective function, in conjunction with 99% arterial weighting in the forward direction. Progression is terrible in the reverse direction, although this might be acceptable for commuting traffic in some situations. The user could optionally apply more weighting to the reverse direction.

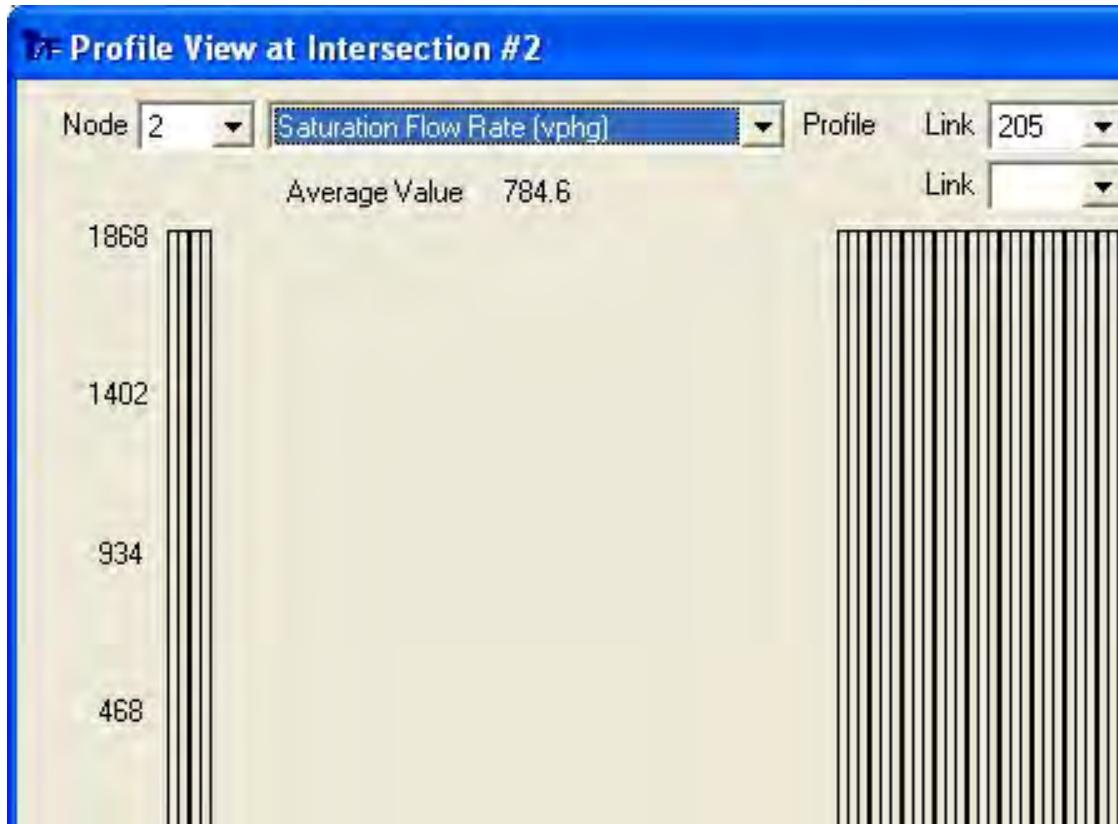


As stated earlier the Profile View screen can also be used to determine whether a platoon is arriving on green, by comparing the "input flow rate" profile with the "saturation flow rate" profile, because the saturation flow profile indicates the green portion of the cycle for any particular movement. To observe this concept, the user should first run Spyglass by clicking on the "bar graph" icon, or by selecting Run > Spyglass. This generates the necessary flow profile data on the hard drive. After Spyglass finishes running, the user is prompted to load the Profile View screen. After loading the Profile

View screen, switch to node #2 and link #205, which represents the eastbound major street through movement. (For left-hand drive, switch to node #1 and link #107, which represents the westbound major street through movement.) The screen image below illustrates that the most intense part of the platoon arrives at "time 75".



Next, switch to the "saturation flow rate" profile. The screen image below illustrates that the effective green time occurs during time 60-100 and 0-5, which indeed engulfs most of the arriving platoon, and is consistent with the earlier time-space diagram. It is helpful to understand that the saturation flow rate profile illustrates the effective green time instead of the actual green time. Effective green time is affected by the start-up lost time, actual green time, extension of effective green time, and queue spillback.



Although hill-climb optimization was used for this simple example, genetic algorithm optimization can be expected to produce better timing plans, when applied properly.

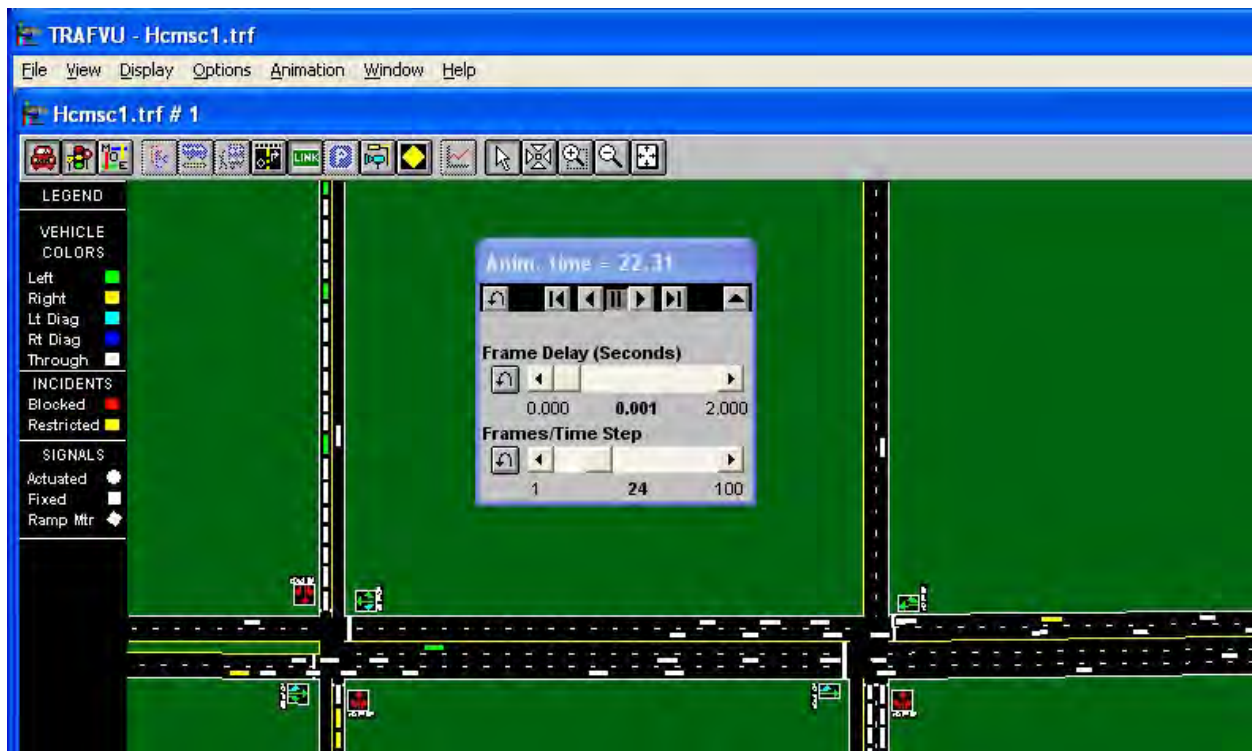
### HCM Example

This exercise begins with the same steps as the HCS file import tutorial:

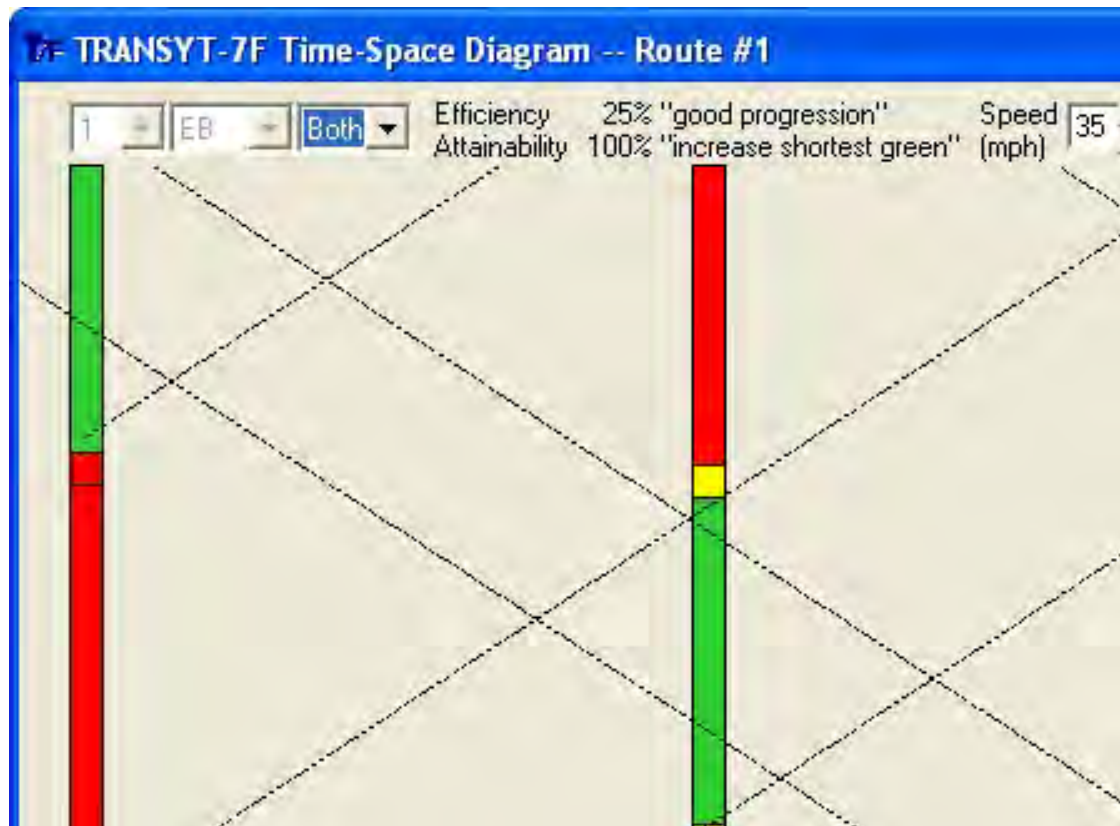
- (Options > Cruise Speed) Specify 35 mph
- (File > Open) Switch "Files of Type" to XHS, and open Hcmsg1.xhs
- (File > HCS > Append) Append Hcmsg2.xhs and Hcmsg3.xhs
- (Map View) Drag-and-drop nodes so they have 500' spacing, with node #2 in the middle
- (Map View) Drag a mouse rubber band around all three nodes, to select them all
- (Map View) Use the Map Properties icon (hotkey = "Z") to specify a Y-coordinate of zero
- (Map View) Select node #1, press "Z", and specify an X-coordinate of zero
- (Map View) Select node #2, press "Z", and specify an X-coordinate of 500
- (Map View) Select node #3, press "Z", and specify an X-coordinate of 1000
- (Map View) Draw a two-way link between nodes 1 and 2
- (Map View) Draw a two-way link between nodes 2 and 3
- (Map View) Drag a mouse rubber band around all three nodes, to select them all
- (Map View) Use the Progression Route icon (hotkey = "X") to specify an eastbound route
- (Edit > Timing) At node #3, make phase #1 non-actuated
- (Edit > Timing) At node #3, specify interval #2 as the reference interval
- (Edit > Analysis) Select a Performance Index (PI) of PROS/DI
- (File > Save) Save the input file
- (Run > TRANSYT-7F) Run optimization
- (File > Open) Switch "Files of Type" to PUN, and open Hcmsg1.pun
- (View > Time-Space Diagram)

Select "Both" to view progression bands in both the forward and reverse directions. Notice on the top of the screen that the two-way efficiency and attainability are reported as 16% and 61%, respectively. Notice also that it is not possible to increase efficiency or attainability (above 16% and 61%) by graphically adjusting the offsets using the mouse. Perfect one-way progression is easier to obtain and, for a peak-period rush hour, this might be preferable. However, perfect two-way progression is only possible when intersections are spaced relatively far apart.

If TSIS-CORSIM is installed on the computer, run one-touch animation by clicking on the "CORSIM Animation" icon. The amount of platoon progression seen in TRAFVU animation, and illustrated below, reflects what was shown in the TRANSYT-7F time-space diagram. That is, most vehicles traveling between node #1 and node #3 are able to pass through node #2 without stopping, but only some vehicles are able to pass through all three intersections without stopping.



To demonstrate perfect two-way progression, reload the original Hcmsg1.tin file. On the Edit > Lanes screen or the Edit > Traffic screen, re-code all link lengths as 2640 feet (one-half mile). In this case, re-coding link lengths on the Edit > Traffic screen would probably be faster. Use Disutility Index as the objective function, and run hill-climb cycle length optimization one more time. Then, reload the punch file Hcmsg1.pun, and load the Time-Space Diagram. Select "Both" to view progression bands in both the forward and reverse directions. The diagram below shows that progression is nearly perfect in both directions, with 100% two-way attainability. Perfect two-way progression is possible because at a speed of 35 mph, the travel time across 5280 feet is approximately equal to the cycle length of 100 seconds.



In the real world, perfect two-way progression is rarely feasible due to imperfect intersection spacings, cycle lengths, or travel speeds. However, the perfect progression design isn't as desirable as it may seem. Lengthy half-mile and one-mile intersection spacings make it difficult for vehicle platoons to stay together, such that progression timings become less beneficial. The ideal cycle length for progression often results in undesirable amounts of vehicle delay, especially on the cross streets. Finally, it is usually best for the oncoming platoon of vehicles to finish arriving several seconds before the downstream end green, so that the slower-than-average drivers do not arrive at the beginning of red. This is why PROS/DI optimization is frequently an effective strategy, because the delay due to inflated cycle lengths and/or standing queues can be taken into account by the DI.

## Queue Spillback and Queue Spillover Examples

This section of the documentation demonstrates the coding and analysis of spillback-related input data within TRANSYT. The "base case" or "base scenario" network is the one generated from scratch in the first exercise of the tutorial chapter, i.e., the four-node "box" shaped network. Alternatively, this file ("FourNode.tin" or "FourNode Lefty.tin") can also be obtained from the TRANSYT-7F Example folder. Note that the results from FourNode.tin throughout this exercise are sometimes slightly different than the ones from FourNode Lefty.tin, due to round-off errors caused by metric unit conversion.

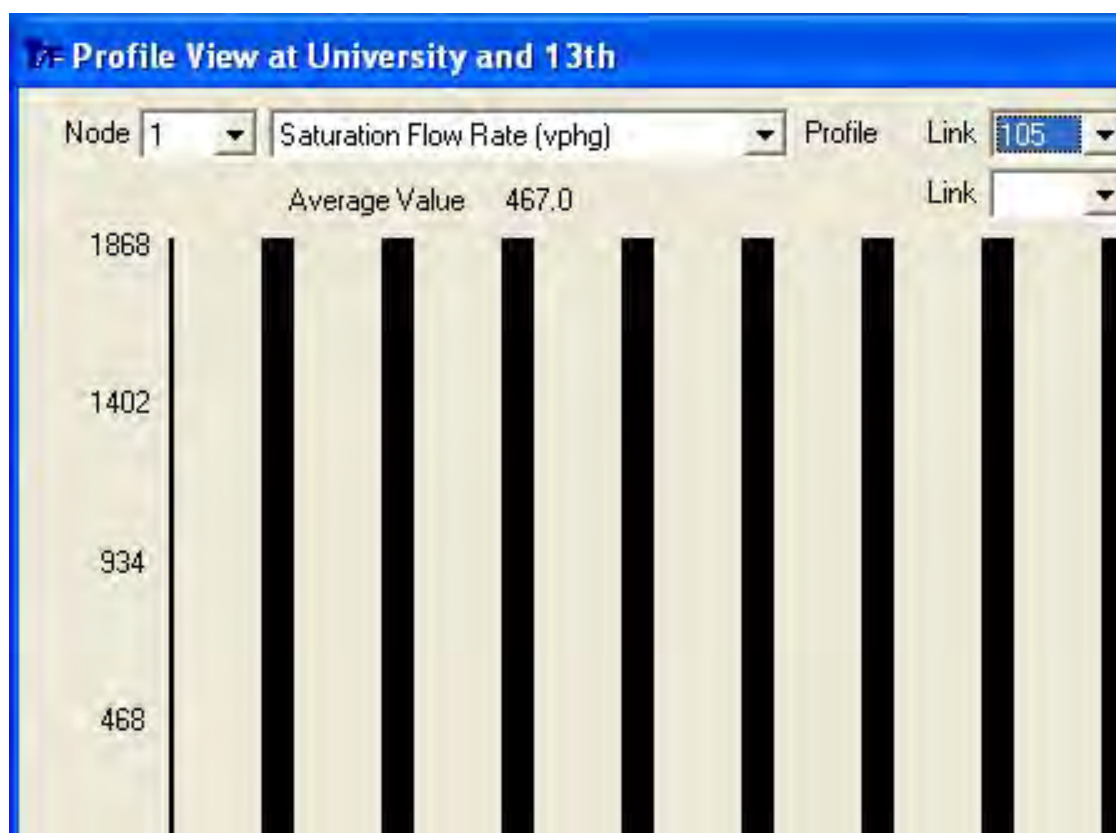
### Queue Spillback Analysis

The exercise begins by proceeding to the Edit > Analysis screen, and selecting "Multi-Cycle" simulation. This step is appropriate in preparation for an analysis of oversaturated conditions, even though the initial conditions are undersaturated. After running the model, notice that the average delay at node #1 for the eastbound through movement is reported as 70.6 seconds per vehicle, and average delay

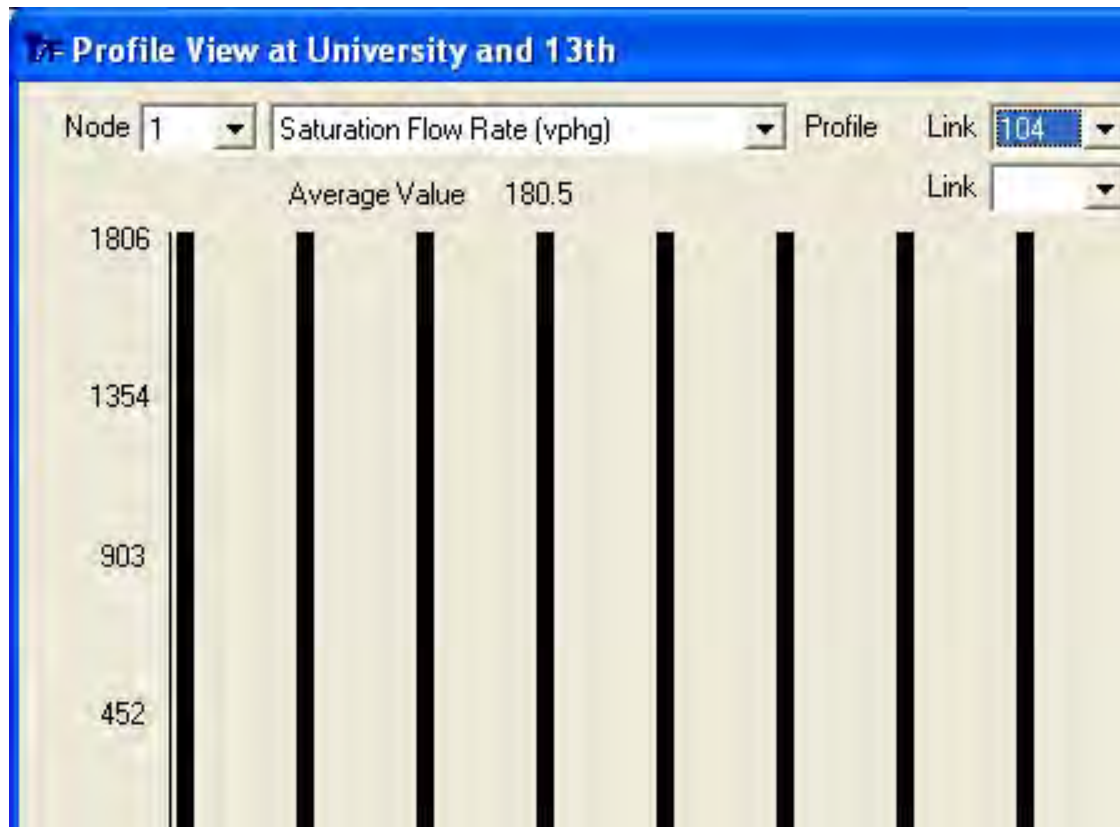


for the southbound left-turn is reported as 46.1 seconds per vehicle. (For left-hand drive, notice that the average delay at node #2 for the westbound through movement is reported as 70.6 seconds per vehicle, and average delay for the southbound right-turn is reported as 46.1 seconds per vehicle.) These results are similar to the ones illustrated in the screen image from the "sensitivity analysis" tutorial exercise, because single-cycle and multi-cycle simulation yield similar results when conditions are undersaturated.

After observing these initial results, close the report by selecting File > Close. The Profile View screen can be used to determine when a movement receives green, by examining the "saturation flow rate" profile. To observe this concept, the user should first run Spyglass by clicking on the "bar graph" icon, or selecting Run > Spyglass. After Spyglass finishes running, the user is prompted to load the Profile View screen. After loading the Profile View screen, switch to node #1 and link #105, which represents the eastbound major street through movement. (For left-hand drive, switch to node #2 and link #207, which represents the westbound major street through movement.) The screen image below illustrates that there are nine discrete regions of effective green time within the 15-minute analysis period, which is expected due to the 100-second cycle length.



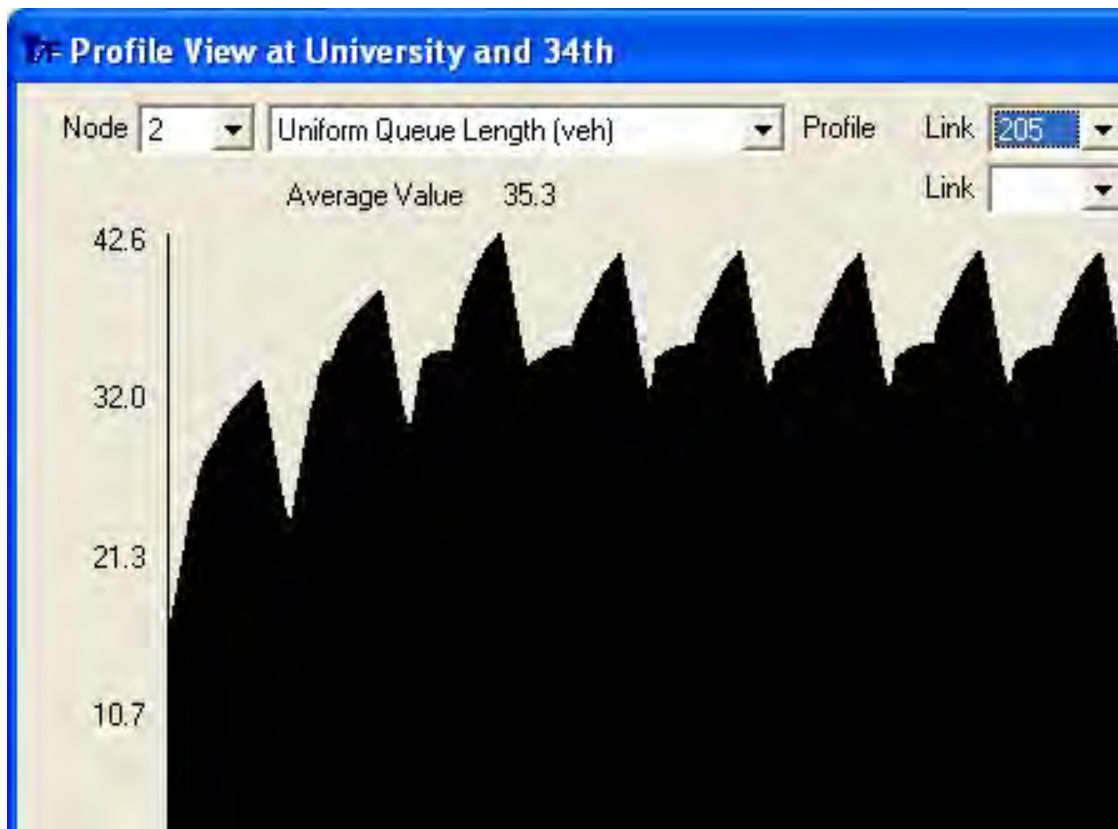
Next, switch to link #104, which represents the southbound minor street left-turn movement. (For left-hand drive, switch to link #210, which represents the southbound minor street right-turn movement.) The screen image below illustrates that there are again nine discrete regions of effective green time within the 15-minute analysis period, which is expected due to the 100-second cycle length. From the graph it is also apparent that the southbound left-turn receives less green time than the eastbound through movement, and that the southbound left-turn moves earlier in the cycle.



Then, proceed to the Edit > Traffic screen, and increase the eastbound through movement traffic volume at node #2 from 400 vph to 650 vph. Next, increase the eastbound through movement mid-block source volume at node #2 from 0 vph to 250 vph. (For left-hand drive, increase the westbound through movement traffic volume at node #1 from 400 vph to 650 vph. Next, increase the westbound through movement mid-block source volume at node #1 from 0 vph to 250 vph.)

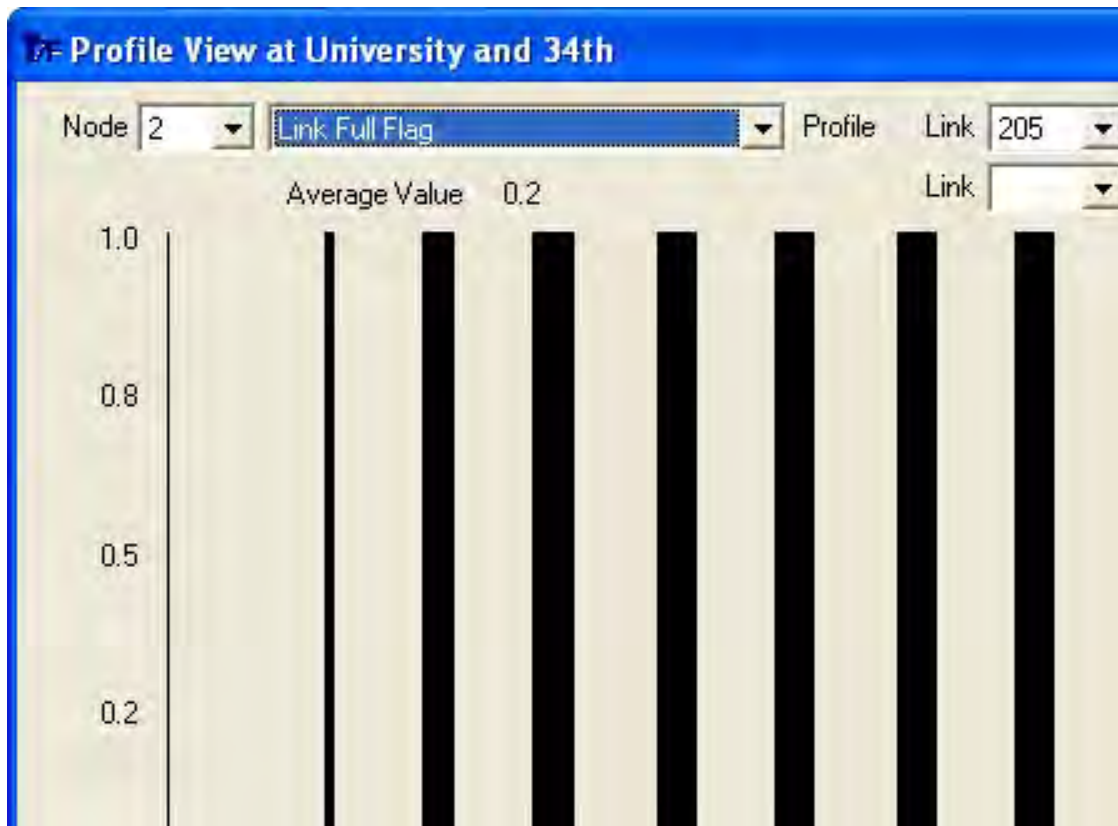
After saving this data and running the model, Flow (the number of vehicles discharged) for the downstream through movement is 530 vph. Because the original shared lane volume was 700 vph for this movement, it implies that not enough green time was available to serve the incoming traffic. Average delay for the upstream through movement remains 70.6 seconds per vehicle. However, average delay for the upstream southbound turn movement is now 377.8, which is much higher than the base case delay of 46.1 seconds per vehicle. At first glance, these results seem unrealistic, because queue spillback from the downstream node should have a similar impact on all feeding links from the upstream node.

The Profile View screen can be used to explain these results more clearly. First, run Spyglass once again. After Spyglass has finished running, load the Profile View screen once again. This time, go to node #2 and link #205, and select "Uniform Queue Length". (For left-hand drive, go to node #1 and link #107, and select "Uniform Queue Length".) The screen image below illustrates a residual queue that exists throughout the analysis period, and is never fully served.

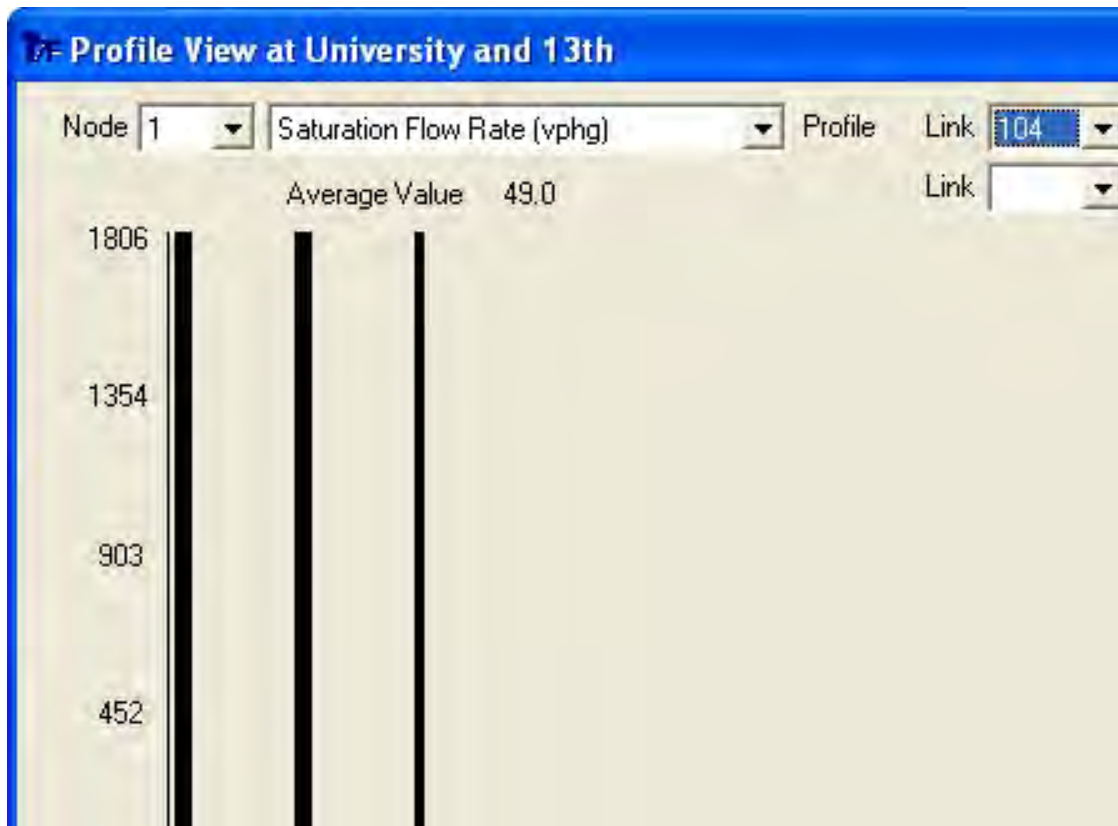


Next, select "Link Full Flag", again for link #205 (the eastbound through movement). (For left-hand drive, select "Link Full Flag", again for link #107 the westbound through movement.) The screen image below illustrates that after a few cycles of simulation, the entire link becomes filled for certain portions of the cycle.





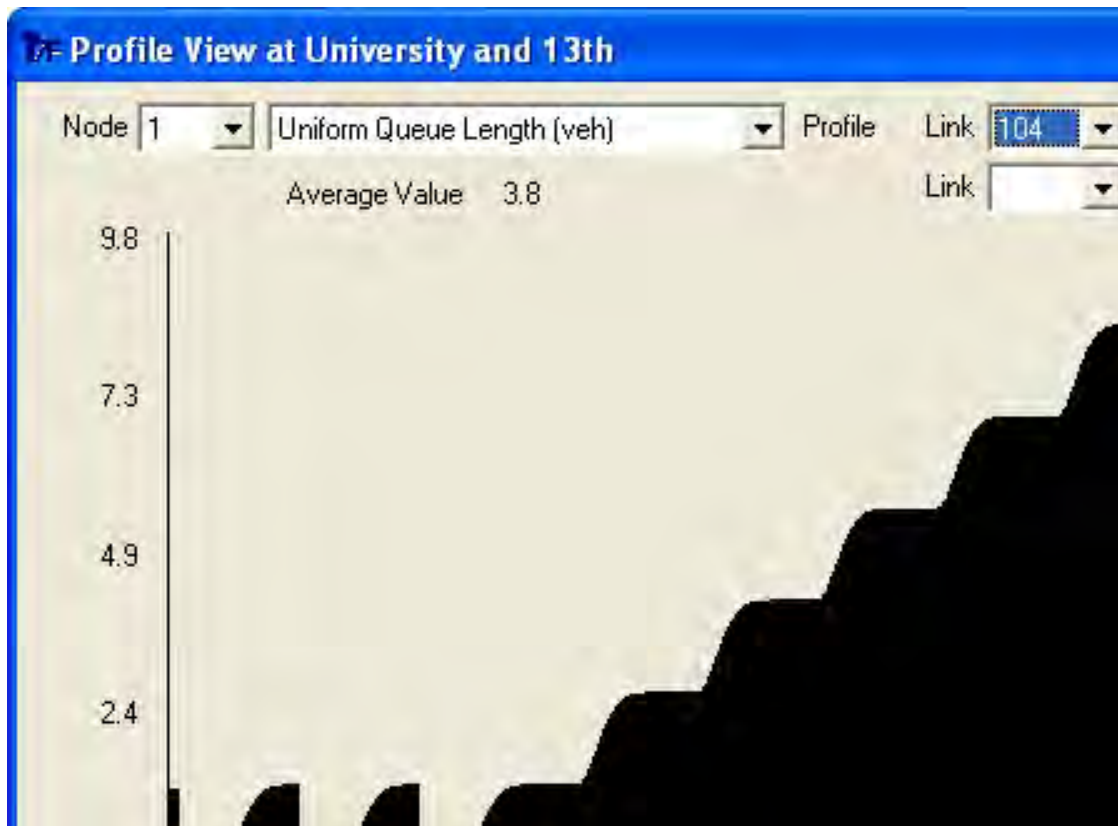
Next, switch back to link #105 at node #1, and select the Saturation Flow Rate profile. (For left-hand drive, switch back to link #207 at node #2, and select the Saturation Flow Rate profile.) This graph looks the same as it did before when conditions were undersaturated, because the downstream link was never full when this link was moving. Finally, switch to link #104. (For left-hand drive, switch to link #210.) The screen image below illustrates that the saturation flow rate profile, which represents the available capacity of the movement, becomes completely eliminated after a few cycles of simulation. This is because upstream link #104 receives green exactly when downstream link #205 is full.



As a final note, consider how the uniform queue length and link full flag both stabilize for the downstream through movement, link #205 (link #107 for left-hand drive). The earlier figures illustrate that once the downstream through movement has become saturated, the uniform queue length appears to stabilize at 42.6 vehicles, and the link is full for approximately one-quarter of the cycle. This result is further corroborated by observing Time Full% in the wide (132-column) output report, for the downstream through movement, which is reported as 23.6%. These conditions repeat themselves throughout each cycle, once the link has become saturated. The reason for this behavior is that residual queues are being passed on to the upstream feeding links.

Although the uniform queue length stabilizes at 42.6 vehicles for the downstream through movement, the maximum back of queue length stabilizes at a higher value of 48.4 vehicles. The maximum back of queue length can also be plotted using the Profile View screen. After multiplying 48.4 by the jam spatial headway (average vehicle spacing) of 25 feet (7.62 meters) per vehicle, the maximum back of queue apparently stretches 1210 feet (361 meters) upstream, which is approximately equal to the queuing capacity. According to the wide 132-column output file report, the queuing capacity for the downstream through movement is listed as 1175 feet (358 meters). The reason that the queuing capacity is somewhat lower than the link length is due to oversaturation, and the internal random queuing model.

To observe how residual queues have been passed on to the upstream feeding links, switch to the uniform queue length profile for the southbound left-turn movement at node #1 (link #104). (For left-hand drive, switch to the uniform queue length profile for the southbound right-turn movement at node #2, link #210.) The screen image below demonstrates that a residual queue is building on this link with each passing cycle, due to the oversaturation from downstream link #205.



Another issue that bears attention is the issue of average vehicle spacing. Queue spillback and queue spillover results are potentially very sensitive to the jam spatial headway and the optimal spatial headway. As an exercise, try coding a network wide average queue spacing of 20 feet per vehicle on the Edit > Optional > Global > Network Parameters screen. This essentially eliminates queue spillback for this sample network, because the queue length is now 20% shorter, and never disturbs the upstream signal.

Although these exercises involved simulation-only, queue spillback modeling is naturally very beneficial in the context of optimization. The level of detail within step-wise simulation allows TRANSYT-7F to devise effective timing plans for oversaturated conditions, or sometimes timing plans that will eliminate oversaturated conditions. One drawback is that multi-cycle simulation, which is preferable for accurately simulating oversaturated conditions, takes much longer on the computer than single-cycle simulation. One multi-cycle simulation run does not take very much time, but hill-climb and (especially) genetic algorithm optimization require hundreds if not thousands of individual simulation runs. Thus, the analyst must sometimes weigh his or her options in terms of running speed versus modeling accuracy.

#### Queue Spillover Analysis

At the outset of this exercise, it is necessary to ensure that the original "base scenario" conditions from the original sample file have been restored. Also, be sure that multi-cycle simulation is selected on the Edit > Analysis screen.

This exercise will involve generating queue spillover from the exclusive turning lane into the through lane, on the westbound approach to node #1. The first step is to go to the Edit > Traffic screen, and code a volume of 225 vph for link #108. Next, code a saturation flow rate of 1700 vph for link #108. Then, code a volume of 225 vph for link #107. The westbound approach volume is still 500 vph, but additional traffic has been shifted into the turn bay.

(For left-hand drive, this exercise will involve generating queue spillover from the exclusive turning lane into the through lane, on the eastbound approach to node #2. The first step is to go to the Edit > Traffic screen, and code a volume of 225 vph for link #211. Next, code a saturation flow rate of 1700 vph for link #211. Then, code a volume of 225 vph for link #205. The eastbound approach volume is still 500 vph, but additional traffic has been shifted into the turn bay.)

Next, proceed to the Edit > Feeders screen. Since the volumes have been changed for links #107 and #108, the feeding volumes for these links must be revised. This can be done manually, or by using "automatic flow balancing". Note that the cruise speed automatically coded by the software is taken from the Options > Cruise Speed screen, and should be set at 35 mph (56 kmph). To invoke automatic flow balancing, type in a zero for each of the feeding link number fields (under links #107 and #108 only). Click on the "Next Links" button to ensure that each feeding link number has been zeroed out. Then, on the top row of fields labeled "Upstream Node Number", type in '2' in the fields for both links #107 and #108. (For left-hand drive, on the top row of fields labeled "Upstream Node Number", type in '1' in the fields for both links #205 and #211.) The appropriate feeding link numbers, volumes, and cruise speeds should automatically be filled in by the software.

Finally, proceed to the Edit > Lanes screen. This screen should illustrate that a turn bay storage of 12 vehicles was previously coded for all exclusive turn pockets. Code a westbound turn bay storage of '0' or blank at node #1, in order to generate an initial set of results without queue spillover. (For left-hand drive, code an eastbound turn bay storage of '0' or blank at node #2, in order to generate an initial set of results without queue spillover.) Additional queuing data could be observed on the Edit > Optional > Queuing screen, if desired. This screen contains certain additional queue spillover-related data, some of which are automatically generated when turn bay storages are coded on the Lane Configuration screen.

After saving this data and running the model, average delay for the node #1 westbound left-turn is reported as 259.5 seconds per vehicle. Average delay for the node #1 westbound through movement is reported as 56.8 seconds per vehicle. Degrees of saturation for the westbound left-turn and through movement are 132% and 60%, respectively. Flow (number of vehicles discharged) for the westbound left-turn and through movement are reported as 225 and 274 vph, respectively. Time Full% for the westbound left-turn is reported as 0.0%.

(For left-hand drive, average delay for the node #2 eastbound right-turn is reported as 259.6 seconds per vehicle. Average delay for the node #2 eastbound through movement is reported as 56.7 seconds per vehicle. Degrees of saturation for the eastbound right-turn and through movement are 132% and 60%, respectively. Flow for the eastbound right-turn and through movement are reported as 225 and 274 vph, respectively. Time Full% for the eastbound right-turn is reported as 0.0%.)

What is happening here is that even though the turn bay is oversaturated, the model is not simulating any queue blockage. To invoke queue spillover, return to the Edit > Lanes screen, and restore the turn bay storage of 12 vehicles.

After saving this data and running the model, average delay for the node #1 westbound left-turn is reported as 206.9 seconds per vehicle. Average delay for the node #1 westbound through movement is reported as 58.6 seconds per vehicle. Degrees of saturation for the westbound left-turn and through movement are 108% and 62%, respectively. Flow (number of vehicles discharged) for the westbound left-turn and through movement are reported as 184 and 274 vph, respectively. Time Full% for the westbound left-turn is reported as 44.4%.

(For left-hand drive, average delay for the node #2 eastbound right-turn is reported as 207.1 seconds per vehicle. Average delay for the node #2 eastbound through movement is reported as 58.5 seconds per vehicle. Degrees of saturation for the eastbound right-turn and through movement are 108% and 62%, respectively. Flow for the eastbound right-turn and through movement are reported as 184 and 274 vph, respectively. Time Full% for the eastbound right-turn is reported as 44.4%.)

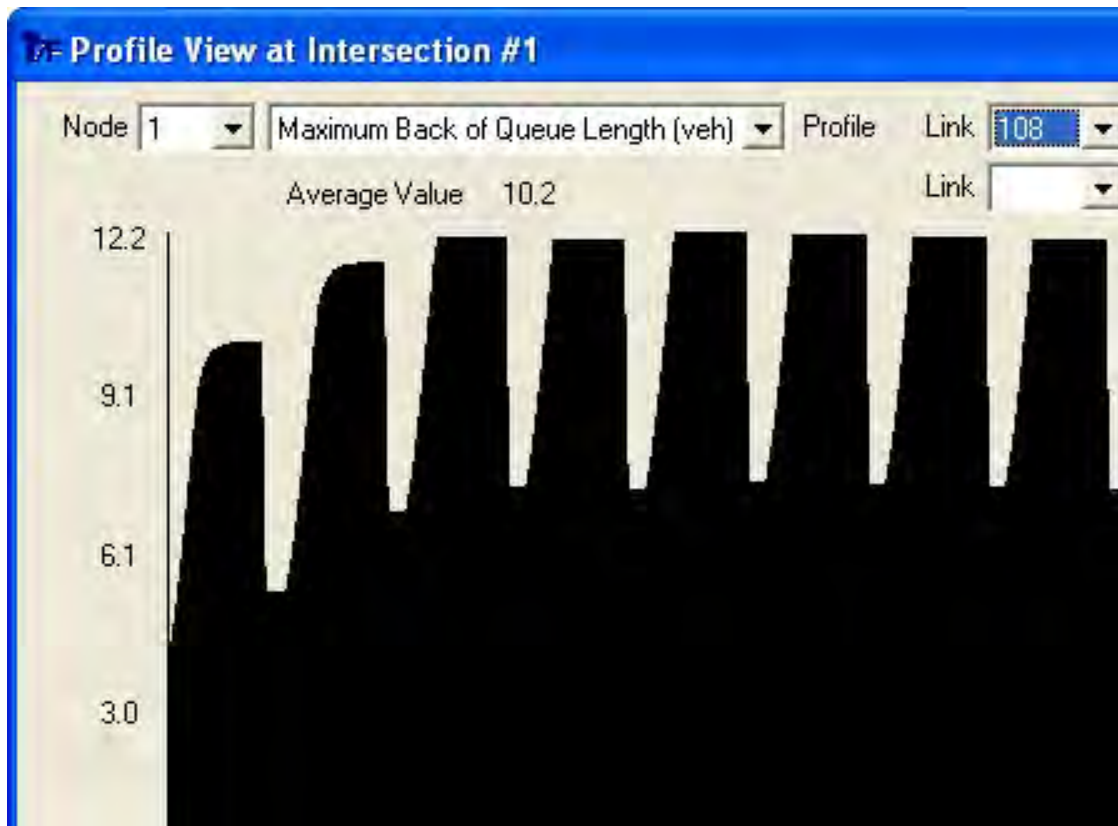
The model is now simulating queue spillover for the first time, and as a result, performance of the adjacent through movement is directly affected. Degree of saturation for the through movement has increased from 60% to 62% due to queue spillover. Delay has risen only slightly, from 56.8 to 58.6 seconds per vehicle (from 56.7 to 58.5 seconds per vehicle for left-hand drive), because degree of saturation (62%) isn't high enough yet to seriously inflate random delay (d2). Although the through movement is temporarily blocked, there is still enough green time for it to discharge all of its Flow, but the turn bay is unable to serve its full input demand.

Now consider the node-specific results. When the turn pocket length is not coded, average delay is reported as 72.4 seconds per vehicle for node #1. When the turn pocket length is coded, average delay is reported as 63.8 seconds per vehicle for node #1. Finally, when link-wise simulation is selected on the Edit > Analysis screen, average delay is reported as 68.8 seconds per vehicle for node #1.

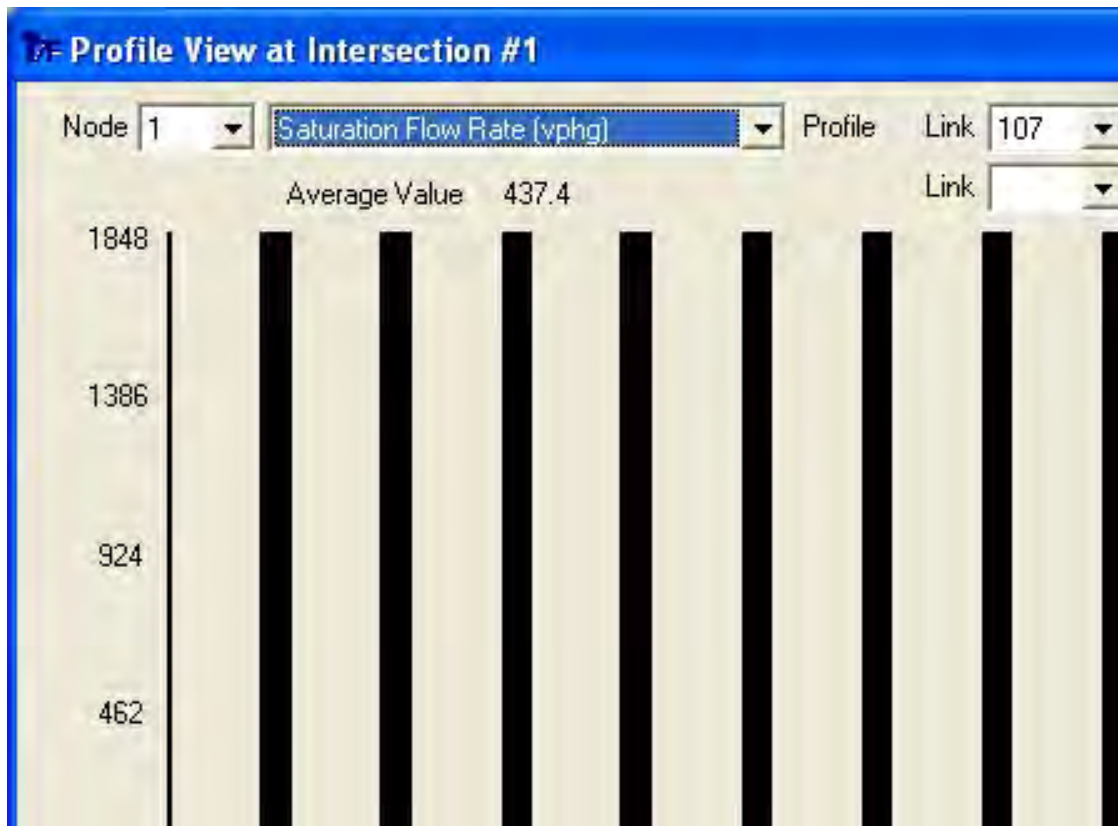
(For left-hand drive, when the turn pocket length is not coded, average delay is reported as 72.4 seconds per vehicle for node #2. When the turn pocket length is coded, average delay is reported as 63.8 seconds per vehicle for node #2. Finally, when link-wise simulation is selected on the Edit > Analysis screen, average delay is reported as 68.8 seconds per vehicle for node #2.)

These results illustrate that the queue spillover scenario is the most optimistic analysis, and probably the most realistic one. The queue spillover scenario may have been the most pessimistic scenario if the adjacent through movement had been congested by itself, to begin with. When using link-wise simulation, results at the adjacent and upstream links were unaffected by anything transpiring at the oversaturated turn bay, because link-wise simulation is not capable of simulating queue spillover or queue spillback. Instead, the vertically stacked queue in the turn pocket was allowed to grow unrealistically high, resulting in overestimated delay for that movement, and underestimated delay elsewhere in the network. This is one of the reasons that in general, link-wise simulation is no longer recommended for use within TRANSYT-7F, especially when any links are potentially oversaturated.

The Profile View screen reveals some interesting things about this analysis. Using the second scenario data file (containing a turn pocket storage of 12 vehicles), the user should first run Spyglass by clicking on the "bar graph" icon, or selecting Run > Spyglass. After Spyglass finishes running, the user is prompted to load the Profile View screen. After loading the Profile View screen, switch to node #1 and link #108, which represents the westbound left-turn movement. (For left-hand drive, switch to node #2 and link #211, which represents the eastbound right-turn movement.) Next, select the Maximum Back of Queue Length profile. The screen image below illustrates a residual queue length building up until it fills the turn bay (12 vehicles), at which time the queue length repeatedly dissipates and fills the turn bay again.



Next, switch to the Saturation Flow Rate profile of link #107, which represents the westbound through movement. (For left-hand drive, switch to the Saturation Flow Rate profile of link #205, which represents the eastbound through movement.) The screen image below illustrates that queue spillover occurs intermittently throughout the analysis period. Whenever a turn bay storage is coded in the Lane Configuration screen, an adjacent lane saturation flow rate reduction percentage is automatically generated on the Queuing screen. For only one adjacent through lane, this reduction percentage is 99%, by default. Looking carefully at the shapes below, it should be possible to tell when the 99% blockage caused by queue spillover is occurring. If desired, the Link Full Flag for the turn bay link can also be plotted, to shed further light on the situation.



In summary, when conditions are oversaturated, multi-cycle step-wise simulation is recommended for the most realistic results. When turn bays do have the potential to become oversaturated, queue spillover will not be simulated unless the turn bay storage is coded on the Edit > Lanes screen. By default, in the absence of any coded turn bay storage, queue spillback will be simulated instead.

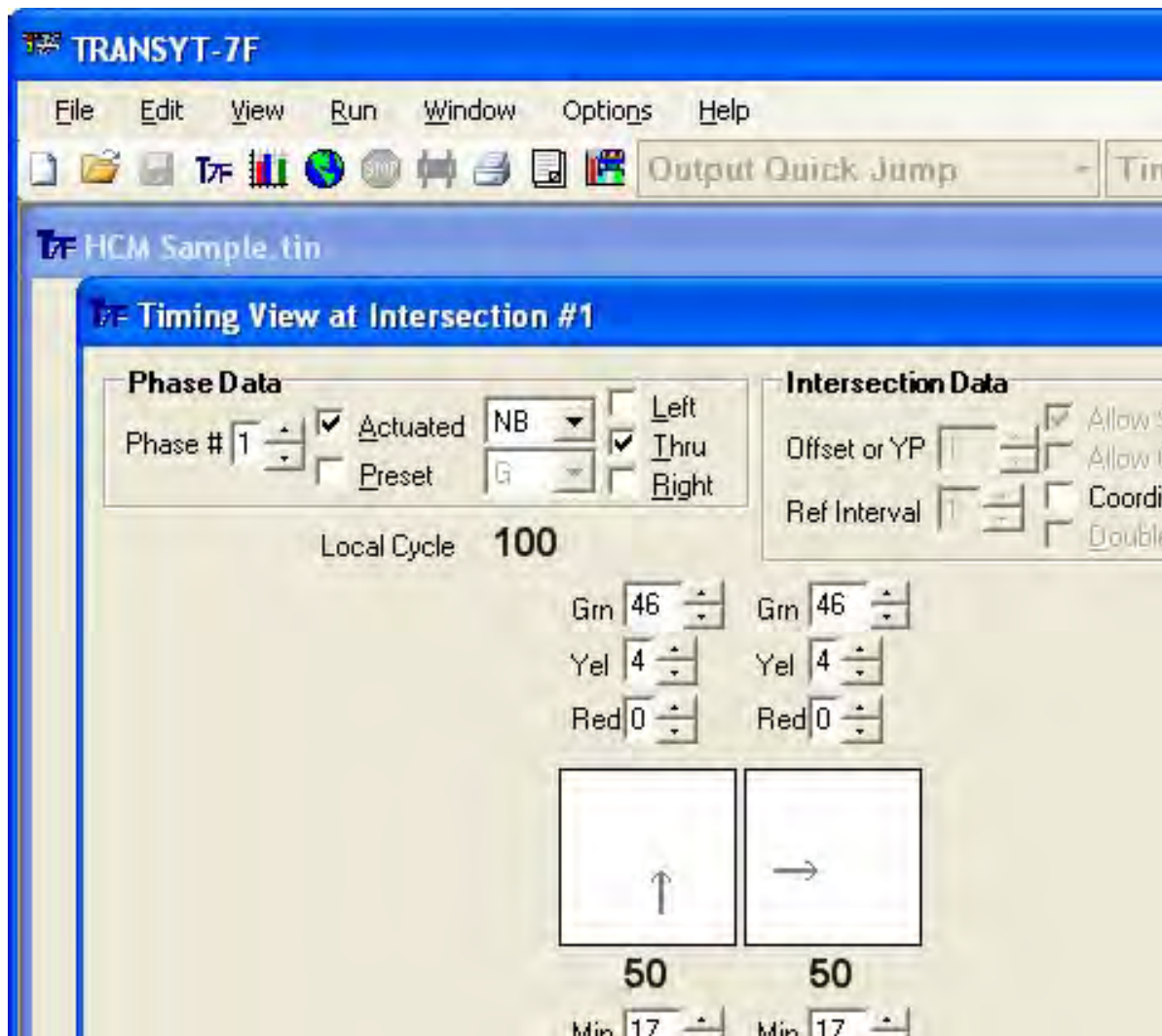
## Actuated Estimation Examples

### Example #1: evaluation of existing conditions, isolated intersection "HCM Sample.tin"

Suppose the maximum green settings were known for an isolated intersection in the field, but now the engineer wanted to know the average actuated phase times that were likely to occur. Knowing the average phase times is helpful not only for purposes of understanding, but also for purposes of applying software such as TRANSYT-7F and HCS, in which the analyst is required to code average phase times.

One solution would be to go out in the field and measure numerous phase times with a stopwatch, and compute the average value. Another solution would be to use the actuated estimation model within TRANSYT-7F. In the sample file "HCM Sample.tin" that is distributed with the software, there are only two phases in the timing plan (illustrated below). This sample data was taken from Highway Capacity Manual [12] Chapter 16, Appendix B. The phase times are coded as 50 seconds each, so this indicates that the maximum green times are 46 seconds apiece.





Proceed to the Edit > Optional > Actuated screen. For this example, the global and link specific gap settings should be 3.0 seconds. The detector lengths should be 30 feet (9 meters). All other settings can be left at their default values. Then, go to the Edit > Optional > Global > Network Parameters screen. The "network-wide approach speed for external links" should be 30 mph (48 kmph). Finally, proceed to the Edit > Analysis screen, and select "Estimation". After running the estimation model, the output file shows that the estimated actuated phase times are 36 seconds each, resulting in a cycle length of 72 seconds:

Node Split #1	Split #2
Init Final	Init Final
1 50 36	50 36

Note that the initial phase times of 50 seconds were coded for the purposes of running estimation, because these represent the maximum green times. However, for subsequently running a normal TRANSYT analysis, it would be preferable to specify the appropriate average phase times of 36 seconds. This would allow the user to obtain more realistic results in terms of queue lengths, delays, and so forth. One convenient way to obtain the average phase times after running estimation, without having to re-type them by hand, would be to load the "punch" file called "HCM Sample.pun".

Next, consider the effect of reducing the gap settings from 3.0 seconds to 2.0 seconds, on the



Edit > Optional > Actuated screen. As stated in the HCM, this reduction would be expected to reduce the green extension time for both phases, and to shorten the resulting cycle length. In this example, the cycle length is reduced from 72 seconds to 58 seconds.

#### Example #2: optimization, isolated intersection "HCMSC3.xhs"

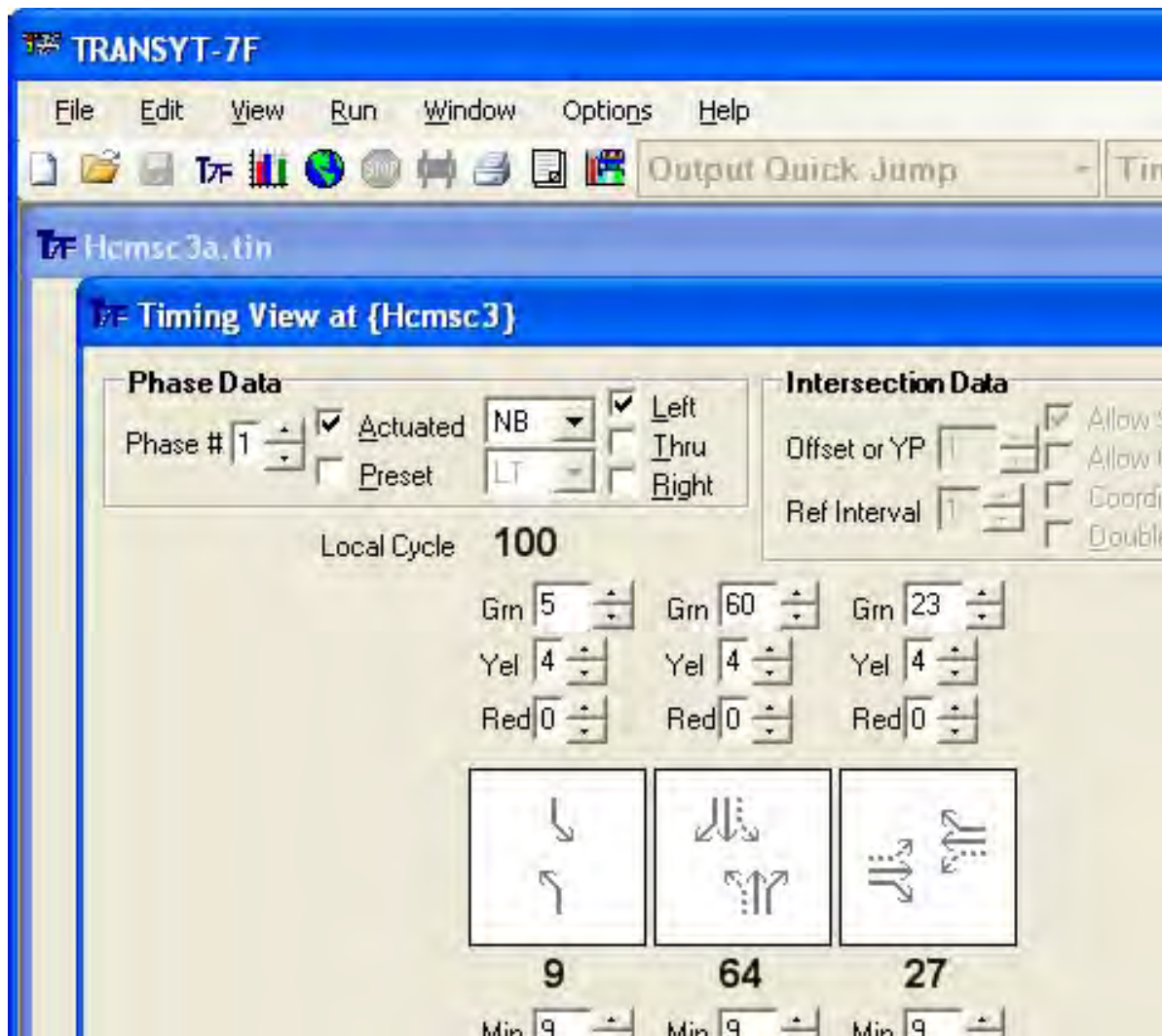
As stated in the Traffic-Actuated Control section of the documentation, the estimation procedure could be used to for two purposes, following a typical TRANSYT optimization run. First, it could be used to confirm that the optimized actuated phase times could indeed materialize in the field. Second, it could be used to determine maximum green and force-off settings for implementation in the field, to achieve the program-recommended optimum actuated phase times.

In this example, sample file "HCMSC3.xhs" is imported into TRANSYT-7F to create "HCMSC3.tin". This sample file is distributed with the software, and is based on sample calculation #3 from the Highway Capacity Manual [12] signalized intersections chapter. This exercise could also be performed using the metric sample file "MHCMSC3.xhs", from the Highway Capacity Software. After importing the HCS file, a cycle length optimization run is performed using the following settings on the Edit > Analysis screen:

- Genetic algorithm optimization of cycle length and splits
- Cycle length range of 80 to 100 seconds, with a search increment of 10 seconds
- In the "Genetic Algorithm Settings" dialog, maximum number of generations = 50, and mutation probability = 4.0% (accept all other default values)

Whenever genetic algorithm optimization is performed on a "small network" data file such as this one, it may be advisable to turn on "Rest Stops" to prevent the optimization process from running too quickly, which can cause memory and performance problems on the computer. Rest Stops can be activated under Options > Genetic Algorithm. Refer to the Genetic Algorithm Options section for more details. Regarding the genetic algorithm settings, a higher maximum number of generations and a lower mutation probability are usually more effective for optimizing larger networks. Refer to the Recommended G.A. Parameter Values section for more details.

Following optimization, load the punch file "HCMSC3.pun", and use File > Save As to select an arbitrary \*.TIN filename (e.g., "HCMSC31.tin") for performing subsequent actuated estimation runs. The Timing View screen below displays the results of the optimization run.



As stated in the Force-Off and Maximum Green section (chapter 4), it is usually best to set the force-offs and maximum greens equal to the optimum splits, which usually allows actuated phases to operate in the efficient 80%-90% degree of saturation range. The exception to this rule-of-thumb occurs when volumes are relatively high, and the resulting actuated phase times are 40 seconds or higher, as with phase #2 above. Under these conditions, force-offs and maximum greens can be a few seconds higher than the optimum splits, while still maintaining an efficient 80%-90% degree of saturation.

The output results below, obtained from three trial-and-error estimation runs, suggest that a maximum green time of 67 seconds is needed to achieve the optimum green time of 60 seconds for the middle phase. The phrase "trial-and-error" implies that the user must repeatedly code different maximum greens to see if they are successful in achieving the optimum phase times. For phases #1 and #3, conditions are such that their maximum greens can be set equal to their optimum green times of 5 and 23 seconds, respectively. Note that 4 seconds of yellow time were added on to the green times of 5, 67, and 23 to obtain the phase times (9, 64, and 27 seconds) shown below.

Node	Split #1	Split #2	Split #3			
	Init	Final	Init	Final	Init	Final
1	9	9	64	61	27	27

Node Split #1 Split #2 Split #3

	Init	Final	Init	Final	Init	Final
1	9	9	67	62	27	27

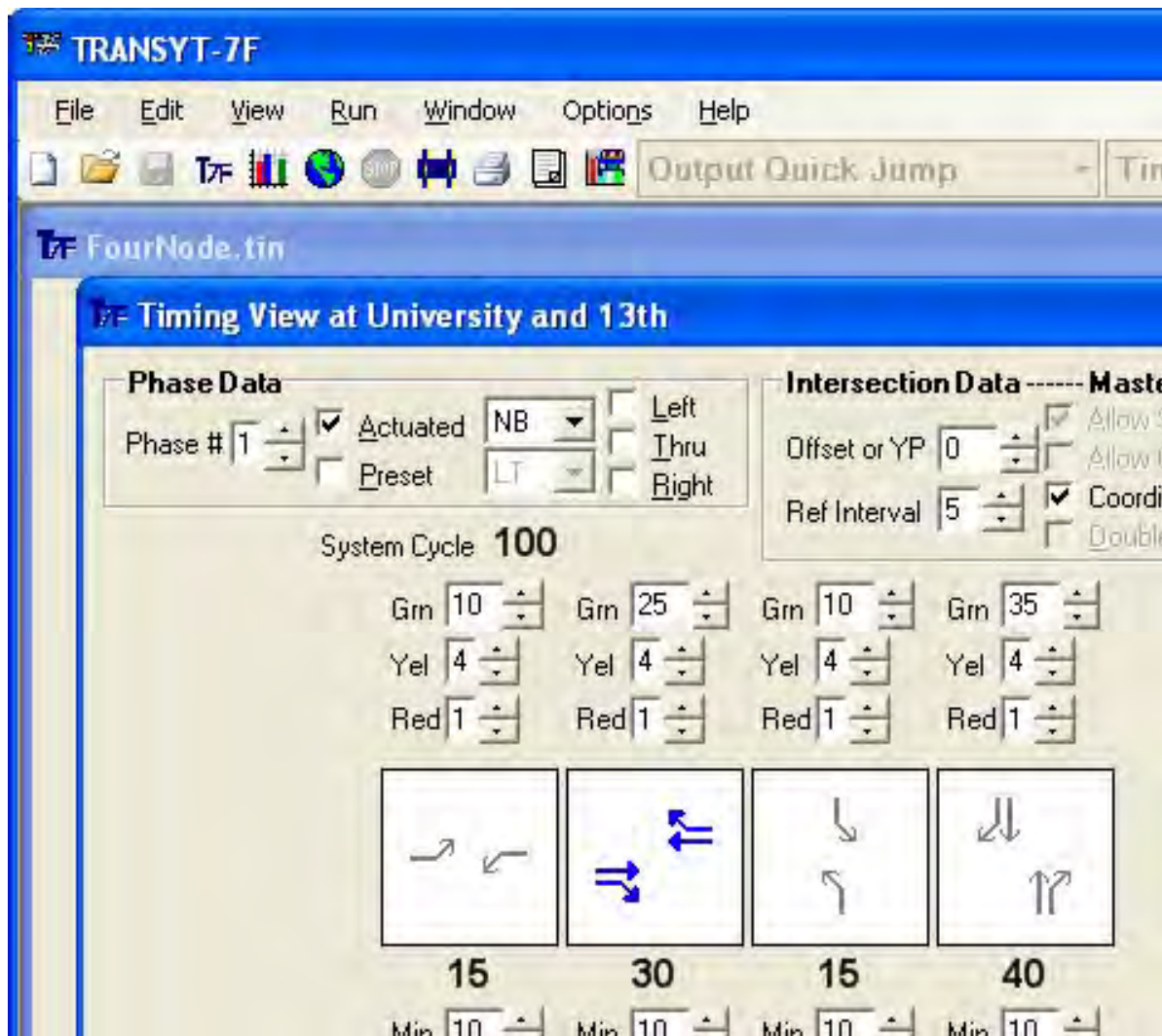
Node	Split #1	Split #2	Split #3			
	Init	Final	Init	Final	Init	Final
1	9	9	71	64	27	27

For each of these trial-and-error runs, after updating the maximum green time on the Edit > Timing screen, the background cycle length would then need to be updated accordingly on the Edit > Analysis screen, to avoid error messages. The user-coded initial background cycle lengths for these three runs would be 100, 103, and 107 seconds.

#### Example #3: evaluation of existing conditions, network "FourNode.tin"

The "base case" or "base scenario" network is the one generated from scratch in the first exercise of the tutorial chapter, i.e., the four-node "box" shaped network. Alternatively, this file ("FourNode.tin" or "FourNode Lefty.tin") can also be obtained from the TRANSYT-7F Example folder. Note that some of the intermediate calculation results for FourNode.tin within this exercise are slightly different than the ones from FourNode Lefty.tin, due to round-off errors caused by metric unit conversion.

This example is similar to actuated estimation example #1 except that the force-off settings are known in advance for the coordinated, semi-actuated system. In example #1, the maximum green settings were known in advance for the isolated, fully-actuated intersection. For this example, the force-off locations can be viewed by going to the Edit > Timing screen, as shown below. Notice that non-actuated phase #2 initially has less green time available to it than minor street through movement phase #4. However, once the unused green time is taken away from the actuated phases, the non-actuated phase time will be the longest (as expected).



Next, proceed to the Edit > Analysis screen, and select "Estimation" to prepare for the actuated estimation run. Run the model by pressing <Ctrl>-R, or by clicking on the blue "T7F" icon. The results are summarized in the output section shown below, showing that 8-9 seconds of unused green time are expected to materialize (during phase #2) on the major street at each intersection, during each signal cycle.

Node	Split #1		Split #2		Split #3		Split #4	
	Init	Final	Init	Final	Init	Final	Init	Final
1	15	13	30	39	15	13	40	35
2	15	13	30	39	15	13	40	35
3	15	13	30	38	15	13	40	36
4	15	13	30	39	15	13	40	35

#### Example #4: optimization, arterial street "Level2Perm.tin"

In this example, sample file "Level2Perm.tin" is optimized. This sample file can be obtained from the TRANSYT-7F Example folder. The data file contains 10 intersections with coordinated, semi-actuated control. Hill-climb cycle length optimization is performed between 60 seconds and 100 seconds, with a 10 second search increment. Following the optimization run, a cycle length of 70 seconds is recommended by the program. At this stage, the actuated estimation procedure can be used to verify

that the optimized timing plan could materialize in the field, and to determine the force-off settings for implementation in the field.

To proceed with the analysis, load the punch file "Level2Perm.pun", and use File > Save As to generate a new filename (e.g., "L2Perm1.tin") with a \*.TIN extension. Then, go to the Edit > Analysis screen and select "Estimation". After running TRANSYT-7F, the output listing below illustrates the estimation run results for the 10-node network.

Node	Split #1		Split #2		Split #3		Split #4		Split #5		Split #6	
	Init	Final	Init	Final	Init	Final	Init	Final	Init	Final	Init	Final
6	7	7	9	8	27	28	7	7	20	20		
7	7	7	5	5	39	39	7	7	3	3	9	9
8	50	50	10	10	10	10						
9	7	7	45	45	18	18						
10	7	7	3	3	42	42	7	7	3	3	8	8
11	7	7	5	5	26	26	7	7	3	3	22	22
12	7	7	3	3	28	28	7	7	3	3	22	22
13	7	8	3	3	20	20	7	7	3	3	30	29
14	7	7	3	3	23	23	37	37				
15	7	7	47	47	16	16						

Since this output listing was generated by the punch file, the initial trial-and-error force-offs were taken directly from the optimized splits. At most intersections (i.e., node numbers 7, 8, 9, 10, 11, 12, 14, 15), the output file shows that the average actuated phase times are identical to the force-offs, indicating that the actuated phases operate in the efficient 80%-90% degree of saturation range. Other intersections (6, 13) exhibit average phase times that are 1 or 2 seconds lower than the force-offs, meaning that the optimized splits could possibly occur in the field if their force-offs could be increased slightly.

To help determine whether or not the optimized splits at nodes #6 and #13 could materialize in the field, a second trial-and-error run could be performed using different force-off locations. Remember that the different force-off locations are specified by simply going to the Edit > Timing screen, and adjusting the initial green times for each phase. Although technically the coordinated non-actuated phase does not have a force-off setting, this initial green time must usually be lowered, in order to allow splits to sum to the cycle length.

## Multi-Period Analysis Examples

This section of the documentation demonstrates the coding and analysis of multi-period-related input data within TRANSYT-7F. The first "simple" example demonstrates the impact of residual queues on the results of a subsequent time period, at an isolated intersection. The second "detailed" example demonstrates the coding of a multi-period data file, along with a subsequent optimization run.

### Simple Example

Sample input files from the *McTrans* web site demonstrate how residual queues from one time period affect results in the next time period. The sample files for multi-period simulation are Example #7c.tin, Example #7d.tin, and Example #7e.tin.

The duration of each individual time period is specified on the Analysis screen, and the time period-specific traffic volumes are coded on the Traffic and Feeders screens. The user can switch between time periods for editing data, or specify (Options > Time Period > Select All) that certain data will be copied throughout all time periods. Most input data (roadway geometry, phasing sequence, optimization settings, etc.) are not expected to change between time periods. Because of this, many input fields and screens are automatically disabled when time periods #2 or higher are selected.

To load and edit the data for a specific time period, use either the Time Period Quick Jump at the top of the GUI, or Options > Time Period > Select. Subsequently, when going to an edit screen such as Traffic or Timing, these screens will only load and save the data for the previously selected time period. The Time Period Quick Jump combo box also controls the Output Quick Jump, which is capable of jumping to sections of the output file containing results for the desired time period.

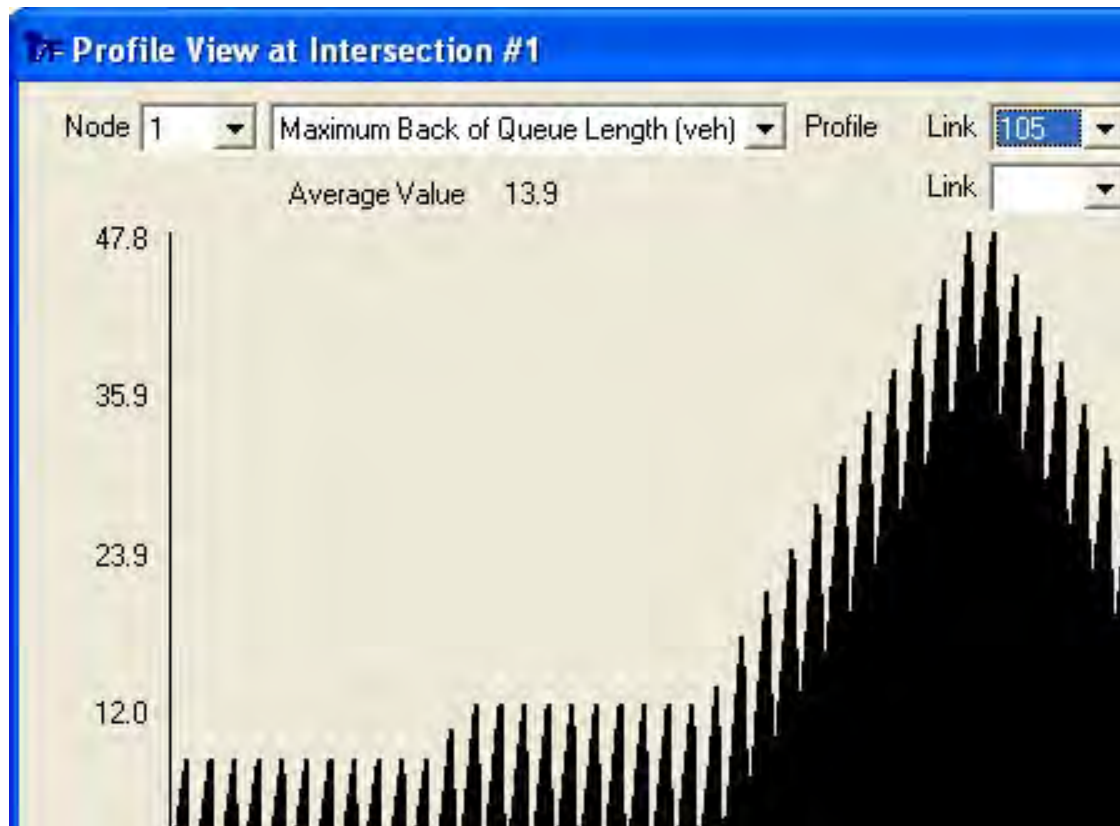
Results (in the tables below) for the eastbound approach demonstrate that although the volumes in time period #4 are the same in these two files, the delay and queue in time period #4 are higher in Example #7d.tin, due to the residual queues from time period #3.

Example #7d.tin	Volume	Capacity	Delay	Queue
Time Period Number				
1	150	221	20	10
2	200	220	34	17
3	250	231	80	62
4	200	230	82	51

Example #7e.tin	Volume	Capacity	Delay	Queue
Time Period Number				
1	150	221	20	10
2	200	220	34	17
3	200	231	30	16
4	200	230	30	16

Notice that the capacity varies slightly between time periods. This can be observed in the degrees of saturation from the legacy report, or in the Profile View screen. On average, the capacity for each time period is expected to be equal to  $s^*(g/C) = 1900*(38/80) = 902.5$  vph, or 226 vehicles per 15-minute period. However, multi-period simulation reveals that the capacity actually varies between time periods, because the cycle length (80 seconds) in this example is not evenly divisible into the time period lengths (900 seconds). Green time for the eastbound thru movement will sometimes straddle a time period boundary, and sometimes not.

The Profile View screen image below for Example #7d.tin illustrates a dissipating residual queue in the final time period, such that the average delay is higher than what it would have been without the preceding time period. Note that the total maximum back of queue length (50) listed above is slightly higher than the uniform maximum back of queue length (47.8) illustrated on the Profile View screen, due to the addition of random (a.k.a. "incremental") delay.



#### Detailed Example

The "base case" or "base scenario" network is the one generated from scratch in the first exercise of the tutorial chapter, i.e., the four-node "box" shaped network. Alternatively, this file ("FourNode.tin" or "FourNode Lefty.tin") can also be obtained from the TRANSYT-7F Example folder. Note that the results from FourNode.tin throughout this exercise are sometimes slightly different than the ones from FourNode Lefty.tin, due to round-off errors caused by metric unit conversion.

The first step in this exercise is to add time periods 2 through 4. To accomplish this, select Options > Time Period > Add three times in a row, and request the addition of time periods 2 through 4. This creates subsequent time periods containing the same input data as the first time period. Deleting time periods from a data file is similarly accomplished via Options > Time Period > Delete, although deleting the first time period data is not allowed.

After adding time periods 2 through 4, it would be possible to run the model now, but the system-wide results would be noticeably similar for each time period. This is because there are no links within the network with degrees of saturation above 100%. As such, multi-period analysis is considered unnecessary when conditions are undersaturated. Thus, oversaturated conditions and volumes must now be synthesized to facilitate this tutorial exercise.

To specify the time period-specific traffic volumes, select "Time Period 1" at the top of the TRANSYT-7F graphical user interface (GUI), and go to the Edit > Traffic screen. At each of the four intersections, code an eastbound through movement volume of 750 vehicles per hour (vph), and increase volumes for all other movements by 40 vph. For increasing all volumes by 40 vph simultaneously it may be easiest to use the "global spinner" on the right-hand side of the screen, and remember that it is possible to switch to the next node either by using the "Node Jump", or by pressing the <PageDown> key. At the nodes where these movements are internal links, code "mid-block source volumes" of 350 vph (for



the eastbound through movement) and 40 vph, respectively. To determine which links are internal, the Edit > Feeders screen or the Map View can be used. Afterwards, the same volume increases should also be coded for time period 2, but the volumes for time periods 3 and 4 should be left alone. The screen image below illustrates the updated volumes for time periods 1 and 2 at node #2.

Eastbound			Westbound			Northbound			Southbound	
Left	Thru	Right	Left	Thru	Right	Left	Thru	Right	Left	Thru
206	205	211	208	207	212	202	201	209	204	203
Total Volume (vph)										
90	750	90	90	440	90	90	440	90	90	440
Saturation Flow Rate (vphg)										
1805	1900	0	1805	1900	0	1805	1900	0	1805	1900
Link Length (ft)										
1320	1320	1320	1320	1320	1320	1320	1320	1320	1320	1320
Mid Block Source Volume (vph)										
40	350	40	0	0	0	0	0	0	40	40
Start-Up Lost Time (sec)										
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Extension of Effective Green Time (sec)										

The next step involves establishing an initial timing plan for optimization. Go to Options > Time Period > Select All. Then, go to the Edit > Timing screen. Change the timing plan at each intersection such that the non-actuated through movement has 35 seconds of green time, and the minor street through movement has 25 seconds of green time.

Next, run the model, and use the Output Quick Jump and Time Period Quick Jump to view the Global Output Summary report for each time period. The following time-period-specific delay results are reported:

- Time Period #1: 110 seconds per vehicle
- Time Period #2: 170 seconds per vehicle
- Time Period #3: 229 seconds per vehicle
- Time Period #4: 172 seconds per vehicle



This example will demonstrate the 9-step optimization strategy from chapter 2. The first step is to identify any intersections with inadequate laneage. That is, what if one or more intersections are guaranteed to be oversaturated due to insufficient laneage, regardless of signal timing? In this example network, although the oversaturated volumes ended in time period #2, it is apparent from the time period #3 results that it takes a while for the residual queues to dissipate. If there are intersections with inadequate laneage, it changes the signal timing approach. That is, it might be preferable to allow side streets to fail (so the major street can flow smoothly), or it might be preferable to optimize splits-only (because cycle length, phasing sequence, and offsets have a limited effect). Since all four of these intersections have identical volumes, it is only necessary to check one intersection. By right-clicking on any one of the four intersections in the Map View and selecting "View HCS results for this node", it is apparent that  $Y_c$  is equal to 0.84 in the HCS Capacity Analysis Worksheet. This means that enough laneage exists for the network to be undersaturated, if timed properly.

The second step of the optimization strategy is to select the analysis period duration and number of time periods. In this case, by modeling four 15-minute time periods instead of only one time period, it should be possible to compute delay and other measures accurately.

The third step of the optimization strategy is to simulate existing conditions, and ensure that the software is reasonably replicating field conditions. For this example, it is assumed that the initial set of results does indeed reasonably replicate field conditions.

The fourth step of the optimization strategy is to select an optimization objective function. For this example, it is assumed that the engineer wishes to minimize delay and stops by using the Disutility Index as their objective function.

The fifth step of the optimization strategy is to choose an underlying analysis engine or simulation engine. For this example, it is assumed that the engineer wishes to apply macroscopic multi-period simulation by using TRANSYT as their simulation engine.

The sixth step of the optimization strategy is to choose between sequential or simultaneous optimization of signal settings. For this sample network, the fact that there are only four intersections means that simultaneous optimization of signal settings should be practical. However, since left-turn volumes are identical everywhere, and progression opportunities are minimal with only two nodes at 1320-foot spacing, it would be reasonable to not optimize the phasing sequence.

The seventh step of the optimization strategy is to decide whether to use the initial timing model. Unfortunately, the initial timing model cannot be used in conjunction with multi-period optimization, although the initial timing model can be used in conjunction with multi-period simulation.

The eighth step of the optimization strategy is to decide which intersections to omit from the optimization node list. Since all four intersections are performing poorly when simulating field conditions, none of the intersections should be omitted from the optimization node list.

The ninth step of the optimization strategy is to choose an optimization search technique. In this case the genetic algorithm must be used, because the hill-climb method cannot optimize multi-period operations.

As a result of this nine-step optimization strategy, a multi-period genetic algorithm optimization of cycle length, splits, and offsets has been selected. As a reference point, a preliminary simulation run can be made using the initial timing model. To do this, proceed to the Edit > Analysis screen, switch back to time period #1, and select "Initial Timing Flag 1". After running the model, the time-period-specific delay results are as follows:

- Time Period #1: 71 seconds per vehicle
- Time Period #2: 79 seconds per vehicle

- Time Period #3: 69 seconds per vehicle
- Time Period #4: 42 seconds per vehicle

Delay is much lower now, because the initial timing model has successfully developed effective timing plans for the volumes in each time period. To prepare a multi-period optimization run, switch back to time period #1. Then, proceed to the Edit > Analysis screen, and code the following values:

- Genetic algorithm optimization of cycle length, splits, and offsets
- Cycle length range of 100-140 seconds, with a search increment of 10 seconds
- In the genetic algorithm settings dialog screen, specify a maximum number of generations of 100
- In the genetic algorithm settings dialog screen, specify a population size of 20

Note that a genetic optimization run involving 100 generations of multi-period simulation might require some extra time on the computer. Refer to the Genetic Algorithm Optimization section of the documentation for various timesaving tips. Also note that the initial timing flag selection on the Analysis screen is automatically disabled, when optimization is selected for the multi-period analysis file. After performing this optimization run, the following time-period-specific delay results are noted:

- Time Period #1: 66 seconds per vehicle
- Time Period #2: 75 seconds per vehicle
- Time Period #3: 55 seconds per vehicle
- Time Period #4: 41 seconds per vehicle

These optimization results are clearly better than the previous results in which only the initial timing model was used. The output listing below shows the results of the optimization run, which produces an optimal cycle length of 140 seconds. Because there is still some oversaturation remaining in the network, it would probably have been better to run more than 100 generations of optimization. According to the HCS sum of critical flow ratios ( $Y_c$ ), each intersection has adequate laneage to be undersaturated, if properly timed.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
 Agency:                      Analysis Time Period:  
 Jurisdiction:  
 Run Title: Four Node Sample File  
 File Name: FourNode TP.tin

Simulation Engine: TRANSYT

#### Genetic Algorithm Input Parameters

```

Crossover Probability (%):      30
Mutation Probability (%):      1
Convergence Threshold (%):     0.01
Maximum Number of Allowable Generations: 100
Actual Number of Optimization Generations: 100
Population Size:               20
Random Number Seed:           7781
Elitist Method:                True
Lower/Original/Upper Cycle Length (sec): 100/100/140
Objective Function:             Disutility Index
  
```

#### Optimization Results (Cycle Length, Splits, and Offsets)

```

Node Cycle   Offset NS Phasing EW Phasing
  
```

	Init	Final	Init	Final	Init	Final	Init	Final
1	100	140	0	0	---	---	---	---
2	100	140	0	129	---	---	---	---
3	100	140	0	85	---	---	---	---
4	100	140	0	70	---	---	---	---

Node	Split #1	Split #2	Split #3	Split #4				
	Init	Final	Init	Final	Init	Final	Init	Final
1	15	12	40	65	15	14	30	49
2	15	12	40	63	15	16	30	49
3	15	14	40	69	15	13	30	44
4	15	12	40	65	15	13	30	50

Control	Total	Fuel	Travel	Performance
Delay	Stops	Consumption	Time	Index
(sec/veh)	(%)	(gal/hr)	(veh-hr/hr)	

Time Period 1 (optimized)					
Initial	110	100	413	400	317.70
Final	65.5	102	324	277	232.40
Time Period 2 (optimized)					
Initial	170	102	535	568	440.50
Final	74.8	104	342	304	253.70
Time Period 3 (optimized)					
Initial	229	115	534	595	466.00
Final	55.0	93	225	184	150.40
Time Period 4 (optimized)					
Initial	172	115	435	461	368.10
Final	41.2	84	190	149	117.20
Time Horizon Summary					
Initial	170	108	479	506	398.07
Final	59.1	96	270	228	188.43

The final solution originated in generation #97  
 To obtain detailed output, process the punch file:  
 FourNode TP.pun

Suppose that the engineer desired to implement the results from this optimization run in the field. Because the timing plan is actuated, it may be helpful to run "actuated estimation", to verify force-offs for implementation. Since the actuated movement volumes were highest in time periods 1 and 2, either one of these time periods could be used. Open the punch file "FourNode TP.pun", and select File > Save As to choose a new file name (e.g., "FourNode TP1.tin"). Next, select Options > Time Period > Delete, and delete time periods 2 through 4. Then proceed to the Edit > Analysis screen, and choose "Estimation".

The output listing below shows the results of the estimation run. The results indicate that these force-offs may be directly implemented in the field (throughout all time periods), with little or no deviation expected from the optimized solution. For example, consider the splits (12, 65, 14, 49) at node #1. Since phase #3 is the first actuated phase following the yield point, the force-offs for field implementation (14, 63, and 75) are obtained by cumulatively summing the actuated phase times. Actuated phase times (and delays) are expected to be lower in time periods 3 and 4, using these same force-offs, due to the lower volumes in those time periods.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:	Date Performed:
Agency:	Analysis Time Period:
Jurisdiction:	
Run Title:	Four Node Sample File

File Name: FourNode TP1.tin

Queue Service Time Model: T7F  
Green Extension Time Model: GP1  
Global Gap Setting (sec): 3  
Global Detector Length (ft): 30

Iter	Node	Phase	Durations	Control Delay
5	1	12 65 14 49 0 0 0	66.5	
	2	12 63 16 49 0 0 0		
	3	14 69 13 44 0 0 0		
	4	13 65 13 49 0 0 0		

Convergence has been achieved  
To obtain detailed output, process the punch file:  
FourNode TP1.pun

Node	Split #1	Split #2	Split #3	Split #4				
	Init	Final	Init	Final	Init	Final	Init	Final
1	12	12	65	65	14	14	49	49
2	12	12	63	63	16	16	49	49
3	14	14	69	69	13	13	44	44
4	12	13	65	65	13	13	50	49

Node 1 (Iteration 5) University and 13th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
WB Left	68	5	1.00	999	3.00	5	13	12	10	
Non-Act	80	5	0	0	0	5	65	110	65	
NB Left	5	5	1.00	8.87	3.00	5	14	14	10	0
NB Thru	19	5	0.98	39.7	3.38	5	48	49	10	

Node 2 (Iteration 5) University and 34th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	59	5	1.00	8.77	3.00	5	12	12	10	
Non-Act	71	5	0	0	0	5	63	110	63	
NB Left	134	5	1.00	8.76	3.00	5	16	16	10	129
SB Thru	10	5	0.98	40.7	3.20	5	49	49	10	

Node 3 (Iteration 5) Archer and 13th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	7	5	1.00	8.87	3.00	5	14	14	10	
Non-Act	21	5	0	0	0	5	69	110	69	
NB Left	90	5	1.00	8.95	3.00	5	13	13	10	85
SB Thru	103	5	0.98	999	3.00	5	44	44	10	

Node 4 (Iteration 5) Archer and 34th

Critlink	Start	MnG	Fq	Gs	Ge	Y+R	Split	Max	Min	Offset
EB Left	137	5	1.00	8.78	3.00	5	13	12	10	
Non-Act	10	5	0	0	0	5	65	110	65	
NB Left	75	5	1.00	8.90	3.00	5	13	13	10	70
NB Thru	88	5	.984	41.2	3.19	5	49	50	10	

## Five Approaches or More

To simulate an intersection with 5 approaches or more, this requires using link numbers 13 and higher. This is also documented to some extent in the Link Numbering Scheme section of the documentation. To code the extra links, go to the Edit > Traffic screen and click on "Next Links". It is then possible to code traffic volumes for up to 50 links. As illustrated in the screen image below, the sample file "Five Approach.tin" contains links 113 and 114 on the fifth approach. Next, to code the timing

plan for a complex intersection having 5 approaches or more, go to the Edit > Optional > Timing screen. On the Edit > Feeders screen, it would be possible to specify link connection information for links 113 and 114, if necessary.

Intersections with five approaches (or more) can have their phasing sequence optimized, but only on approach numbers 1 through 4 (northbound, southbound, eastbound, westbound). If the movements from approach #5 (and higher) do not move in their own exclusive phase, phasing sequence optimization would require the coding of "associated links". Associated links can be coded on the Edit > Optional > Global > Associated Links screen. Refer to the Global Screen section of the documentation (in chapter 3) for more details. In the sample file Five Approach.tin, links 113 and 114 both move in their own exclusive phase; so for this network, phasing sequence optimization could be performed without coding associated links.

**TRANSYT-7F**

File Edit View Run Window Options Help

Output Quick Jump

**Five Approach.tin**

**Traffic at University and 13th**

Link	Link	Link	Link	Link	Link	Link	Link	Link	Link	Link
118	117	123	120	119	124	114	113	121	116	115
Total Volume (vph)										
0	0	0	0	0	0	100	50	0	0	0
Saturation Flow Rate (vphg)										
0	0	0	0	0	0	1800	1800	0	0	0
Link Length (ft)										
0	0	0	0	0	0	1320	1320	0	0	0
Mid Block Source Volume (vph)										
0	0	0	0	0	0	0	0	0	0	0
Start-Up Lost Time (sec)										
2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Extension of Effective Green Time (sec)										

## Sign Control Examples

This section of the documentation demonstrates the coding and analysis of sign control-related

input data within TRANSYT-7F. The first example demonstrates the creation of an isolated intersection data file, along with preliminary analysis of results. The second example demonstrates analysis of an unsignalized intersection that experiences platooned vehicle arrivals from nearby upstream signals.

### Example #1

The screen image below illustrates the initial settings for the new sign control data file. This includes setting the number of intersections equal to 1, the default traffic volume equal to 300 vph, and the default number of phases equal to 2. All other settings are left alone.

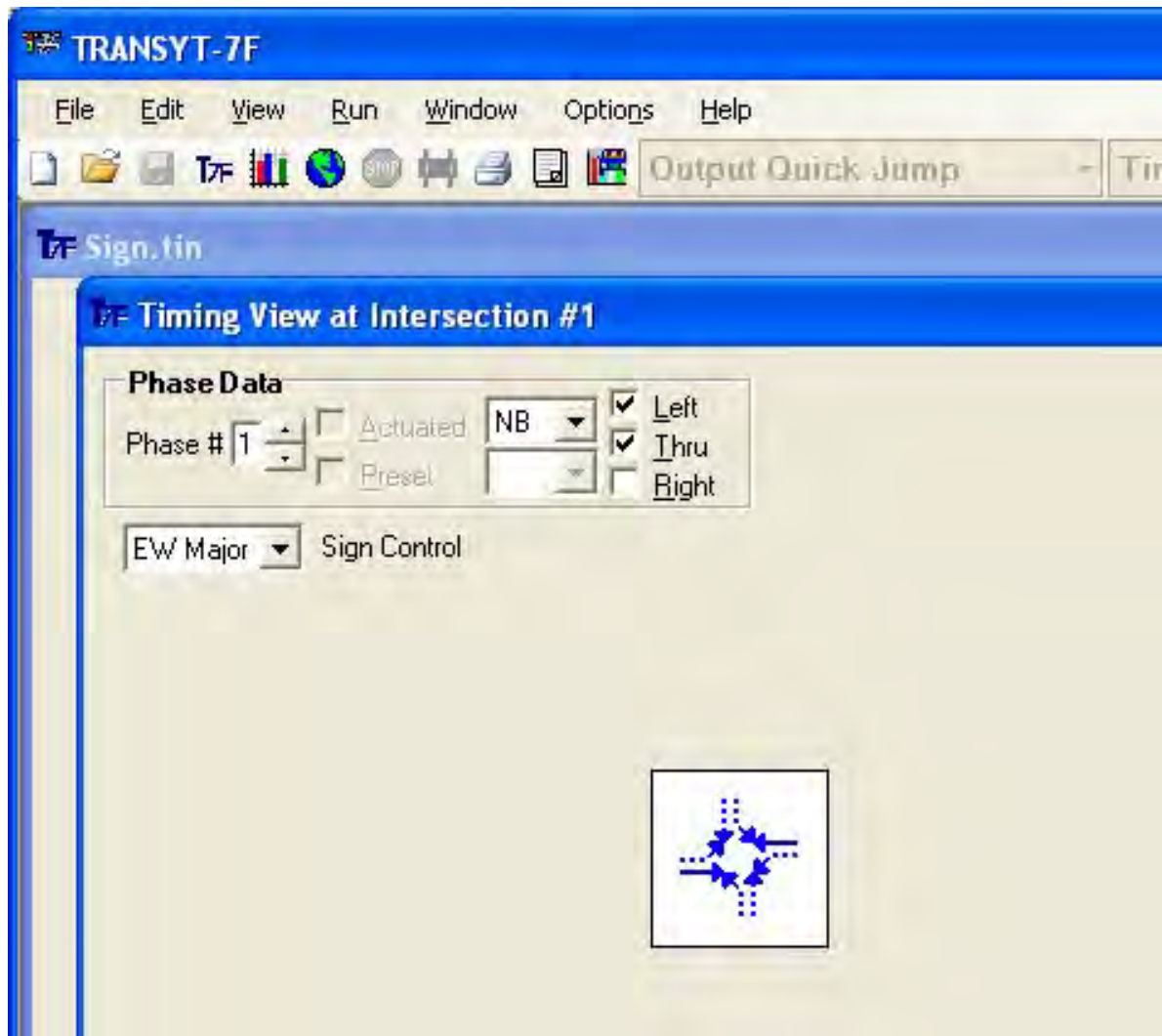
File Name	Sign.tin	Default Traffic Volume (vph)	300
Run Title	Sign Control Example #1	Default Peak Hour Factor	1.0
Link Number Scheme	TRANSYT-7F Standard	Default Saturation Flow Rate (vph)	1900
Number of Intersections	1	Default Link Length (ft or m)	100
Number of Routes	0	Default Free-Flow Speed (mph or km/h)	35
Cycle Length (sec)	100	Default Number of Phases	2
Duration of Analysis (min)	15	Default Yellow Time (sec)	4.0
Number of Time Periods	1	Default All-Red Time (sec)	3.0
<input checked="" type="checkbox"/> English Units <input checked="" type="checkbox"/> Right-Hand Drive		Default Gap Setting (sec)	2.0
		Default Detector Length (ft or m)	100

Since this will be an unsignalized intersection, the cycle length (and other timing settings) will have no effect on results. In addition, the default saturation flow rate will have no effect on results, although the unsignalized "maximum flow rate" will be edited later on. Finally, the default number of phases should be set to its minimum value of 2, although later on the second phase will be deleted in order to create a single-phase, unsignalized node.

After clicking on "OK" to generate the brand-new data file, select File > Save in order to save the file. The next step is to delete the secondary shared turning movements that were automatically generated when creating the new file. This can be done on either the Edit > Lanes screen, or the Edit > Traffic screen. On the Lane Configuration screen, assuming right-hand drive, one would select lane #2 on each approach and switch the lane use from "TR" to "T". On the Traffic screen, unwanted links are deleted by coding a volume of zero. Next, proceed to the Edit > Timing screen.

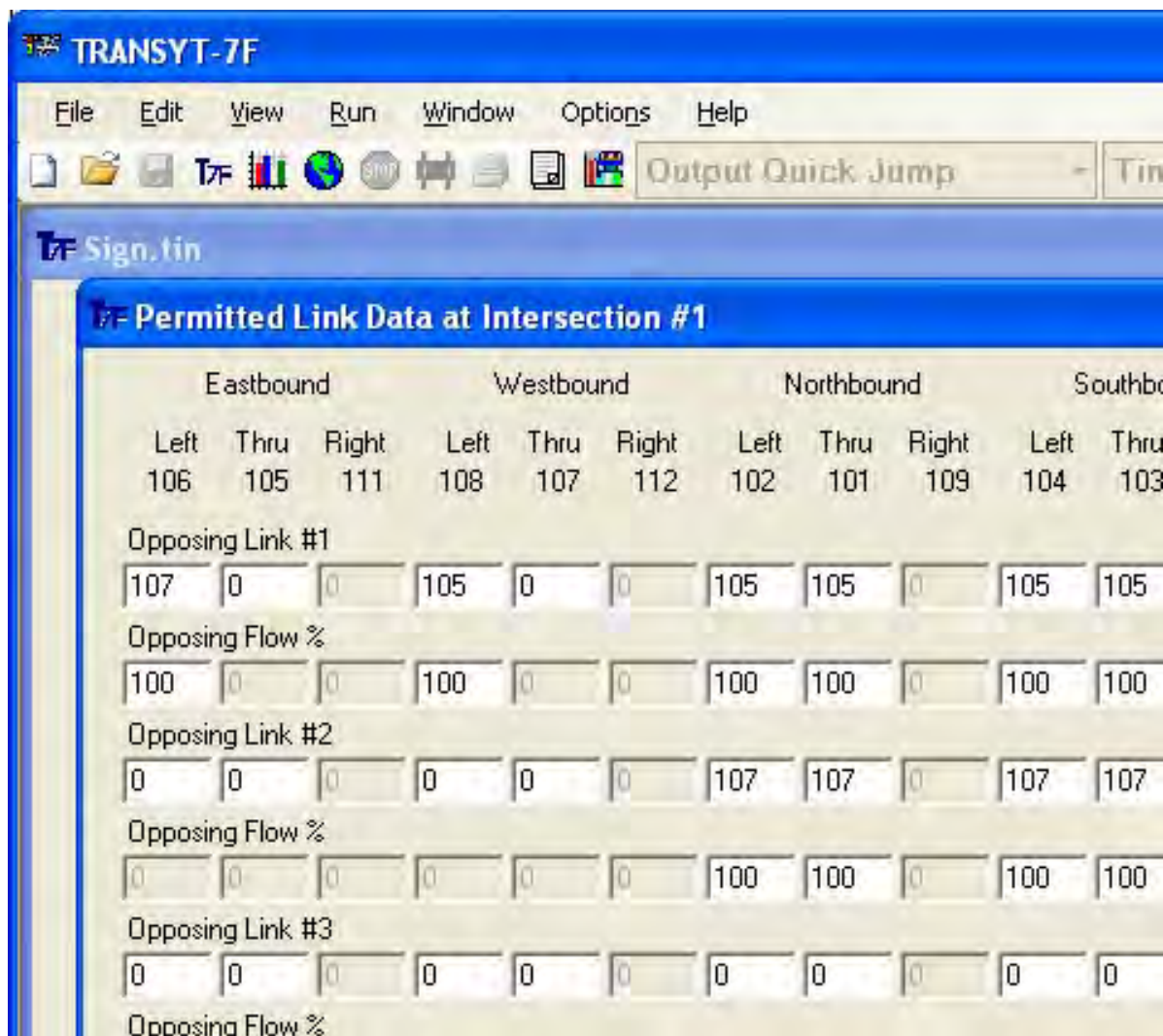
For this example, it is assumed that the "major street" direction of the two-way stop controlled intersection is the East-West direction. It will be necessary to code all traffic movements moving in phase #1, and then to delete phase #2. First, use the software controls within the "Phase Data" frame to code all traffic movements moving in phase #1. Next, add a single movement to phase #2, and then immediately delete this same movement from phase #2. This should automatically delete phase #2. Finally, make sure that the major street combo box that appears in the upper left-hand corner of the

Timing View screen says "EW Major". The completed Timing View screen is shown below.



Notice that The North-South minor street through movements are permitted movements that must yield to the East-West major street through movements. Proceed to the Edit > Optional > Permitted screen. On this screen, it is necessary to code the opposing link numbers for the stop-controlled minor street movements. For right-hand drive, the minor street movements are links 101, 102, 103, and 104. For left-hand drive, the minor street movements would be links 101, 109, 103, and 110. TRANSYT does not allow an opposing link to be a permitted link itself. Because of this, the opposing links for all minor street movements should be coded as the major street through movement links (e.g., links 105 and 107). Internally, the program will use the link numbering scheme to determine which minor street links will be subjected to the "unsignalized cross street left" model, and which ones will be subjected to the "unsignalized cross street through" model. The completed input screen is shown below.





It should now be possible to run the model (either by clicking on the blue "T7F" icon, or by pressing <Ctrl>-R) and obtain initial results. Assuming right-hand drive, the northbound left-turn degree of saturation is reported as 63%. Normally, degree of saturation is affected by the saturation flow rate coded on the Edit > Traffic screen (or record type 28). However, for sign-controlled links, degree of saturation is affected by the maximum flow rate within the permitted movement model. A copy of the completed data file (called "Sign.tin") is also available within the TRANSYT-7F Example folder.

To observe how these results are computed, select Run > Spyglass, and load the Profile View screen. Plot the saturation flow rate profile for link 102, the northbound left-turn movement. For left-hand drive, plot the saturation flow rate profile for link 109, the northbound right-turn movement. Notice that the saturation flow rate is 474 vphg during each second of simulation. This saturation flow rate value was derived from the "unsignalized cross street left" model for 2+ opposing lanes,  $MFR = 600 - 0.21 * V_o$ . Since the total opposing volume on the major street was 600 vph, the maximum flow rate for the northbound turning movement was computed as  $600 - 0.21 * 600 = 474$ . Refer to the Permitted Movement Methodology section of the documentation for more details.

Suppose the analyst wanted to calibrate the northbound turning movement to make the results somewhat more pessimistic. To do this, return to the Edit > Optional > Permitted screen, and then click on "Next Data". Code a maximum flow rate of 500 vph for the northbound turning movement. After



running the model, degree of saturation for this movement has increased to 80%. Next, run Spyglass again and plot the saturation flow rate profile for link 102 (or link 109, if left-hand drive), using the Profile View screen. Notice that the saturation flow rate is now 374 vphg during each second of simulation. The maximum flow rate for the northbound turning movement is now being computed as  $500 - 0.21 * 600 = 374$ .

This calibration technique could also be used to emulate permitted shared lane operation at a sign-controlled intersection. This is because TRANSYT does not allow sign-controlled links to be coded on the Edit > Optional > Global > Shared Lanes screen (or record type 7). Therefore, the maximum flow rate could be reduced for a permitted shared lane group in order to achieve more pessimistic and realistic results. The global unsignalized model coefficients could also be calibrated on the Edit > Optional > Global > Model Coefficients screen (or record type 8), to make results more pessimistic, but this adjustment would apply to all unsignalized links in the network. The maximum flow rate (on the Edit > Optional > Permitted screen) can be calibrated for individual links, and does not need to apply globally.

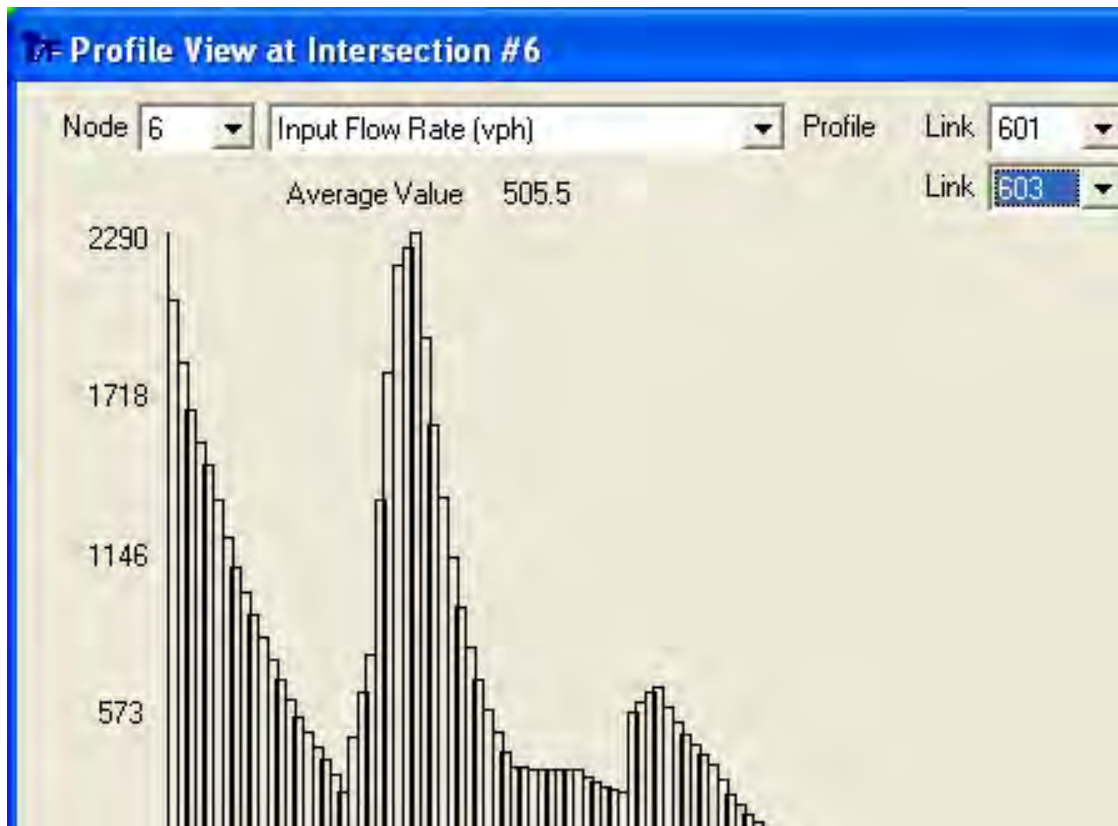
### Example #2

This second example demonstrates analysis of an unsignalized intersection that experiences platooned vehicle arrivals from nearby upstream signals, using sample file Daytona.tin. This file can be obtained from the TRANSYT-7F Example folder.

After loading the file Daytona.tin, select Edit > Timing, and switch to intersection #6. This node is an unsignalized intersection having stop signs on the east and west approaches. Next, close the Timing View screen and then run the model. When viewing the link performance output at node #6, degrees of saturation are somewhat higher for the minor street movements in the east-west direction, as expected.

Next, select Run > Spyglass, and load the Profile View screen. By viewing the input flow rate profile, it is possible to observe that every link at node #6 is experiencing platooned vehicle arrivals from the upstream signals. Next, switch to the saturation flow rate profile. Notice that major street through movement links 601 and 603 experience a constant saturation flow rate of 3400 vphg, which indicates that the minor street movements possess 4 opposing lanes.

However, the saturation flow rate profiles for all other links are non-constant. The permitted links are able to discharge the majority of their vehicles after the conflicting platoons have passed by. The Profile View screen below illustrates those times within the flow profile when the major street platoons are arriving, and capacities for the permitted links would be reduced.



## Lane-by-Lane Analysis Example

Lane-by-lane analysis becomes helpful when there are significantly different volumes or queue lengths in each lane. Under these conditions, if a lane-by-lane analysis were not performed, the model would only be capable of providing a single delay and queue length estimate for the entire lane group. A lane-by-lane analysis can provide more realistic results by modeling each lane separately. The disadvantage of lane-by-lane analysis is that more input coding is required.

The simplest case of lane-by-lane analysis is a situation in which there are no shared lanes. Instead, there are simply uneven volumes distributed across multiple thru lanes, or multiple turn lanes. In this case, the following four-step procedure can be used to code the data:

1. On the Edit > Lanes screen, select the appropriate lane number, and then click on the "Lane-by-Lane" radio button.
2. On the Edit > Traffic screen, code lane-specific volumes and saturation flow rates for each associated link. It will be necessary to click on the "Next Links" button to access the data for "extra" link numbers 13 and higher.
3. On the Edit > Feeders screen, code feeding flows that reflect the extra link numbers. Automatic flow balancing cannot be used to compute the feeding flows from extra links, so these must be coded by the user. If more than 4 feeding links are associated with a downstream link, code only the links with the highest feeding volumes.
4. On the Edit > Optional > Timing screen, code the extra link numbers into the phasing sequence. This assumes that the standard link numbers (1 through 12) have already been coded into the phasing sequence automatically by using the graphical Edit > Timing screen.

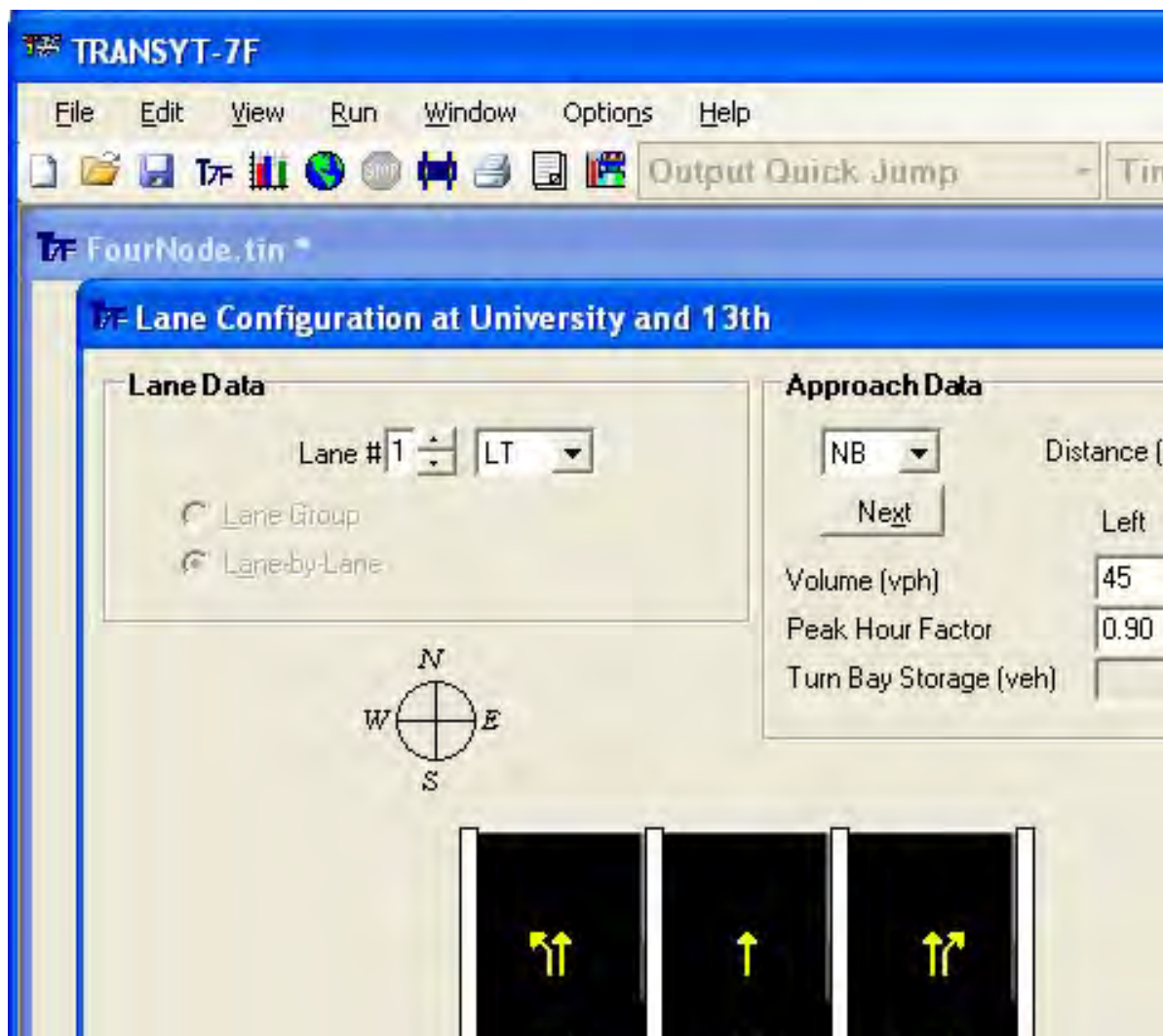
The more complex case of lane-by-lane analysis is a situation in which shared lanes are indeed involved. In this case, lane assignments must be coded in a specific sequence on the Lane Configuration screen. Once this has been accomplished, steps 2 through 4 above can be followed as usual.

### Sample Problem

The "base case" or "base scenario" network is the one generated from scratch in the first exercise of the tutorial chapter, i.e., the four-node "box" shaped network. Alternatively, this file ("FourNode.tin" or "FourNode Lefty.tin") can also be obtained from the TRANSYT-7F Example folder. Note that the results from FourNode.tin throughout this exercise are sometimes slightly different than the ones from FourNode Lefty.tin, due to round-off errors caused by metric unit conversion.

Suppose that the northbound lane configuration at node #1 includes a shared thru+left lane, an exclusive thru lane, and a shared thru+right lane. (For left-hand drive, this data should be coded at node #2.) The shared thru+left lane will serve permitted left-turns, which introduces one element to benefit from a lane-by-lane analysis. Instead of distributing the original 400 vph evenly across all three lanes, suppose that the right-most lane has a volume of 200 vph because of a downstream shopping center. This introduces a second element to benefit from a lane-by-lane analysis.

The first input coding step involves the Edit > Lanes screen. Start by creating lane #3, either by right-clicking on lane #2 and selecting "Add trailing", or by using the Add/Delete pushbutton. Then code a lane use of "T" (thru only) for all three lanes, starting at lane #1. Next, switch back again to lane #1, and click on the "Lane-by-Lane" radio button. Next, switch to lane #3, and click on the "Lane-by-Lane" radio button. This establishes extra links 113 and 114 for modeling this complex approach. Code a lane use of "TR" for lane #3, and a lane use of "LT" for lane #1. This re-establishes the shared lanes. Finally, restore the original volumes and peak hour factors for each turning movement. This completes step #1 of the four-step lane-by-lane coding process mentioned earlier. The screen image below shows the completed Lane Configuration screen for this example.



Step #2 in the coding process involves the Edit > Traffic screen. The adjusted saturation flow rates (1900 vphg) for each northbound thru link, as well as the adjusted northbound left-turn and right-turn volumes (50 vph), do not require modification. However, the volumes for each thru link (101, 113, and 114) must be corrected at this stage. For link 101, code a volume of 100 vph. Then click on the "Next Links" button. Code a volume of 200 vph for link 114, and a volume of 100 vph for link 113.

Step #3 in the coding process involves the Edit > Feeders screen. Only the thru movement feeding flows, from node #1 into node #3, require modification. Therefore, switch to node #3. For link 301, change feeding link #1 from 101 to 113, with a feeding flow of 100 vph. Then, click on the "Next Data" button. For link 301, code feeding link #4 as 114, with a feeding flow of 200 vph. The cruise speed for each feeding link should be 35 mph (56 kmph).

Step #4 in the coding process involves establishing the phasing sequence. Proceed to the Edit > Timing screen, and switch back to node #1. Since one of the northbound turning movements is both permitted and shared, delete phase #4, and code all of the northbound and southbound movements in phase #3. Increase the green time for phase #3 to 50 seconds, to restore the original cycle length of 100 seconds. At this time it is recommended to confirm that the "extra" link numbers (113 and 114) have been successfully included within phase #3. This can be confirmed on the Edit > Optional > Timing screen. The Timing View screen contains logic to automatically include extra links within the phasing

sequence, if they were originally coded within the Lane Configuration screen.

This completes the entire four-step coding process. The screen image below illustrates the new results in the legacy report after running the model.

Movement/ Node No.	Flow vph	Sat Flow vphg	Degree of Sat %	Total Travel veh-mi	Travel Time Total veh-h	Time Avg sec/v	----- Del Uniform veh-h	Random veh-h
NB THRU	100	1900	11	24.98	1.09	39.1	0.37	0.00
SB THRU	450	1900P	48	112.41	6.63	53.1	3.37	0.03
LEFT	50	0	8	12.49	0.66	47.4	0.30	0.00
EB THRU	450	1900P	96*	112.41	12.06	96.5	4.63	4.20
LEFT	50	1805	28	12.49	0.95	68.4	0.58	0.01
WB THRU	449	1900P	96*	112.41	13.33	106.8	7.65	2.44
LEFT	50	1805	28	12.49	1.01	72.4	0.65	0.00
1 113	150	1900P	35	37.47	1.63	39.2	0.54	0.02

Link 113, which is the left-most thru lane, has the highest degree of saturation (35%) among all three thru lanes, because it is shared with the permitted left-turners. Link 114, which is the right-most thru lane, has the next highest degree of saturation (27%), because it has the highest volume. Link 101, which is the middle thru lane listed as "NB THRU", has the lowest degree of saturation (11%), because it has the lowest volume. The node summary report, which can be loaded on the Output Quick Jump by selecting "Node Output Summary", contains the same overall results in a convenient one-page summary. However, the node summary reports are not capable of providing the individual movement results for extra links such as 113 and 114.

## CORSIM Processing and Optimization Examples

This section of the documentation provides examples of CORSIM processing and optimization. The "base case" or "base scenario" network is the sample file called "actctrl.trf", which is available within the TRANSYT-7F Example folder. This is a single arterial street with several actuated controlled intersections. One of the intersections is governed by fully-actuated control, three intersections have semi-actuated control, and one of the intersections is fully pre-timed. This is considered to be a

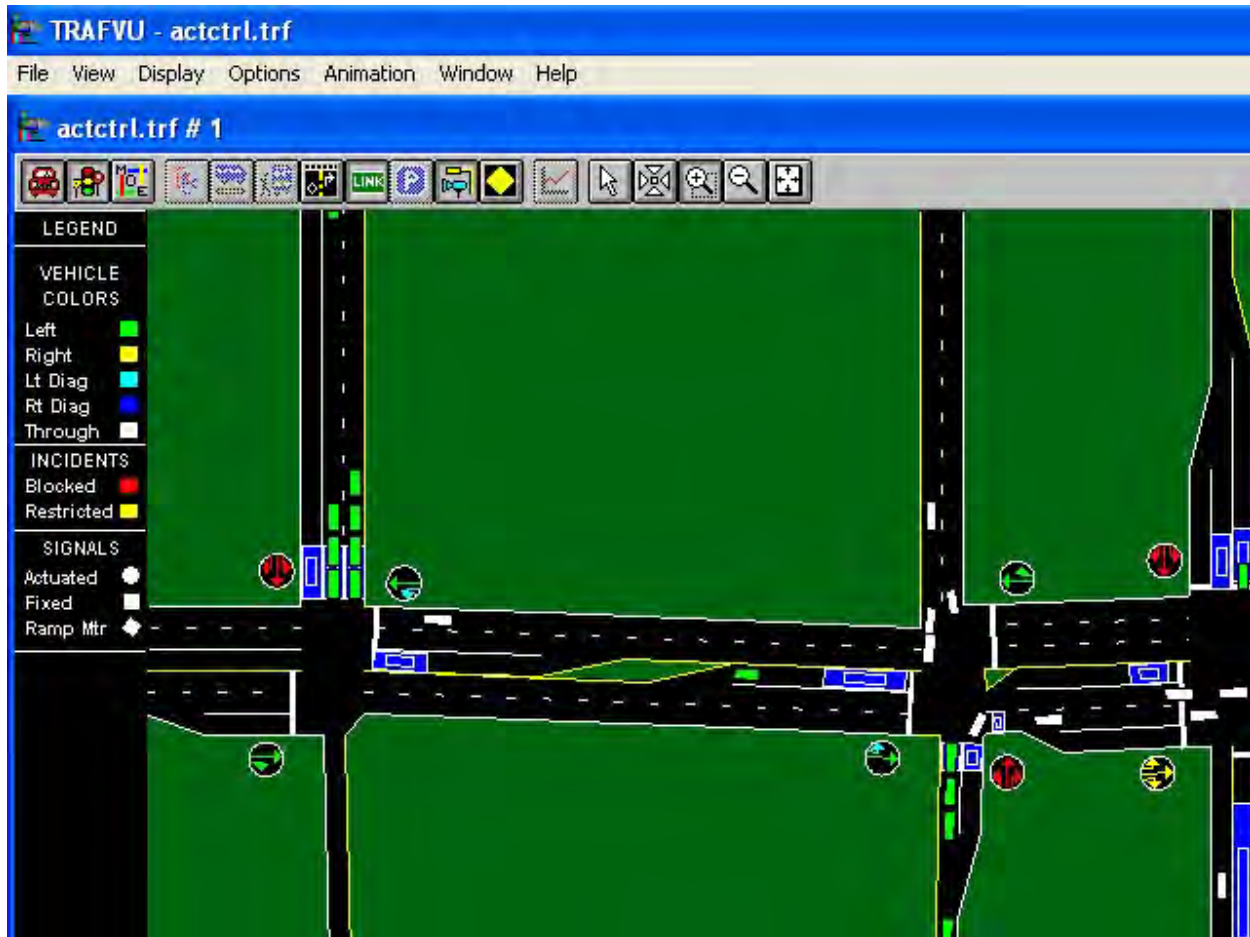
"NETSIM" file, because there are no freeway links or "FRESIM" components. CORSIM is fully capable of processing NETSIM-only or FRESIM-only input data files, in addition to any integrated CORSIM networks. The file "actctrl.trf" was taken from the TSIS 6 installation, and the results from this section were generated by CORSIM 6.

The initial set of program runs will be used to demonstrate CORSIM processing without optimization. The next set of runs will demonstrate "direct CORSIM optimization", in which TRANSYT-7F manages the genetic algorithm, and CORSIM evaluates the candidate timing plans through its own simulation. The next set of runs will demonstrate "indirect CORSIM optimization", in which TRANSYT-7F performs macroscopic optimization, and the optimized timings are then exported back to the TRF file. The final run will demonstrate "grouped node" optimization, using CORSIM as the underlying simulation engine.

### CORSIM File Import

To begin the exercise, select Options > Preferences from the TRANSYT-7F GUI. Turn on the checkboxes that allow importing of two-way stop-controlled intersections from CORSIM, and importing of fully-actuated intersections from CORSIM. Click on File > Open, and select Files of type > CORSIM Input Files (\*.trf). Browse to locate actctrl.trf, and click on Open to start the import process. If the import process is successful, TRANSYT-7F will issue a message that says "actctrl.tin now contains the processing and optimization settings associated with actctrl.trf". If actctrl.tin already exists in the associated subdirectory, it will not be re-created, but will simply be re-loaded. CORSIM processing can then be performed by selecting Run > TRANSYT-7F (or clicking on the blue T7F icon), and CORSIM animation can be viewed by selecting File > CORSIM > View animation for this network (or clicking on the CORSIM Animation icon).





### Network Geometry

The TRAFVU screen image above illustrates the connection between three coordinated, semi-actuated controllers. Reading from left to right, these CORSIM node numbers are 151, 152, and 255. The fully-actuated intersection (node #450) is off the screen to the left. The fully pre-timed intersection (node #190) is off the screen to the right. For this example it will be assumed that the northbound direction runs straight up the screen, and that the major street runs east and west (left to right and vice-versa). Notice that the actuated control detectors are present on all lanes and approaches, except for the major street through movements. The northern approach at the middle node (#152) is an "exit-only" approach, on which all vehicles are leaving the intersection.

Besides the five signal controlled intersections, there is also a stop-sign controlled intersection (node #350) off the screen to the left. All other node numbers are "dummy" nodes used to introduce vehicles on to the network.

### CORSIM Processing Without Optimization

If this file were processed from within TSIS 6, the following output statistics would be listed at the bottom of CORSIM's legacy report:

**TOTAL VEHICLE- MILE = 278.44    VEHICLE-HOURS OF: MOVE TIME = 9.21 , DELAY TIME = 6.42 ,  
TOTAL TIME = 15.63**

**AVERAGE SPEED ( MPH)= 17.82 MOVE/TOTAL = 0.59 MINUTES/MILE OF: DELAY TIME = 1.38 ,  
TOTAL TIME = 3.37**

Proceed to the Edit > Analysis screen. This screen shows that a simulation run (without optimization) will be performed by default. The four settings applicable to CORSIM processing are Simulation Period (5 minutes), Random # Seed Variations (1), Initialization Period (900 seconds), and Volume Multiplier (100%). The Simulation Period and Initialization Period values were taken directly from the TRF file, whereas the other values are simply defaults.

Close the Analysis screen, and then run the program by selecting Run > TRANSYT-7F, or by clicking on the blue "T7F" icon. The initial results should show an average network-wide control delay of 8.3 seconds per vehicle. Also notice that the network-wide Delay Time and Speed are identical to those obtained directly from TSIS. This time, change the Simulation Period to 15 minutes, change the Volume Multiplier to 150%, and run the model one more time. The new results, which are shown below, should show an average network-wide control delay of 20.8 seconds per vehicle. Higher delays such as this should be better for demonstrating the optimization capabilities of TRANSYT-7F.

Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst: Date Performed:  
Agency: Analysis Time Period:  
Jurisdiction:  
Run Title: Run Title  
File Name: Actctrl.tin

CORSIM Version 6  
Random # Seed Variations: 1  
Volume Multiplier: 1.5  
Simulation Period (sec): 900  
Initialization Period (sec): 900

Processing Results:

```
<-----NETSIM Approach Links-----> Network-Wide
Control Total Stop Queue Vehicle Percent Delay
Delay Delay Delay Delay Trips Stops Speed Time Speed
------(sec/veh)----- (%) (mph) (veh-hr) (mph)

20.8 25.3 17.9 19.4 5567 50.5 11.4 40.5 11.7
```

Links

```
14.7 16.0 13.4 13.7 189 40.0 10.1 ( 375, 450)
18.8 19.7 17.0 17.5 79 83.0 10.2 ( 246, 450)
38.6 47.8 28.5 31.9 644 82.0 7.0 ( 375, 350)
3.9 5.3 3.8 3.8 198 6.0 11.0 ( 151, 350)
11.1 14.8 10.7 11.2 614 32.0 4.7 ( 350, 151)
121.7 142.7 117.9 129.1 254 92.0 2.0 ( 150, 151)
11.2 14.1 10.1 10.4 213 39.0 11.4 ( 152, 151)
17.9 24.4 14.6 16.7 713 63.0 7.2 ( 151, 152)
8.8 11.9 8.2 8.6 237 46.0 7.4 ( 255, 152)
29.0 31.0 23.7 25.3 192 83.0 8.1 ( 153, 152)
0.9 2.6 0.7 0.7 594 9.0 17.1 ( 152, 255)
8.3 8.6 7.4 7.5 83 84.0 16.3 ( 256, 255)
7.6 10.9 6.5 6.8 143 35.0 22.1 ( 190, 255)
27.5 28.3 25.7 26.1 59 83.0 8.0 ( 257, 255)
23.1 27.4 17.1 18.7 440 65.0 13.6 ( 255, 190)
18.5 19.6 17.7 17.9 37 59.0 13.7 ( 2, 190)
10.4 11.0 8.6 9.0 196 54.0 14.2 ( 1, 190)
```



4.8	5.6	3.2	3.5	612	33.0	23.8	( 100, 450)
22.6	23.1	20.9	21.4	61	83.0	9.2	( 245, 450)
486.9	493.0	482.7	486.2	9	100.0	1.1	( 351, 350)

#### Average Queue by Lane

Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6	Lane 7	Links
--------	--------	--------	--------	--------	--------	--------	-------

0.0	0.0	0.0	0.0	0.0	0.0	2.0	( 375, 450)
0.0	0.0	0.0	0.0	0.0	0.0	0.0	( 246, 450)
12.0	12.0	0.0	0.0	0.0	0.0	0.0	( 375, 350)
0.0	0.0	0.0	0.0	0.0	0.0	1.0	( 151, 350)
3.0	3.0	0.0	0.0	0.0	0.0	0.0	( 350, 151)
17.0	19.0	0.0	0.0	0.0	0.0	0.0	( 150, 151)
0.0	1.0	0.0	0.0	0.0	0.0	1.0	( 152, 151)
5.0	5.0	0.0	0.0	0.0	0.0	2.0	( 151, 152)
0.0	1.0	1.0	0.0	0.0	0.0	0.0	( 255, 152)
4.0	0.0	0.0	0.0	0.0	0.0	1.0	( 153, 152)
0.0	0.0	0.0	0.0	0.0	0.0	0.0	( 152, 255)
0.0	0.0	0.0	0.0	0.0	0.0	0.0	( 256, 255)
1.0	0.0	0.0	0.0	0.0	0.0	0.0	( 190, 255)
0.0	0.0	0.0	0.0	0.0	0.0	1.0	( 257, 255)
5.0	4.0	0.0	0.0	0.0	0.0	0.0	( 255, 190)
0.0	0.0	0.0	0.0	0.0	0.0	0.0	( 2, 190)
0.0	1.0	0.0	0.0	0.0	0.0	0.0	( 1, 190)
1.0	1.0	0.0	0.0	0.0	0.0	0.0	( 100, 450)
0.0	0.0	0.0	0.0	0.0	0.0	1.0	( 245, 450)
11.0	0.0	0.0	0.0	0.0	0.0	3.0	( 351, 350)

#### Maximum Queue by Lane

Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6	Lane 7	Links
--------	--------	--------	--------	--------	--------	--------	-------

2.0	2.0	0.0	0.0	0.0	3.0	5.0	( 375, 450)
2.0	0.0	0.0	0.0	0.0	3.0	2.0	( 246, 450)
30.0	28.0	0.0	0.0	0.0	0.0	0.0	( 375, 350)
0.0	0.0	0.0	0.0	0.0	0.0	4.0	( 151, 350)
7.0	7.0	0.0	0.0	0.0	0.0	2.0	( 350, 151)
26.0	25.0	0.0	0.0	0.0	0.0	2.0	( 150, 151)
4.0	4.0	0.0	0.0	0.0	0.0	5.0	( 152, 151)
19.0	19.0	0.0	0.0	0.0	0.0	6.0	( 151, 152)
3.0	4.0	6.0	0.0	0.0	0.0	0.0	( 255, 152)
12.0	0.0	0.0	0.0	0.0	0.0	3.0	( 153, 152)
4.0	2.0	0.0	0.0	0.0	2.0	2.0	( 152, 255)
1.0	0.0	0.0	0.0	0.0	1.0	3.0	( 256, 255)
5.0	2.0	0.0	0.0	0.0	0.0	1.0	( 190, 255)
3.0	0.0	0.0	0.0	0.0	1.0	4.0	( 257, 255)
16.0	17.0	0.0	0.0	0.0	0.0	0.0	( 255, 190)
2.0	1.0	0.0	0.0	0.0	0.0	0.0	( 2, 190)
2.0	6.0	0.0	0.0	0.0	0.0	0.0	( 1, 190)
7.0	7.0	0.0	0.0	0.0	2.0	2.0	( 100, 450)
2.0	0.0	0.0	0.0	0.0	1.0	3.0	( 245, 450)
23.0	0.0	0.0	0.0	0.0	0.0	3.0	( 351, 350)

Average Occupancy	Percent Storage	Phase Failures	Lane Changes	Links
-------------------	-----------------	----------------	--------------	-------

4.2	9.9	8.0	37.0	( 375, 450)
2.7	9.7	0.0	0.0	( 246, 450)
34.3	52.8	0.0	167	( 375, 350)
1.5	8.7	0.0	0.0	( 151, 350)
9.3	63.0	0.0	44.0	( 350, 151)
37.8	59.6	12.0	48.0	( 150, 151)
4.2	10.6	0.0	49.0	( 152, 151)

20.3	51.2	0.0	256	( 151, 152)
3.5	13.6	0.0	178	( 255, 152)
7.3	26.1	0.0	0.0	( 153, 152)
3.4	14.9	0.0	50.0	( 152, 255)
2.1	6.6	0.0	0.0	( 256, 255)
4.1	3.1	0.0	35.0	( 190, 255)
2.5	7.7	0.0	0.0	( 257, 255)
19.5	19.5	0.0	93.0	( 255, 190)
1.8	2.4	0.0	5.0	( 2, 190)
4.0	9.2	0.0	10.0	( 1, 190)
14.7	14.1	0.0	91.0	( 100, 450)
2.4	8.3	0.0	0.0	( 245, 450)
14.5	55.6	0.0	0.0	( 351, 350)

#### Signalized Intersection Level of Service

Node	Control Delay (sec / veh)	LOS	CORSIM_ID
------	------------------------------	-----	-----------

1	19.1	B	Intersection #190
2	35.3	D	Intersection #350
3	37.1	D	Intersection #151
4	17.9	B	Intersection #152
5	4.5	A	Intersection #255
6	9.1	A	Intersection #450

At this stage, it would be possible to view these CORSIM results graphically in the Formatted Report, and in the Map View. By coding a progression route inside the Map View, it will also be possible to obtain a Time-Space Diagram. Therefore, open the Map View, and select nodes 3-4-5 for defining as a progression route. Nodes #2 and #6 should not be included in the progression route because they represent a two-way stop and a fully-actuated signal, respectively. Two-way stops should only be included when they are in the middle of a route, and fully-actuated signals are incapable of achieving reliable progression. Node #1 should not be included because it has a different cycle length than nodes 3-4-5. The Map View image below illustrates intersection delay and level of service, free-flow speeds, and the progression route along nodes 3-4-5.



The following steps are then used to create a time-space diagram:

- (Edit > Analysis) Specify a background cycle length of 90 seconds
- (View > Time-Space Diagram) Switch the diagram from "Fwd" to "Both"
- (View > Time-Space Diagram) Specify a speed of 30 mph

The time-space diagram should now show a bandwidth efficiency of 11% ("poor progression") and a bandwidth attainability of 33% ("major changes needed"). Later on, a second time-space diagram can be plotted after optimization. This will determine whether the optimized solution contains better progression opportunities, relative to these original results.

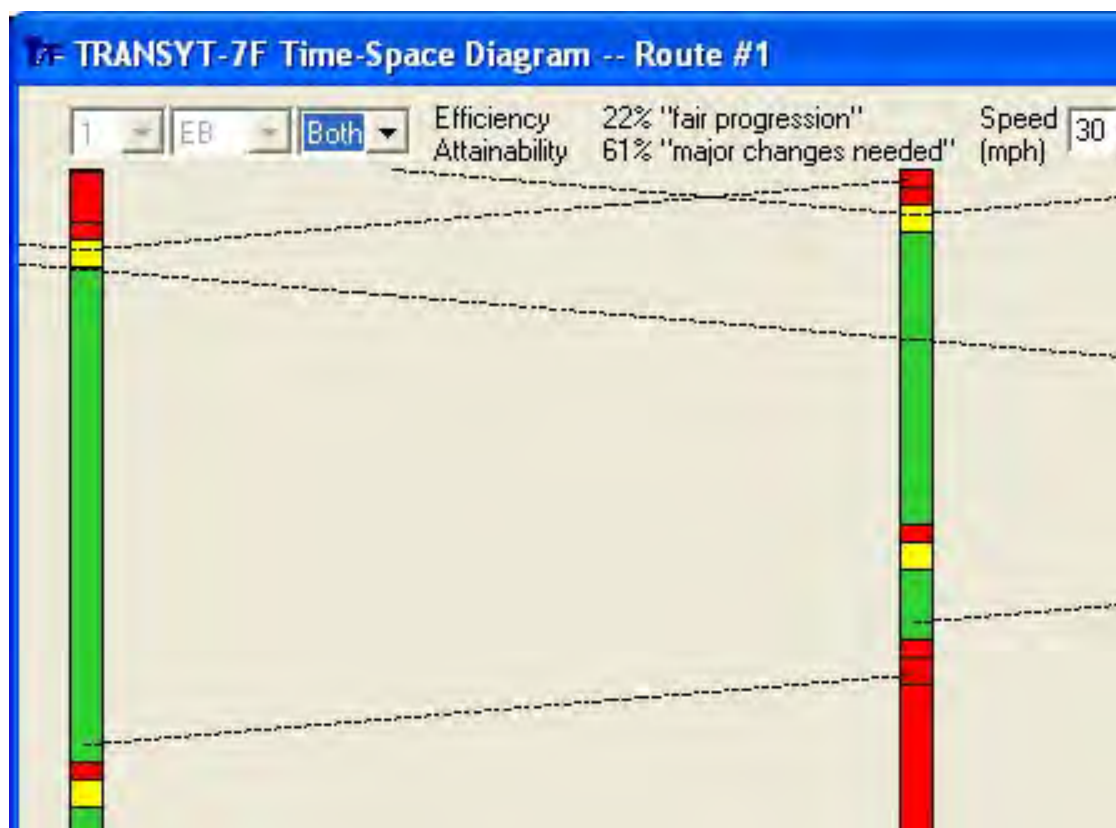
#### Direct CORSIM Optimization

Proceed to the Edit > Analysis screen, and begin by selecting the Optimization radio button. Next, code '10' for the Number of Generations. Then, click on the "Genetic Algorithm Settings" pushbutton, and code a Mutation Probability of 4%. Higher mutation probabilities tend to be more effective for simple optimization runs with only a few signals and signal settings being optimized. Finally, go to the Edit > Optional > Global screen. Click on the Optimization Node List tab, and remove nodes #5 and #6 from the list by replacing them with zeroes. The Map View image shown earlier indicates that nodes #5 and #6 are already operating at LOS A, and optimization tends to be more efficient when such nodes are removed from the list. Then click on the Network Parameters tab, and code '1' as the master controller node number. This will prevent the offset from being optimized at node #1. After this, begin optimization by selecting Run > TRANSYT-7F, or by clicking on the blue "T7F" icon.

Following optimization, the summary report lists several items of interest. First, the average control delay in NETSIM was successfully reduced from 20.8 sec/veh to 13.6 sec/veh. Secondly, the bottom of the summary report says "The final solution originated in generation #10", and, "To obtain detailed output, process the punch file: Genetic.trf". Because only 10 generations were executed, it is possible that more generations of optimization would yield a better timing plan. In addition, the punch file Genetic.trf could optionally be used for subsequent TRANSYT-7F or TSIS processing, because this TRF file contains the optimized timings.

Another item of interest is that the cycle lengths for the three semi-actuated nodes have changed, even though cycle length optimization was not requested. This occurred because the individual nodes were marked as "Coordinated Signals" on the Edit > Timing screen, and coordinated signals must adopt the background cycle length during the optimization process. To allow one of these intersections to have a unique cycle length after optimization, the user could either uncheck the Coordinated Signal checkbox for that node on the Edit > Timing screen, or delete the associated node number from the Edit > Optional > Global > Optimization Node List screen. Upon opening any of the node-specific Edit screens such as Lanes or Timing, it can be seen that CORSIM node #350 is associated with TRANSYT node #1, CORSIM node #190 is associated with TRANSYT node #2, etc. Also on the optimization summary report, it can be seen that there is a different cycle length at node #450, which is the isolated fully-actuated node.

In order to take a closer look at the optimized solution, open the punch file "Genetic.trf". Select Run > TRANSYT-7F to generate the detailed output results. By opening the Map View and activating the intersection-wide delay and LOS, it is possible to see that there have been significant delay reductions at node #1 and node #2, relative to the original results. Next, select nodes 3-4-5, and re-define the same progression route that there was in the original file. Then, select View > Time-Space Diagram. By plotting a 30-mph bandwidth as shown in the screen image below, it is possible to see that efficiency and attainability are now 22% and 61%, respectively. Therefore, the optimized solution has approximately double the progression opportunities, relative to the original results.



## Companion Macroscopic Analysis

It may be helpful to analyze a particular NETSIM network macroscopically, using the TRANSYT engine. Although TRANSYT is capable of analyzing two-way stop-control (TWSC) and fully-actuated control, these types of control cannot be automatically converted between macroscopic and microscopic analysis. Therefore, select Options > Preferences from the TRANSYT-7F GUI, and turn off the checkboxes that allow importing of TWSC and fully-actuated intersections from CORSIM. Next, select File > Open, to import actctrl.trf once again. The actctrl.tin file from the previous exercise might need to be deleted first, so it doesn't automatically get reloaded.

After importing actctrl.trf, select Run > TRANSYT-7F, in order to post-process the CORSIM results. The baseline node-specific results from NETSIM are listed below. Note that network-wide control delay for all NETSIM links is listed as 10.3 seconds per vehicle. In the upcoming experiment for Timing Export to CORSIM, this original network-wide control delay will be compared to an optimized network-wide control delay, using the timings from TRANSYT optimization.

Node	Control Delay	LOS	
1	19.0	B	Intersection #190
2	13.5	B	Intersection #151
3	6.8	A	Intersection #152
4	4.4	A	Intersection #255

Next, select File > CORSIM > Convert > to macroscopic analysis. After converting the file to macroscopic analysis, open the Edit > Analysis screen. It shows that multi-cycle step-wise simulation is activated, for consistency with CORSIM. Without editing any input data, proceed to run the TRANSYT model. The node-specific results, which can be viewed easily by clicking the "LOS" icon on the Map View, are listed below.

Node	Control Delay	LOS	
1	28	C	Intersection #190
2	31	C	Intersection #151
3	20	B	Intersection #152
4	114	F	Intersection #255

So why is there such a significant discrepancy in results? The first reason is that TRANSYT is simulating the force-offs from NETSIM, which are shown on the Edit > Timing screen, as average green times. To convert force-offs to average green times, it is best to use the actuated estimation procedure. Reload the Edit > Analysis screen, select "Estimation", and code a lower cycle length of 90 seconds. Then load the Edit > Timing screen, and turn off the Coordinated Signal checkbox for node #1. Click on File > Save in order to save actctrl.tin. It should now be possible to run the TRANSYT actuated estimation model and obtain results. To obtain the node-specific results, load the punch file "actctrl.pun" following completion of the actuated estimation run, and then perform a simulation-only run. The new node-specific results are listed below.

Node	Control Delay	LOS	
1	17	B	Intersection #190
2	21	C	Intersection #151
3	12	A	Intersection #152
4	7	A	Intersection #255

Since these new results still disagree slightly with those from CORSIM, more input data review is required. The next input discrepancy of note is that the turning movement volumes were imported directly from TRF record type 21, but some of these volumes were coded as fractions. For example, the southbound turning movement volumes at node #4 were coded as 20 left, 6 thru, 78 right. However, at upstream entry node 8025, the entry volume was coded as 169 vph. Therefore, for consistency with

CORSIM the volume in TRANSYT should be 33 left, 10 thru, 126 right.

A complete list of volume revisions for this example is shown below. Before coding these volumes, be sure to re-load actctrl.tin, since actctrl.pun was just used to generate the previous results. These volume revisions should be coded on the Edit > Lanes screen, or on the Edit > Traffic screen.

- WB at node #1: change 0, 80, 20 to 0, 60, 15
- SB at node #1: change 40, 0, 60 to 160, 0, 240
- NB at node #3: change 239, 0, 226 to 195, 0, 184
- SB at node #4: change 20, 10, 78 to 33, 10, 126

In a real-world project, it might be preferable to re-code these relative turn volumes in CORSIM, and then re-import the TRF file into TRANSYT-7F. However, if all turning movement volumes in a given TRF file are coded as actual volumes (instead of fractions), this volume revision step will not be necessary.

After correcting the volumes, automatic flow balancing should be re-applied on either the Edit > Feeders screen or the Map View. The easiest way is to delete and then re-draw two-way link connections between each node on the Map View, so do this now.

The next input discrepancy of note is that gap settings for actuated controllers are not imported from the TRF file. Instead, a default gap setting of 3.0 seconds is used. Since most of the gap settings in actctrl.trf are 2.0 seconds, proceed to the Edit > Optional > Actuated screen, and change all gap settings to 2.0 seconds at each node.

Another reason that CORSIM delays are sometimes lower than TRANSYT delays is the start-up lost time. The CORSIM Processing Methodology section from Chapter 4 explains that for external links, a start-up lost time of 2.0 seconds in CORSIM is often consistent with a start-up lost time of zero seconds in TRANSYT. To save time, the global start-up lost time will be used in this example. To code a start-up lost time of zero seconds in TRANSYT, go to the Edit > Optional > Global > Run Controls screen, and code a "Network-Wide Startup Lost Time" of '0'. After re-running the actuated estimation model, the new node-specific results from actctrl.pun are listed below.

Node	Control	Delay	LOS
1	17	B	Intersection #190
2	18	B	Intersection #151
3	8	A	Intersection #152
4	5	A	Intersection #255

These results now match the original results from CORSIM fairly closely. The discrepancy at node #2 might be explained by the fact that part of the eastbound queue length stretches beyond a stop-controlled intersection. Because of this, NETSIM is only able to report part of the eastbound control delay. This is one example of extra information that can be gleaned from a companion macroscopic analysis. The inconsistency between CORSIM entry volumes and turning movement counts was another discovery. There are still a couple of flow balancing issues remaining at the top of the TRANSYT legacy report, and the macroscopic results could probably be calibrated further, but for now these results will be considered acceptably consistent with the microscopic simulation results.

#### Timing Export to CORSIM

At this stage it is possible to optimize timings using TRANSYT and then export the results to CORSIM. Normally, this type of "indirect" CORSIM optimization would not be expected to produce a better timing plan than the direct CORSIM optimization described earlier, unless phasing sequence optimization were involved. Optimized phasing sequences cannot be exported to a pre-existing CORSIM file, although they could be exported to a brand-new CORSIM file via "one-touch" CORSIM animation. In this exercise, a simple optimization of cycle length, splits, and offsets is attempted, so that the timings can

be exported back to the original CORSIM input file.

Macroscopic (TRANSYT-based) optimization is based on the average green times at actuated controllers, instead of the force-offs, so the punch file (actctrl.pun) from the previous actuated estimation run should be used as the starting point for optimization. Proceed to File > Save As, and save actctrl.pun as actctrl.tin, which will overwrite the previous TIN file.

Next, open the Edit > Analysis screen. Select single-cycle step-wise optimization of cycle length, splits, and offsets, using the genetic algorithm. Code a lower cycle length of 60, an original cycle length of 90, an upper cycle length of 100, and a cycle length increment of 10. Select "Control Delay Only" under the DI combo box, to obtain an objective function that is consistent with NETSIM. Code '100' as the Number of Generations. Turn on "Initial Timing Flag 1".

Then, open the Edit > Timing screen. For node #1, decrease the phase #2 green time from 41 seconds to 36 seconds, and turn on the Coordinated Signal checkbox.

Finally, start the optimization run via Run > TRANSYT-7F. The optimization summary report shows that average control delay for the system has decreased from 11.0 sec/veh to 8.0 sec/veh, with a recommended cycle length of 60 seconds.

The next step is to once again open the punch file actctrl.pun, and select File > Save As to change the filename back to actctrl.tin. Next, select File > CORSIM > Convert to microscopic analysis. Then, select File > CORSIM > Export TRANSYT timing to CORSIM for this network, to export cycle length, splits, and offsets. It is allowable to export only a subset of signal settings by unchecking boxes on the Export Settings dialog, or to export timing only at certain nodes by using the Export button on the Edit > Timing screen, but for this example all timings will be exported at all nodes.

Once the timings have been exported to the TRF file, a copy of the original TRF file (called Copy of actctrl.trf) having the original timings is automatically created on the computer. At this time, if a CORSIM processing run is performed without optimization, the report shows that system-wide average control delay has decreased from 10.6 (refer to original results from the Companion Macroscopic Analysis exercise) to 9.3 seconds per vehicle, using the signal settings recommended by TRANSYT. These microscopic simulation results approximate the macroscopic simulation results that said delay would drop from 11.0 to 8.0 seconds per vehicle. Again, microscopic (direct CORSIM) optimization could probably reach a better solution, but performing a companion macroscopic analysis can potentially offer additional insights, as discussed earlier.

### Grouped Node Optimization

It may be helpful to define closely spaced intersections as "grouped nodes". This allows the closely spaced intersections to retain the same number of phases during optimization, the same phasing sequence, the same green times, and the same difference of offset. Although grouped nodes share the same green times, these green times can still be optimized, and the offsets can be optimized as well. Refer to the Grouped Nodes section of the documentation for more details.

If the experiment for Timing Export to CORSIM was recently completed, the latest actctrl.trf file now has optimized timings from TRANSYT, and the original file is now named "copy of actctrl.trf". For this experiment, the original actctrl.trf file should be used. Therefore, rename or delete actctrl.trf, and then rename "copy of actctrl.trf" back to actctrl.trf.

To set the stage for this experiment, select Options > Preferences from the TRANSYT-7F GUI. Turn on the checkboxes that allow importing of TWSC and fully-actuated intersections from CORSIM, and click on the "OK" button. Next, select File > Open, to import actctrl.trf once again. The actctrl.tin file from the previous exercise might need to be deleted first, so it doesn't automatically get reloaded.

After importing actctrl.trf, open the Edit > Timing screen, and scroll through the timing plan at all

six intersections. It is clear that intersections 3, 4, and 5 all have similar three-phase timing plans. Note that TRANSYT-7F node numbers 3, 4, and 5 correspond with CORSIM node numbers 151, 152, and 255. Looking at either the Map View or one-touch CORSIM animation, it is clear that intersections 3, 4, and 5 are all within 500 feet of each other. Closely spaced intersections with similar timing plans are good candidates for grouping, as described earlier.

To define these three nodes as grouped nodes, open the Edit > Optional > Global > Optimization Node List screen. Code node numbers 3 and 4 with negative signs, so that the overall list reads "1, 2, -3, -4, 5, 6". This means that grouped master node #5 will be optimized normally, and grouped slave nodes #3 and #4 will have their green times automatically matched to node #5.

Next, open the Edit > Analysis screen. Select optimization of cycle length, splits, and offsets. Code a lower cycle length of 60, an original cycle length of 90, an upper cycle length of 100, and a cycle length increment of 10. Code '10' as the Number of Generations. Finally, go to the Options > Genetic Algorithm dialog, and code '70' as the upper limit on maximum phase times for fully-actuated controllers in CORSIM. After clicking "OK" inside the Genetic Algorithm dialog, begin optimization by selecting Run > TRANSYT-7F, or by clicking on the blue "T7F" icon.

The output listing below illustrates the results of the optimization run. Notice that the optimized offsets at node numbers 151, 152, and 255 have all changed by exactly the same amount (44 seconds), relative to the original offsets. In addition, the optimized green splits are equal among all three grouped nodes.

#### Traffic Network Study Tool (TRANSYT-7F, United States Version) Release 11.3

Analyst:                      Date Performed:  
Agency:                      Analysis Time Period:  
Jurisdiction:  
Run Title:    Run Title  
File Name:    actctrl.tin

Simulation Engine: CORSIM 6

Random # Seed Variations:            1  
Volume Multiplier:                    1  
Simulation Period (sec):               300  
Initialization Period (sec):            900

#### Genetic Algorithm Input Parameters

-----  
Crossover Probability (%):            30  
Mutation Probability (%):            1  
Convergence Threshold (%):            0.01  
Maximum Number of Allowable Generations: 10  
Actual Number of Optimization Generations: 10  
Original Cycle Length (sec):            90  
Population Size:                        10  
Random Number Seed:                   7781  
Elitist Method:                        True  
Upper Limit on Maximum Phase Times (sec): 70  
Objective Function:                    Control Delay (NETSIM only)

#### Optimization Results (Cycle Length, Splits, and Offsets)

-----

Node	Cycle	Offset	NS Phasing	EW Phasing
	Init	Final	Init	Final
190	95	60	0	38



```

350 --- --- --- --- --- --- --- ---
151 90 60 18 2 --- --- --- ---
152 90 60 85 39 --- --- --- ---
255 90 60 0 44 --- --- --- ---
450 70 111 --- --- --- ---

```

Node	Split #1	Split #2	Split #3	Split #4	Split #5	Split #6	Split #7
	Init	Final	Init	Final	Init	Final	Init
190	36	21	45	24	14	15	
350	--	--					
151	40	21	35	22	15	17	
152	35	21	40	22	15	17	
255	30	21	45	22	15	17	
450	50	60	20	51			

	<-----NETSIM Approach Links----->								Network-Wide	
	Control	Total	Stop	Queue	Vehicle	Percent			Delay	
	Delay	Delay	Delay	Delay	Trips	Stops	Speed	Time	Speed	
	----- (sec/veh) -----				(%)	(mph)	(veh-hr)	(mph)		
Initial	8.3	10.4	7.1	7.4	2048	36.6	15.7	6.4	17.8	
Final	6.7	8.8	5.6	5.9	2033	39.5	16.1	5.4	19.0	

The final solution originated in generation #10  
 To obtain detailed output, process the punch file:  
 Genetic.trf



# Appendix A: TRR Paper

## Appendix A: TRR Paper

The publisher of this documentation credits the Transportation Research Record: Journal of the Transportation Research Board, No. 1811. This paper is reprinted with permission of the Transportation Research Board, Washington, D.C. This work originally appeared in *Transportation Research Record* 1811, Transportation Research Board, National Research Council, Washington, D.C., 2002, pp. 84-91. This material does not imply endorsement of any particular product, method, or practice by the Transportation Research Board or its parent organization, the National Academy of Sciences.

## PREDICTION OF TRAFFIC-ACTUATED PHASE TIMES ON ARTERIAL STREETS

by

David K. Hale, Ph.D.  
Assistant in Engineering  
University of Florida

and

Kenneth G. Courage, P.Eng.  
Professor  
University of Florida

5747 words + 7 (Tables and Figures) · 250 = 7497 words

## ABSTRACT

Traffic congestion in U.S. cities has grown rapidly in recent years, and numerous solutions are needed to address the problem. This paper describes new research for producing basic improvements to the practice of traffic signal timing design, which is one of the many available weapons for fighting congestion and delay. Existing models in the literature for prediction of traffic-actuated phase times are oversimplified. In some cases they do not recognize the signal settings, associated with today's controllers, which affect phase times. In other cases models from the literature do not recognize operational effects (e.g., progression, queue blockage, permitted left-turn, critical movement, early return to green, and stochastic effects) impacting phase times, or are applicable only to isolated signals instead of coordinated signals. The consequence is that the recommended average phase times are not practical, and the resulting performance estimates and timing plan designs are unlikely to materialize in the field. The overall goal of this paper is to describe and demonstrate an improved methodology for predicting actuated phase times. An improved methodology is needed to improve the overall timing plan design process. Experimental results indicate an improvement in the accuracy of actuated phase time calculation, based on the improved methodology.

Keywords: traffic-actuated, stochastic, phase times, progression, oversaturated, critical movement, degree of saturation, Highway Capacity Manual, TRANSYT-7F, CORSIM

## INTRODUCTION

Research suggests that traffic congestion in U.S. cities has grown rapidly in recent years, and that "a full array of solutions and measures are essential in addressing the mobility problem" (1). While no single device or strategy can be expected to "solve" the problem of traffic congestion by itself, signal timing analysis and optimization has been demonstrated to be a cost-effective strategy (2, 3). Various signal system analysis models are now available for practitioners and engineers.

### Problem Statement

Stochastic simulation models offer a realistic interpretation of traffic flow and performance, but their function is limited to evaluating the performance of a specified operational alternative, and they require much more input data than a simpler deterministic model. Deterministic models, which are more efficient in terms of their input data requirements and program running times, can attempt to find the "best" set of operating parameters for a specified performance objective. However, they often lack certain detailed treatment of traffic flow characteristics found within the stochastic simulation models. The simplifying assumptions required to support a more efficient analysis have, in many cases, compromised the quality of the final product. The result is that the performance estimates from deterministic analyses, in addition to the timing plans being implemented in many traffic control systems today, still have significant room for improvement in terms of their accuracy and their correlation with real-world field conditions.

One of the most promising areas for improvement is the modeling of "traffic-actuated" control, in which instantaneous information from traffic detectors is incorporated into the control tactics for fine-tuning intersection performance. Existing deterministic models apply the simplifying assumption of non-fluctuating, average phase times, in which the operation at each intersection is characterized by a series of fixed-duration intervals. When applicable, the optimization process determines the optimal duration and position of each interval under the assumption that phase times do not fluctuate between cycles. Some adjustments are made in an attempt to represent the effects of fluctuating phase times, but the results to this point have not proven to be entirely satisfactory.

### Objectives

The objectives of this paper are, 1) to identify specific shortcomings of the existing models, 2) describe a set of proposed modeling improvements, and 3) to test the relative performance of different strategies for prediction of actuated phase times within deterministic traffic analysis models.

## **BACKGROUND**

### **TRANSYT Deterministic Traffic Analysis Model**

The Traffic Network Study Tool (TRANSYT) (4) was developed by the Transport Research Laboratory in the United Kingdom. For decades now, it has been used in various parts of the world for traffic control system timing design and evaluation. TRANSYT is referenced throughout this paper, and is a typical example of a deterministic model that stands to benefit from improved prediction of traffic-actuated phase times. TRANSYT-7F (5) was developed as a derivative work by the University of Florida Transportation Research Center, and is also referenced throughout this paper.

### **Target Degree of Saturation Strategy**

There are several ways to model actuated control, including the prediction of vehicle delay, and the design of control parameters such as detector layout and the maximum green setting. This paper focuses on one specific aspect of actuated control modeling, namely the prediction of average phase times. Actuated phase time calculation models from the literature have generally followed two strategies: 1) the target degree of saturation strategy, and 2) the queue service time -- green extension time strategy.

The target degree of saturation strategy (5, 6, 7, 8, 9) specifies that actuated phases facilitate efficient signal operation when operating at predictable degrees of saturation, often slightly below capacity. The primary goal of an actuated phase is to terminate the right-of-way as soon as the initial queue of vehicles has been served. Ideally the degree of saturation would be 100%, and the right-of-way would be terminated immediately after the queue was served.

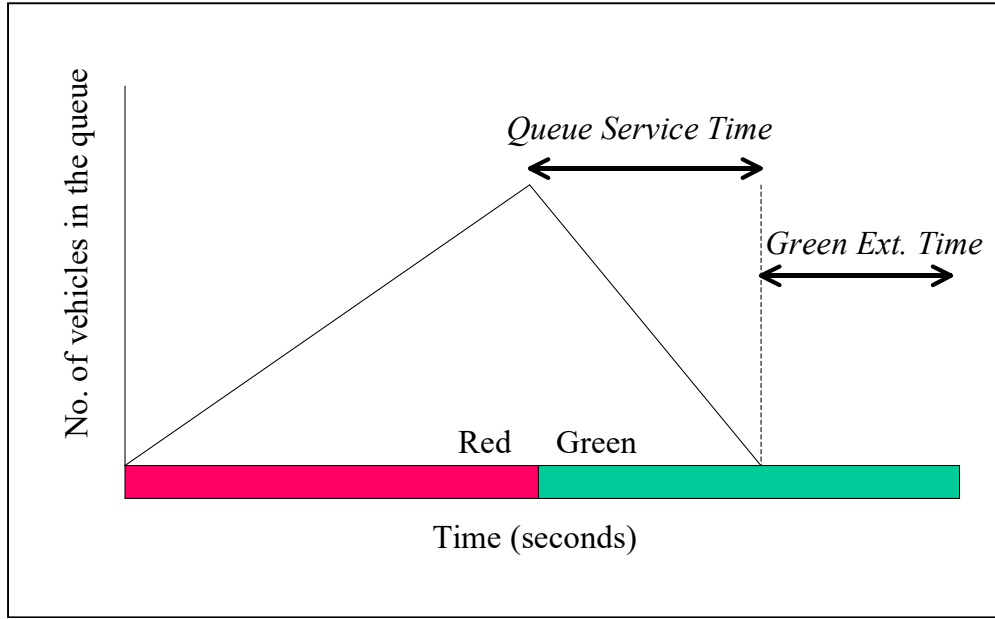
In the field, however, drivers in a standing queue are sometimes not paying attention, and conservative drivers sometimes prefer a large gap between themselves and their lead vehicle. Because of this, detectors sometimes sense a gap between vehicles before the queue has been served, causing premature phase termination and excessive delay times. In practice, there should be at least a 2- or 3-second gap requirement to avoid premature phase termination.

Because of this required 2- or 3-second gap, the degree of saturation tends to be somewhat lower than 100% on actuated phases. Some models from the literature (6, 7, 8) have typically recommended a fixed degree of saturation such as 85%. Another model (9) has used more detailed calculations to estimate the degree of saturation on actuated phases.

### **Queue Service Time -- Green Extension Time Strategy**

The queue service time -- green extension time strategy (7, 10) specifies that the length of an actuated phase can be estimated by summing the queue service time ( $gs$ ), the green extension time ( $ge$ ), and the intergreen (yellow plus all-red) time. The queue service time represents the amount of time necessary to serve the initial standing queue. Deterministic models are able to apply the queue accumulation polygon (QAP) concept in order to estimate queue service times.

Figure 1 illustrates a polygon in the shape of a triangle, implying that the rate of arrivals on red and departures on green are both uniform. The shape of the polygon is useful in determining queue service time and other measures. The shape of the polygon can become complex in response to field conditions causing non-uniform arrivals or departures, allowing queue service time to be computed accurately under various conditions.



**Figure 1: Sample Queue Accumulation Polygon**

Equations 1 and 2 (7) provide an example of how queue service time can be computed as an algebraic interpretation of the queue accumulation polygon.

$$g_s = f_q \frac{q_r r}{(s - q_g)} \quad (1)$$

where,

$q_r$  ,  $q_g$  = red arrival rate (veh/s) and green arrival rate, veh/s, respectively,  
 $r$  = effective red time, s,  
 $s$  = saturation flow rate, veh/s, and  
 $f_q$  = queue calibration factor

$$f_q = 1.08 - 0.1 \left( \frac{\text{actual green}}{\text{maximum green}} \right)^2 \quad (2)$$

Although the queue accumulation polygon is a strong indicator of queue service time, it is not as useful in determining the green extension time. Figure 1 illustrates that the green extension time takes place beyond the polygon boundaries. The green extension time represents the amount of time necessary for a detector to locate the necessary gap (typically 2 or 3 seconds) in the traffic stream, following the queue service time. Equations 3, 4, and 5, developed by Akcelik (11), illustrate how green extension time can be computed as a function of several actuated control parameters.

$$g_e = \frac{e^{\lambda(e_0 + t_0 - \Delta)}}{\phi q} - \frac{1}{\lambda} \quad (3)$$

where

$e_0$  = unit extension time setting

$t_0$  = time during which the detector is occupied by a passing vehicle

$$t_0 = \frac{(L_d + L_v)}{v} \quad (4)$$

where

$L_v$  = vehicle length, assumed to be 5.5 m

$L_d$  = detector length, DL, m,

$v$  = vehicle approach speed, SP km/h

$\Delta$  = minimum arrival (intra-bunch) headway, s,

$\phi$  = proportion of free (unbunched) vehicles, and

$\lambda$  = a parameter calculated as:

$$\lambda = \frac{\phi q}{1 - \Delta q} \quad (5)$$

where  $q$  is the total arrival flow, veh/s for all lane groups that actuate the phase under consideration.

Once the queue service times and green extension times have been calculated, the phase times are calculated by adding on the yellow and all-red times, which are known values to the analyst, and do not require modeling or calculation. Once the phase times have been calculated, additional recalculations become necessary, in adherence to numerous constraints (maximum green, minimum green, background cycle length, etc.), and because actuated phase times are affected by other actuated phase times within the same signal or network. When a background cycle length is in effect, the non-actuated, major street through movement phase typically receives all of the remaining green time within the cycle, following computation of the actuated phase times.

## MOTIVATION

The initial objective of this paper is to identify shortcomings of the existing models. Regarding the target degree of saturation strategy, recommended phase times are potentially impractical because of the inherent difficulty in controlling degrees of saturation that are likely to occur in the field. Although actuated degrees of saturation are desired to be somewhat lower than 100%, the actual degrees of saturation can vary widely, and can even exceed 100% in certain cases. Moreover, these degrees of saturation are a function of several control parameters (minimum green, maximum green, gap setting, detector layout) as well as complex and unpredictable traffic flow behavior including progression, queue spillback, permitted movement, and stochastic effects. Because the degrees of saturation are difficult to control, the model-recommended phase times are sometimes impossible to achieve in the field.

Regarding the queue service time -- green extension time strategy, available models from the literature are oversimplified in a few ways. The queue service time and green extension time formulas are primarily designed to analyze isolated signals, and are not detailed enough to realistically model coordinated signals and network-wide traffic flow patterns. When analyzing isolated signals, the existing methodologies do not provide the proper treatment for stochastic effects, which may compromise accuracy of the results.

## SPECIFIC SHORTCOMINGS AND ASSOCIATED PROPOSED IMPROVEMENTS

The queue service time -- green extension time modeling strategy was chosen as the underlying structure for an effective deterministic model. Upcoming discussions will demonstrate that actuated phase times are calculated with greater accuracy when it becomes possible to analyze the queue service times and green extension times in greater detail.

The following is a discussion that identifies six specific shortcomings and, to address the second overall objective of this paper, proposes candidate improvements for each.

### **Candidate Improvement #1: Retrieval of Queue Service Times from the Queue Profile**

The first improvement is related primarily to coordinated signals. Traffic flow patterns at coordinated signals are subject to progression, queue blockage, and permitted movement effects. Each of these effects potentially changes queue service times at a coordinated signal, including the uncoordinated approaches. Queue service time formulas are unable to realistically model network-wide progression, queue blockage, and permitted movement effects.

Deterministic models such as TRANSYT continuously monitor queue lengths throughout the network. These queue lengths are appropriately sensitive to any complications simulated by the program, including progression, queue blockage, and permitted movement effects. Therefore, obtaining unadjusted queue service times from a program-generated "queue profile" is an ideal mechanism for recognizing network-wide traffic flow conditions.

Other deterministic models such as the Highway Capacity Manual signalized intersection procedures (7) should be able to adjust queue service times in response to permitted movement effects, but are limited in analyzing progression effects, and unable to analyze queue blockage effects. For models such as this, obtaining queue service times from a queue profile may not be practical.

### **Candidate Improvement #2: Adjustment of Queue Service Times for Stochastic Effects**

The second candidate improvement is one out of three that are applicable to isolated signals as well as coordinated signals. Three out of six candidate improvements (including this one) are applicable to both isolated and coordinated signals.

Equation 1 listed earlier contains the variable  $f_q$  denoting the queue calibration factor. This factor is unrelated to the queue accumulation polygon, and is included in the formula "to account for randomness in arrivals in determining the average queue service time" (7). Randomness indeed has a significant impact on phase times when there are "competing" movements or lanes within the phase. In this paper, the effect of competing movements or lanes will be referred to as stochastic effects.

In the literature, actuated phase times are designed to handle the critical movement having the highest flow ratio ( $v/s$ ), or volume divided by saturation flow rate. However, non-critical movements or lanes are sometimes responsible for extending the phase. For example, a non-critical movement could be responsible for terminating the phase length on 3 out of 10 cycles. Because of this, average phase times are higher in the field than they are in a model that ignores the non-critical movements. An exception occurs when non-critical movements or lanes have an extremely low flow ratio, such that they have little chance of extending the phase.

Upcoming experimental results indicate that an adjustment to the original model-calculated queue service time allows for improved accuracy. The results also indicate that such an adjustment, to compensate for stochastic effects, is only necessary when competing movements or lanes exist in the same phase. For example, no queue service time adjustments are necessary for an actuated phase completely governed by a single lane, such as a protected left-turn phase having a green arrow. Because of this, a selective application of the queue calibration factor is indicated, as opposed to automatically applying the factor, as the Highway Capacity Manual queue service time formula does.

### **Candidate Improvement #3: Green Extension Times that Reflect the Flow Profile**



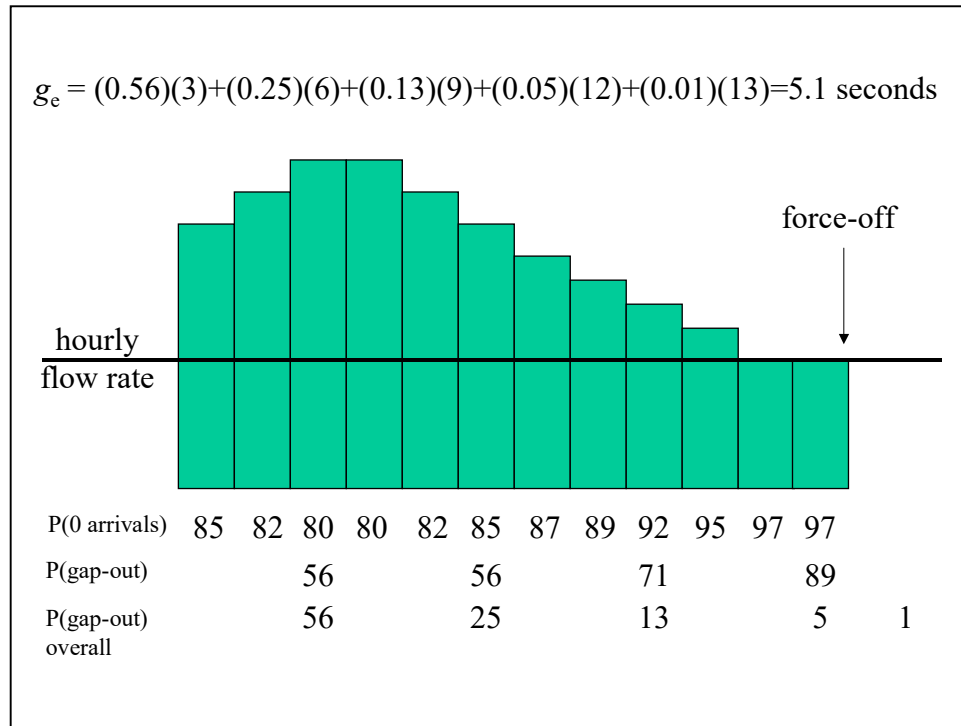
In theory this candidate improvement is primarily applicable to coordinated signals. Upcoming experimental results indicate an improvement modeling isolated signals as well. It was stated earlier that traffic flow patterns at coordinated signals are subject to progression, queue blockage, and permitted movement effects, although permitted movements affect isolated signals as well. These effects also potentially change the green extension time on every phase of a coordinated signal, including the uncoordinated approaches. Uncoordinated approaches, although not directly affected by progression, may be indirectly affected due to changes in the major street phase times. Green extension time formulas are unable to realistically model network-wide progression, queue blockage, and permitted movement effects.

Deterministic models such as TRANSYT are able to monitor flow profiles throughout the network. These flow profiles are sensitive to any complications simulated by the program, such as progression, queue blockage, and permitted movement effects. Unlike queue service times, unadjusted green extension times cannot be extracted directly from the results of macroscopic simulation. However, accuracy of the predicted green extension times can improve significantly, based on flow profile-supplied information.

Figure 2 illustrates a sample calculation for computing green extension time based on the flow profile. The top row of values beginning with "85" represents the Poisson probability of zero vehicle arrivals during that step of the cycle. A probability of 85% is derived from a flow rate of 600 vehicles per hour (expected at that point in the cycle) as illustrated by equation (6):

$$P(0) = e^{-600/3600} \approx 0.85 \quad (6)$$

The second row of values beginning with "56" represents the unadjusted probability of phase termination. A probability of 56% is derived by multiplying the preceding probabilities ( $0.85 \times 0.82 \times 0.80$ ). In this example, the previous three probabilities are multiplied because a gap setting (a.k.a. unit extension) of 3 seconds is in effect, and it is necessary to have three consecutive seconds with zero arrivals before terminating the phase.



**Figure 2: Sample Calculation of Green Extension Time**

The third row of values, also beginning with "56", represents the adjusted probability of phase termination. This is best understood by considering the values while reading from left to right. The overall probability of phase termination is 56% after three seconds, which indicates a 44% chance of extending the phase. Although the unadjusted probability of phase termination is also 56% after six seconds, this must be factored down by 44% to obtain the actual probability of phase termination, in this case  $0.56 \times 0.44$  to obtain 0.25. The overall phase termination probabilities are computed in this manner until the maximum green time, or force-off point, is reached, at which time the probability of phase termination is computed as 100% minus the sum of all previous probabilities.

This green extension time model, which is based on applied probability analysis of the flow profile, computes results that reflect the level of progression. It also appears to compute realistic results even when there is no progression, and the flow profile is completely flat. Deterministic models for isolated signals, such as the Highway Capacity Manual signalized intersection procedures, could still apply this model, even though they are limited in their ability to simulate the flow profile.

#### **Candidate Improvement #4: Green Extension Time Adjustment for Stochastic Effects**

This candidate improvement is applicable to both isolated and coordinated signals. It was stated earlier that stochastic effects potentially have a significant impact on average phase times when there are "competing" movements or lanes moving during the phase. Although models in the literature typically scrutinize critical movements, non-critical movements or lanes are sometimes responsible for extending the phase, which may increase the average green extension time. Upcoming experimental results indicate that an adjustment to the original model-calculated green extension time allows for improved accuracy.

As discussed earlier, the adjustment to compensate for stochastic effects is only necessary when competing movements or lanes exist during the same phase. Under these conditions, there is a higher probability of extending the green, based on the competing traffic volumes. Simply adding the competing

traffic volumes, and computing green extension probabilities based on the total traffic volume, is not an ideal treatment. For example, a 200 vehicle per hour (vph) movement together with a 240-vph movement in the same phase, extends the green differently than a single movement having 440 vph. Instead of simple addition, it is appropriate to use the additive law of probability (12) illustrated by equation (7):

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \quad (7)$$

Calculation of the adjusted volume, given a 200-vph movement moving together with a 240-vph movement, is illustrated by equation (8):

$$\text{Adjusted Volume} = 3600 \times \left[ \left( \frac{200}{3600} \right) + \left( \frac{240}{3600} \right) - \left( \frac{200}{3600} \times \frac{240}{3600} \right) \right] = 427 \text{ vph} \quad (8)$$

Ideally this calculation is repeated throughout the cycle, to reflect a changing flow profile, if the signal is coordinated. Subsequently the adjusted volumes can be used to compute the overall green extension times.

#### **Candidate Improvement #5: Critical Movement Search**

This candidate improvement is also applicable to both isolated and coordinated signals. Under simple conditions the critical movement is determined by calculating the flow ratio (volume divided by saturation flow rate) for each movement. The movement having the highest flow ratio is considered critical, and most methodologies focus their attention on modeling this movement. In the case of overlapping actuated phases, the critical movement is the one that would require the longest phase time, meaning that phase times for each individual movement must be calculated according to the "full-blown" model (i.e., computation of queue service time and green extension time).

In determining the unadjusted queue service time, it is necessary to determine which movement is responsible for having the queue that extends further into the cycle than any other movement's queue. Once this critical queue service time has been obtained, an adjustment for stochastic effects may be applied as described earlier. In determining the unadjusted green extension time, the previously computed queue service time indicates where in the cycle the flow profile can begin to be examined.

#### **Candidate Improvement #6: Dynamic Green Time Reallocation within the Cycle**

This final candidate improvement is only applicable to coordinated signals having a background cycle length. Deterministic actuated control models from the literature specify that, once the actuated phase times have been determined, any unused green time within the cycle is assigned to the major street non-actuated phase. In the field, actuated phases with higher degrees of saturation have been observed to absorb unused green time (from the preceding actuated phases) for themselves, such that the unused green time does not reach the major street non-actuated phase. Moreover, modern controller software can further control the allocation of unused green.

Because of the potential for this type of behavior, it is necessary to calculate the actuated phase times sequentially, beginning with the "first" actuated phase that immediately follows the major street non-actuated phase. When calculations are performed in this order, it is possible to correctly reassign the unused green time.

### **EXPERIMENTAL DESIGN**

The third and final overall objective of this paper is to test the relative performance of different modeling strategies, both new and old. To test the proposed improvements for predicting actuated phase

times, it is necessary at the outset to choose a specific deterministic model for improvement, as well as a stochastic simulation model for evaluation of the improvements. These two choices will establish the scope of the experimental design.

### **Validation Involving CORSIM As a Surrogate for Field Data**

Ideally testing and validation of the experimental results, involving improved prediction of traffic-actuated phase times, would be accomplished through the collection and analysis of field data; however, an adequate empirical validation would require resources several orders of magnitude beyond those available to this project. This limitation dictates the need for simulation as a surrogate for field data collection.

The choice of a simulation model is not difficult in this case. The CORridor SIMulation (CORSIM) (13) model has been developed and enhanced continually over the past 30 years by the Federal Highway Administration (FHWA). It was designed specifically for the purpose intended by this project (i.e., a surrogate for field data) and is used extensively for this purpose.

CORSIM is especially strong in the treatment of actuated control. It models this type of control explicitly, using a software module based on the real-time control logic contained within the actual field hardware (13). In addition, previous research indicates that actuated phase times simulated by CORSIM closely replicate those that are observed directly in the field (10). It is noted that CORSIM is unable to simulate certain esoteric actuated control strategies, and is unable to simulate the unusual or unexpected behavior of certain drivers. Nevertheless, CORSIM is used for testing and validation of the phase time calculation enhancements described within this paper.

### **Implementation of Recommended Improvements within TRANSYT-7F**

The TRANSYT program introduced earlier is a logical choice for the deterministic model. The specific version of the TRANSYT model used in this study is TRANSYT-7F, release 9.1. The recognition of TRANSYT-7F among practitioners, combined with the institutional support of the University of Florida, make it a natural choice for purposes of this project. In addition, the ability of TRANSYT-7F to provide network-wide information on the flow profile and queue profile allows for better testing of the recommended improvements described earlier.

### **Candidate Models**

The first model (hereinafter referred to as Model 1) to be tested is the original target degree of saturation model as implemented within TRANSYT-7F. Model 2 possesses five of the six proposed improvements, but implements Akcelik's green extension time formula. Model 3 contains all six proposed improvements described earlier in this paper. Results from these three deterministic models will be compared against results from the CORSIM stochastic simulation model.

By choosing the candidate models this way, it is not possible to gauge the effectiveness of each of the six proposed candidate improvements. Instead, the most highly refined model having all six new elements is to be compared to the original target degree of saturation strategy, and Akcelik's green extension time formula is directly compared to the proposed strategy, which computes green extension times to reflect the flow profile.

### **Testing Conditions**

Table 1 lists the testing conditions that were designed to stress specific aspects of candidate models. The initial experiments (#1a, b, c) involve a simple two-phase timing plan, vehicle arrivals on two approaches (northbound and eastbound) only, and through movements only. Experiment #1d involves the same conditions from #1a, b, and c, except that vehicle arrivals occur on four approaches in order to examine competing movement effects. A postprocessor was used in the initial experiments to compute average phase times in CORSIM based on two hours of simulation, which allows for precise

determination of CORSIM's average phase time. A range of gap settings (2-5 seconds) and traffic volumes (300-700 vph) are used in order to ascertain behavior under conditions that are both stable and unstable, congested and uncongested.

**Table 1: Experimental Design**

<u>Experiment Characteristics</u>	<u>Experiment Number</u>								
	1				2		3		4
	-a-	-b-	-c-	-d-	-a-	-b-	-a-	-b-	
Single Intersection	*	*	*	*					
Arterial Street					*	*	*	*	*
Background Cycle Length	*		*	*	*	*	*	*	*
Multilane Lane Groups			*		*	*	*	*	*
Competing Links				*	*	*	*	*	*
Progression					*		*	*	*
Permitted Left-Turns							*	*	
Optimization									*

Experiments #2, #3, and #4 introduce additional complications to obtain further insight. CORSIM and TRANSYT-7F data sets for these experiments were obtained from a University of Florida project for developing traffic engineering benchmark data (14). The arterial street conditions involve 1000-foot signal spacing, and differing operating characteristics at each individual intersection, including traffic volume, degree of saturation, lane channelization, phase sequence, and left-turn treatment. Numerous intersections having competing lanes and multilane groups allows for additional testing of stochastic effects beyond experiment #1.

Whereas the CORSIM average phase times from experiment #1 are based on two hours of simulation, average phase times from experiments #2 and #3 are based on 30 minutes of observed simulation (animation). Experiment #2 involves multiple offset designs, in order to evaluate various progression effects. Experiment #3 introduces permitted left-turns as an additional complication.

## EXPERIMENTAL RESULTS

### Experiment #1: Single Intersection

Table 2 illustrates the results of experiments #1a, b, c, and d. Table values indicate average actuated phase times with the exception of experiment #1b results, in which table values indicate the actuated cycle lengths. Some of the actuated cycle lengths appear to be unrealistically low. In general, model testing does not require the type of high minimum green times and cycle lengths typically implemented in the field to satisfy safety requirements, although such constraints are nonetheless present in experiments #2 and #3.

**Table 2: Experiment #1 Results**

Model*	-a- Semi-Actuated				-b- Fully-Actuated			-c- Semi-Act Multilane				-d- Semi-Act Competing Link			
	C	1	2	3	C	2	3	C	1	2	3	C	1	2	3
300vph Gap															
2	19.6	18.0	20.3	19.0	17.4	20.5	17.1	20.4	18.0	20.6	19.5	22.4	18.0	21.5	20.2
3	20.7	18.0	21.5	20.4	20.5	24.1	21.2	22.2	18.0	22.5	21.6	24.4	18.0	23.3	22.4
4	21.9	18.0	22.6	21.8	23.1	28.0	25.6	24.7	18.0	24.2	24.3	25.9	18.0	25.2	24.8
5	23.4	18.0	24.1	23.4	27.7	32.1	29.9	26.2	18.0	26.3	27.2	29.1	18.0	27.4	27.6
500vph Gap															
2	27.9	28.0	28.6	27.6	25.8	29.7	24.1	29.8	28.0	29.8	28.8	30.3	28.0	31.0	29.4
3	29.4	28.0	30.1	29.2	30.2	35.6	31.7	32.7	28.0	31.6	31.7	33.5	28.0	33.9	32.8
4	31.1	28.0	31.6	31.3	38.1	42.1	41.0	37.3	28.0	34.7	36.5	35.5	28.0	36.7	36.8
5	33.0	28.0	33.3	33.3	44.5	50.0	49.8	41.9	28.0	38.4	40.5	40.5	28.0	41.5	41.2
700vph Gap															
2	36.7	39.0	37.5	36.3	49.8	57.3	47.2	38.9	39.0	39.7	38.0	39.5	39.0	41.3	39.5
3	37.5	39.0	38.7	38.3	59.2	70.9	64.9	44.2	39.0	41.2	42.3	45.1	39.0	44.9	43.2
4	39.4	39.0	40.6	40.4	79.2	87.0	87.2	49.9	39.0	45.5	47.5	48.7	39.0	52.3	47.9
5	42.5	39.0	43.0	43.0	92.7	107.6	111.0	56.9	39.0	51.7	52.4	53.6	39.0	65.6	51.7
Error	9.2	0.6	0.2		52.7	41.0		75.6	6.4	3.1		69.9	14.1	2.0	
*Models:															
C: CORSIM stochastic simulation model															
1: Original TRANSYT-7F target degree of saturation model															
2: Model with 5 of 6 candidate improvements, plus Akcelik's green extension model															
3: Model with all 6 proposed improvements, including the flow profile-based green extension model															

The three candidate models are judged according to their correlation with CORSIM. Because it is difficult to evaluate the correlation by inspection, the average error for each model is listed at the bottom of the table. Calculation of the error value is illustrated by equation (9):

$$\text{Error} = (\text{Model Phase Time} - \text{CORSIM Phase Time})^2 \quad (9)$$

First, the results from Table 2 indicate that Model 2 and Model 3, which are based on the queue service time -- green extension time strategy, achieve much more consistent correlation with CORSIM than Model 1, which is based on the target degree of saturation strategy. This result was fully expected because the target degree of saturation strategy does not currently recognize numerous factors affecting actuated phase times.

Second, results indicate that Model 3, containing all six recommended modeling improvements

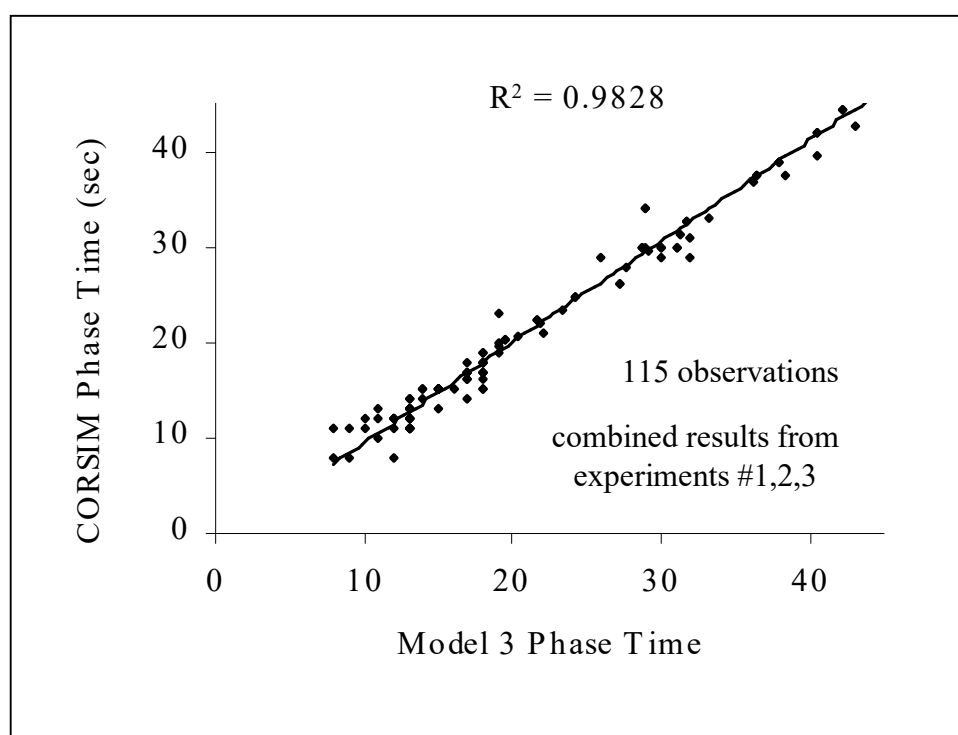
from this paper, achieves moderately better correlation than Model 2, which contains five out of six recommended improvements plus Akcelik's green extension time formula. This result was unexpected, because Model 2 was designed to handle isolated intersections.

Third, correlation becomes poor for fully-actuated signals having high volumes. This is not as troublesome as it appears. Fully-actuated cycle lengths are naturally unstable under high volumes and high gap settings. Not only is this a difficult situation to model with precision, but also cycle lengths in the field would be likely to fluctuate wildly, based on the exact pattern of vehicle arrivals.

Finally, results for Model 1 are not listed within experiment #1b for fully-actuated control. This is because the model requires a fixed cycle length to compute the target degree of saturation. Ability to examine fully-actuated controllers (having no fixed cycle length) is another advantage of the queue service time -- green extension time strategy.

### Experiments #2 and #3: Arterial Street Evaluation

Table 3 values indicate the results of experiments #2 and #3, as well as combined results from all three experiments. Table values indicate the coefficient of determination ( $R^2$ ), where values closest to 1.0 indicate the best correlation. Correlation of Model 3 with CORSIM, based on all three experiments, is also graphically illustrated in Figure 3.



**Figure 3: Traffic-Actuated Phase Time Comparison -- CORSIM vs. Model 3**

Similar to the results from experiment #1, Table 3 results indicate that Model 2 and Model 3 demonstrate much better correlation with CORSIM than Model 1. Although Model 3 consistently has slightly better correlation than Model 2, the biggest difference is observed on internal movements in experiment #2. This is expected because Model 3 is designed to reflect progression effects and the flow profile. Overall correlation is mediocre on internal movements, possibly due to insufficient variety of the internal movement phase times (10-15 seconds) and internal movement lengths (1000 feet).

**Table 3: Experiment #2 and #3 Results**

Model	Coefficient of Determination ( $R^2$ ) Values				Experiments
	Experiment #2		Experiment #3		#1 #2 and #3
	Internal	External	Original	Calibrated	Overall
M3	0.8205	0.9671	0.9190	0.9362	0.9828
M2	0.7593	0.9623	0.9044	0.9292	0.9755
M1	0.6418	0.8099	0.6538	0.6473	0.8187

Overall correlation is also mediocre in the first part of experiment #3, possibly due to differences in the permitted left-turn movement models of TRANSYT-7F and CORSIM. Correlation improves in the second part of experiment #3 when the CORSIM permitted left-turn model is calibrated to become consistent with TRANSYT-7F; i.e., zero left-turn jumpers per cycle and two left-turn sneakers per cycle. Experiment #3 provides evidence that even though traffic served during the permitted phase affects the actuated phase times, Model 2 and Model 3 are able to reflect this with their queue service times, but Model 1 is not able to reflect this. The same could be said regarding queue blockage effects, which are elegantly handled by the proposed model due to its queue profile analysis.

#### **Experiment #4: Arterial Street Optimization**

In this experiment, optimization of offsets is performed within TRANSYT-7F, following determination of the actuated phase times. Following optimization, the recommended offsets are coded into the corresponding CORSIM data file. In both CORSIM and TRANSYT-7F, offsets for this experiment are referenced to the "yield points", or the ending points of the coordinated, major street through movement phases.

Theoretically, if the actuated phase times are realistic, this should improve the optimization process and achieve better results within CORSIM. If the model-predicted actuated phase times are unrealistically low, the amount of green time available to the major street would be unrealistically high. The optimization process may recommend offsets to achieve a wide progression band that has little chance of materializing in the field. It also follows that measures of effectiveness reported by the optimization model are likely to be overly optimistic, if the amount of green time available to the major street is unrealistically high.

Table 4 illustrates that offsets resulting from Model 3 are indeed more effective, according to CORSIM, than those resulting from Model 1. Correct calculation of actuated phase times is not an optimization technique per se, but facilitates optimization by providing the optimization model (and the analyst) with more realistic information.

**Table 4: Optimization Comparison under Old and New Methodologies**



Offset design generated by:			CORSIM results based on timing from:		
Node	Old T7F	New T7F		Old T7F	New T7F
6	0	0	Control Delay	25.6	23.0
7	15	40	Total Delay	29.7	26.9
8	29	57	Stop Delay	22.0	19.6
9	41	63	Vehicle Trips	8116	8167
10	53	65			
11	27	50			
12	27	81			
13	57	25			
14	18	86			
15	64	34			

## CONCLUSIONS AND RECOMMENDATIONS

The existing models are constrained by specific shortcomings. The proposed model utilizes the queue service time -- green extension time framework from the literature, but incorporates six candidate improvements that allow for better prediction of traffic-actuated phase times by deterministic traffic analysis models. For a model that estimates actuated phase times to be realistic, it must reflect the flow profile, the queue profile, and the stochastic effects of multiple lanes and competing movements. The model used to predict individual phase times must be applied to all actuated movements, to confirm which one produces the critical actuated phase time. Following computation of the individual phase times, reallocation of unused green time must be done in a dynamic manner reflecting the true operation of the controller.

Some of these improvements are applicable to coordinated signals only, but other improvements are applicable to isolated signals as well. Although only tested on single intersections and arterial streets, the proposed model is naturally extensible to a network of signals. Improved actuated phase time prediction is expected to improve the effectiveness of evaluation-only models such as the Highway Capacity Manual signalized intersection procedures, and optimization models such as TRANSYT-7F.

Additional research is recommended regarding the overall signal timing design process. The findings of this paper imply a somewhat inefficient two-step process where actuated phase times are first optimized using the target degree of saturation model, and then "validated" using the proposed model. Additional research should confirm that the two-step process facilitates improved signal timing. Secondly, it would be helpful to research the possibility of a more efficient one-step model combining the strengths of the target degree of saturation model, and the proposed model from this paper.

Further examination of stochastic effects is also recommended for future research. The queue calibration factor ( $f_q$ ) formula does not explicitly consider competing movement volumes, so this formula could be further developed. Other possible areas of related research include the effect of detector setback, phase skipping, multi-cycle simulation, and offset optimization, on actuated phase times.

## REFERENCES

1. Lomax, T. and D. Schrank, "Traffic Jams: Not Just Big-City Problem," Texas Transportation Researcher Vol. 34 No. 3, College Station, TX, 1998.

2. Deakin, E.A., May, A.D. and A. Skabardonis, "Energy Savings from Signal Timing Optimization: Evaluation of California's Statewide Program," *Compendium of Technical Papers*, Institute of Transportation Engineers, San Francisco, CA, 1984.
3. Wallace, C.E., K.G. Courage and D.P. Reaves, "National Signal Timing Optimization Project -- Final Evaluation Report," prepared for the Federal Highway Administration by the University of Florida Transportation Research Center, 1981.
4. Robertson, D.I., "TRANSYT: Traffic Network Study Tool," Fourth International Symposium on the Theory of Traffic Flow, Karlsruhe, Germany, 1968.
5. Wallace, C.E., K.G. Courage, M.A. Hadi, and A.C. Gan, TRANSYT-7F User Guide, Volume 4 of a series prepared for FHWA by COURAGE AND WALLACE, Gainesville, FL, March 1998.
6. Skabardonis, A. and A. Weinstein, "TRANSYT-7FC, TRANSYT Model for Actuated Signals," Institute for Transportation Studies, University of California, Berkeley, CA, April 1988.
7. Highway Capacity Manual, Special Report 209, Transportation Research Board, 1985, 1994, 1997, 2000.
8. Courage, K.G., "Signal Operations Analysis Package Version 84 User's Manual," Transportation Research Center, University of Florida, Gainesville, Florida, 1985.
9. Akcelik, R. and M. Besley, SIDRA 5 User Guide, ARRB Transport Research Ltd., Vermont South, Victoria, Australia, January, 1999.
10. Courage, K.G., Akcelik, R., and D. Fambro, "Capacity Analysis of Traffic-Actuated Intersections, NCHRP Project 3-48," prepared for NCHRP by the University of Florida Transportation Research Center, Texas Transportation Institute, and ARRB Transport Research Ltd., December 1996.
11. Akcelik, R., "Estimation of Green Times and Cycle Length for Vehicle-Actuated Signals," Paper No. 94 0446, 73rd Annual Meeting of Transportation Research Board, Washington D.C., January, 1994.
12. Montgomery, D.C. and G.C. Runger, Applied Statistics and Probability for Engineers, Second Edition, John Wiley and Sons, New York, NY, 1999.
13. CORSIM Users Manual Version 1.04, ITT Systems and Sciences Corporation, Colorado Springs, CO, March, 1998.
14. Henry, M.A., "The Generation of an Arterial Benchmark Data Set and Its Application to the Effect of Phase Sequence on Arterial Progression," Masters Degree Report, Gainesville, FL, 1999.

## **EDITORIAL COMMENTS (December 2003)**

Although it was unknown at the time of writing the paper, experiment #4 was apparently a misapplication of both Model #1 (i.e., the target degree of saturation model) and Model #3 (i.e., the actuated estimation model). In the TRR Paper, experiment #4 implies that force-offs and maximum greens must somehow be determined and specified prior to the estimation or computation of actuated phase times. However, what is not mentioned is that the target degree of saturation model is in fact designed to determine these same force-offs and maximum greens. The target degree of saturation model was not designed to estimate actuated phase times based on previously determined force-offs and maximum greens, but nevertheless, this is how the model was applied in experiment #4. In addition, the actuated estimation model is not well-suited for the design and optimization of green splits, but this is how the model was applied in experiment #4.

Although the TRR Paper is effective at summarizing the actuated estimation modeling methodology and the preliminary validation experiments, other parts of the documentation (Traffic-Actuated Control, Actuated Estimation Methodology, etc.) should be used as guides for proper model application. When evaluating existing conditions, the actuated estimation model is useful in determining phase times that are likely to occur in the field, based on the known maximum greens and force-offs. In the context of optimization, the actuated estimation model is useful in verifying whether the program-optimized green splits are feasible, and is useful in determining maximum greens and force-off settings for implementation in the field. When performing a multi-period analysis, the actuated estimation model is potentially helpful in determining unique phase times that will materialize in the field during each time period.

The traditional target degree of saturation model (within the initial timing model) is still an important part of the optimization process. This is the only "design-oriented" model available, to facilitate location of the optimal actuated phase times. However, the "evaluation-oriented" actuated estimation model should be applied following optimization, to confirm that the optimized actuated phase times are feasible. Because the target degree of saturation model is not simulation-based, it has the potential to occasionally recommend infeasible phase times.



# Appendix B: Traffic Network Study Principles

## Calibration

"The process of comparing model parameters with real-world data to ensure that the model realistically represents the traffic environment. The objective is to minimize the discrepancy between model results and measurements or observations." (from HCM2000 [12])

## Empirical Model

A model that "describes system performance based on statistical analysis of field data". (from HCM2000 [12])

## Analytical Model

A model that "relates system components using theoretical considerations tempered, validated, and calibrated by field data" (e.g., Highway Capacity Manual procedures). (from HCM2000 [12])

## Simulation Model

A program that "uses mathematical models to conduct experiments with traffic events on a transportation facility or system over extended periods of time" (e.g., CORSIM). (from HCM2000 [12])

## Macroscopic Model

A simulation process (e.g., T7F) that "keeps track of flow rate variables" (flow profiles, queue profiles, etc.). (from HCM2000 [12])

## Microscopic Model

A simulation process (e.g., CORSIM) that "keeps track of location, speed, acceleration, and performance of individual vehicles". (from HCM2000 [12])

## Deterministic Model

A model (e.g., HCM, T7F) for which a given set of input conditions will always yield a fixed set of output conditions.

## Stochastic Model

A model (e.g., CORSIM) for which a given set of input conditions may yield various sets of output conditions.

## Network Partitioning

Traffic operational interaction between intersections can only happen in two ways:

1. Platoon progression
2. Queue spillback

Where there is no possibility of progression or queue spillback effects, the network can (and sometimes should) be divided (partitioned) at that location, for separate analyses.

## Phase Failures

A phase failure occurs when the queue is not completely served during the available effective green time. When phase failures occur consistently and repeatedly, this is an indicator of oversaturated conditions, and a degree of saturation greater than 100%. When phase failures occur intermittently, this is an indicator of near-saturated conditions, and a degree of saturation somewhat lower than 100%. Persistent phase failures are the source of residual delay (d3), whereas intermittent phase failures are the source of random (a.k.a. incremental) delay (d2).

## Red Light Running

The following conditions from the literature were summarized in a research report by Bonneson and Son [23] as factors that could increase the probability of red light running:

- dense platoons arriving toward the end of green
- lower cycle lengths (due to the increased frequency of red lights per hour)
- expected delay

Regarding dense platoons arriving toward the end of green, this consideration is naturally only applicable to internal links having a nearby upstream signal. The Profile View screen can be used to determine whether platoons are arriving during the beginning of green, or during the end of green.

Regarding lower cycle lengths, it is tricky to recommend these for increased safety, because it could potentially increase delay, thus conflicting with factor #3 (expected delay). Perhaps the best bet is to simply optimize the signal timing as much as possible, using the normal TRANSYT-7F optimization procedures.

## Internal Links

Internal links have a "nearby" upstream signal, may experience platooned vehicle arrivals, and may cause queue spillback.

## External Links

External links have no nearby upstream signal, and experience random vehicle arrivals (with a uniform flow profile).

## Peak Hour Factor

The peak hour factor (PHF) allows for analysis of the most important time period, which is typically the peak 15-minute period. If the analysis period is a full hour, the PHF should be set equal to 1.0.

$$\text{PHF} = \frac{\text{hourly volume (vph)}}{4 \times (\text{peak 15 minute volume})}$$

Peak hour factors can be coded in the Lane Configuration (Edit > Lanes) screen, if desired. By default, volumes are coded as hourly flow rates (vph). Normally, the analyst would either:

- 1) Code the observed hourly flow rate in the field, set PHFs equal to (hourly volume / (4 \* peak 15-minute volume)), and set the analysis period equal to 15 minutes, or,
- 2) Code volumes equal to (peak 15-minute volume \* 4), set PHFs equal to 1.00, and set the analysis period equal to 15 minutes.

There is also an option for coding vehicles per time period (vpp), under Options > Preferences. So there is a third option:

- 3) Specify vpp coding under Options > Preferences, code the actual 15-minute volumes during the peak period, set PHFs equal to 1.00, and set the analysis period equal to 15 minutes.

Coding strategy #1 above is recommended in most situations, because of its consistency with HCM [12] example problems, but the other coding options would yield identical results if applied properly. As discussed in the Analysis Period section of Appendix B, a multi-period analysis is recommended for severely oversaturated conditions. When a multi-period analysis is performed, volumes may be coded in either the vph or the vpp format, but all PHFs must be set equal to 1.00.

## Analysis Period

- Also known as "Duration"
- Length of time being analyzed
- Coded on the Edit > Analysis screen, or the Edit > Optional > Global > Run Controls screen
- Traffic volume demand is assumed to remain constant during the analysis period
- To analyze non-constant and severely oversaturated traffic volumes, multi-period analysis is recommended
- If the network is undersaturated or mildly oversaturated, analysis of the peak 15-minute period is recommended, and the appropriate peak hour factor should be used

To describe the various degrees of network-wide oversaturation, the following categorization is helpful. A "severely oversaturated" network is one in which most links are oversaturated, i.e., most links have a degree of saturation above 100%. A "mildly oversaturated" network is one in which only a few links are oversaturated during the analysis period.

## Control Delay

The Highway Capacity Manual [12] defines control delay as the portion of total delay attributed to traffic signal operation at signalized intersections. Control delay includes initial deceleration delay, queue move-up time, stopped delay, and final acceleration delay. For more information, refer to the documentation on Delay and Level of Service.

## Objective Function

In any program or process that employs hill-climb, genetic algorithm, or brute force optimization, the optimization process is based on a series of simulation or evaluation runs having various signal timing plans. The simulation run whose timing plan results in the best performance, according to the objective function, is recommended as optimal. In TRANSYT-7F, the optimization objective function is reflected by the "performance index" (PI).

## Verification and Validation

Prior to an optimization project, field studies related to travel times and platoon dispersion, as well as advance considerations for a signal timing review process [24], are recommended. Field verification can be performed both before and after re-timing. Important considerations include field observation of progression, field measurement of actuated phase times, and moving-vehicle runs in the field.





To simplify this process, it is important to focus on the underlying traffic analysis model, instead of worrying about the optimization process. The optimization process (i.e., genetic algorithm or hill-climb) affects timing plan efficiency, but has no effect on whether the same delays and levels of service reported by the software would be observed in the field. Verifying the effectiveness of an optimization process would require changing the timing plan inside the software, perhaps manually, to determine whether the objective function could be further improved by some other process. As discussed in chapter 5, the genetic algorithm is the most effective optimization process, when applied properly.

Examples of underlying traffic analysis models include macroscopic TRANSYT simulation, microscopic CORSIM simulation, and the analytical HCS procedures. The genetic algorithm can be applied to any of these three models to achieve optimization. If traditional TRANSYT optimization is performed, the underlying traffic model is macroscopic TRANSYT simulation. If direct CORSIM optimization is performed, the underlying traffic model is microscopic CORSIM simulation. If optimization is performed inside the HCS-Signals module, the underlying traffic model is the HCM signalized intersection procedure.

To ensure that the same levels of service reported by the software will be observed in the field, an appropriate underlying traffic analysis model must first be selected. For example, if conditions are oversaturated, it is important to select a model that can accurately propagate residual queues across multiple signal cycles and/or time periods. If queue blockage exists, it is important to select a model that can realistically simulate queue spillover and queue spillback across multiple signal cycles and/or time periods. In a progressive system, it is important to select a model that can accurately simulate platoon dispersion. In a traffic-actuated system, it is important to select a model that can accurately analyze actuated control. Although macroscopic TRANSYT simulation meets these modeling criteria, CORSIM has moving-vehicle animation (to assist in verification), and HCS is more easily applied.

The selected model must then be calibrated properly, to closely approximate conditions in the field. Ideally, a simulation-only run would be performed prior to any optimization, and the results of this run would match existing conditions in the field.

Regarding field observation of progression, travel times of the vehicle platoons (or the moving-vehicle runs) should reflect free-flow speeds coded within the software. Platooning intensities observed in the field should reflect the observed flow profiles within the software. In TRANSYT, platoon dispersion can be observed in the Profile View screen, and can be calibrated using the platoon dispersion factor (PDF). In CORSIM, platoon dispersion can be observed in TRAFVU (animation), and can be calibrated using the free-flow speed distribution. Any significant mid-block source or sink flows observed in the field should be accounted for in the coding of traffic volumes, because these flows can have a significant impact on progression.

Regarding field measurement of actuated phase times, average phase times observed in the field must be similar to those in the software. Force-off and maximum green settings programmed in the field must be capable of producing any green times recommended by the software. One common mistake is to implement force-off and maximum green times that greatly exceed the average green time. This causes average green times in the field to exceed program-recommended average green times, and leads to wasted green time on the minor movements. As a rule of thumb, force-offs and maximum green times should be set equal to the optimized average green times, or perhaps slightly higher.



# References

## References

1. Wallace, C.E., K.G. Courage and M.A. Hadi, TRANSYT-7F User's Guide, Volume 4 of a series prepared for FHWA by COURAGE AND WALLACE, Gainesville, FL, March 1998.
2. Robertson, D.I., "TRANSYT: Traffic Network Study Tool," Fourth International Symposium on the Theory of Traffic Flow, Karlsruhe, Germany, 1968.
3. Robertson, D.I., "TRANSYT: A Traffic Network Study Tool," Road Research Laboratory Report, LR 253, Crowthorne, 1969.
4. Robertson, D.I. and P. Gower, "User Guide to TRANSYT Version 6," Transport and Road Research Laboratory Supplementary Report 255, Crowthorne, 1977.
5. Hunt, P.B. and J.V. Kennedy, "A Guide to TRANSYT/7," Transport and Road Research Laboratory Note UN78/11, Crowthorne, 1978.
6. Crabtree, M.R. "TRANSYT/9 Users Manual," Transport and Road Research Laboratory Application Guide 8, Crowthorne, United Kingdom, 1988.
7. Wallace, C.E., K.G. Courage and D.P. Reaves, "National Signal Timing Optimization Project-Final Evaluation Report," prepared for the Federal Highway Administration by the University of Florida Transportation Research Center, 1981.
8. Hillier, J.A. and R. Rothery, "The Synchronization of Traffic Signals for Minimum Delay," Transportation Science, 1967.
9. Wallace, C.E., "Development of a Forward Link Opportunities Model for Optimization of Traffic Signal Progression on Arterial Highways," Ph.D. Dissertation, University of Florida, 1979.
10. Wallace, C.E. and K.G. Courage, "Arterial Progression-a New Design Approach," Transportation Research Record 881, 1982.
11. Cohen, S.L. and C.C. Liu, "The Bandwidth-Constrained TRANSYT Signal-Optimization Program," Transportation Research Record 1057, 1986.
12. \_\_\_\_\_, Highway Capacity Manual, Special Report 209, Transportation Research Board, 1985, 1994, 1997, 2000.

13. Wallace, C.E. and F. White, "Development of Algorithms for Permitted Traffic Movements in TRANSYT-7F," prepared for Federal Highway Administration, University of Florida Transportation Research Center, 1986.
14. \_\_\_\_\_, A Policy on Geometric Design of Highways and Streets, 1990, American Association of State Highway and Transportation Officials, Washington, DC, 1990.
15. \_\_\_\_\_, Traffic Control Devices Handbook, Federal Highway Administration, Washington, DC, 1983.
16. Akçelik, R., "Traffic Signals: Capacity and Timing Analysis," Australian Road Research Board (ARRB) Research Report 123, Victoria, Australia, 1981.
17. Lorick, H.C., C.E. Wallace and R.E. Jarnagin, "Analysis of Fuel Consumption and Platoon Dispersion Models," University of Florida Transportation Research Center, Report No. UF-TRC-U32-TR-02, 1980.
18. Hadi, M.A. and C.E. Wallace, "A Progression-Based Optimization Model for TRANSYT-7F," Transportation Research Record 1360, Washington, DC, 1992.
19. Skabardonis, A. and A. Weinstein, "TRANSYT-7FC, TRANSYT Model for Actuated Signals," Institute for Transportation Studies, University of California, Berkeley, CA, April 1988.
20. Tarnoff, P.J. and P.S. Parsonson, "Selecting Traffic Signal Control at Individual Intersections, Volume 1," National Cooperative Highway Research Program Report 233, June 1981.
21. \_\_\_\_\_, "CPI Detailed Report," U.S. Department of Labor, Bureau of Labor Statistics, December 1996.
22. Register, R.P., "A Comparison of PASSER III and TRANSYT-7F Diamond Intersection Signal Timing," Masters Report, University of Florida, 1989.
23. Bonneson, J.A. and H.J. Son, "Prediction of Expected Red-Light-Running Frequency at Urban Intersections," paper presented on the 82nd Annual Meeting of the Transportation Research Board, Washington, D.C., January 2003.
24. Avery, D.J., "A Design Review Process for the Development of Traffic Signal Timing Plans," Masters Report, University of Florida Transportation Research Center, 1991.
25. Hadi, M.A., "Improved Strategies for Traffic Responsive Control in Arterial Signal Systems," Ph.D. Dissertation, University of Florida, 1990.
26. Skabardonis, A., "Progression Through a Series of Actuated Controllers," prepared for FHWA, DHS, Inc., October 1988.
27. Akçelik, R., "Opposed Turns at Signalized Intersections: The Australian Method," ITE Journal, Vol. 59, No. 6, pp. 21-27, June 1989.
28. Fambro, D.B., C.J. Messer and D.A. Andersen, "Estimation of Unprotected Left-Turn Capacity at Signalized Intersections," Transportation Research Record 644, 1977.
29. Machemehl, R.B. and A.M. Mechler, "Procedural Guide for Left-Turn Analysis," Center for Transportation Research, Report No. CTR 3-18-80-258-3F, University of Texas at Austin, 1983.
30. Michalopoulos, P.G., J. O'Conner and S.M. Nova, "Estimation of Left-Turn Saturation Flows," Transportation Research Record 667, 1978.

31. Nemeth, Z.A. and J.R. Mekemson, "Guidelines for Left-Turn Treatment at Signal Controlled Intersections," Ohio State University, 1984.
32. Penic, M.A. and J. Upchurch, "TRANSYT-7F: Enhancement for Fuel Consumption, Pollution Emissions, and User Costs," Transportation Research Record 1360, 1992.
33. Akçelik, R., "Time Dependent Expressions for Delay, Stop Rate and Queue Length at Traffic Signals," Australian Road Research Internal Report, ARRB Ltd., Vermont South, Victoria, 1980.
34. Showers, R.H., "Development of a Moment-Based Platooning Index," University of Florida Transportation Research Center, 1993.
35. Chen, P.J., "Hand-Held Microcomputer Applications in Data Collection for Measuring Platoon Dispersion Based on Cyclic Flow Profiles," Masters Report, University of Florida Transportation Research Center, Spring, 1993.
36. Hale, D.K., Modeling Traffic-Actuated Control with TRANSYT-7F, modified Ph.D. Dissertation, University of Florida, February 2001.
37. Hale, D.K. and K.G. Courage, "Prediction of Traffic-Actuated Phase Times on Arterial Streets", Transportation Research Record 1811, pp. 84-91, 2002.
38. Skabardonis, A., "Signal Timing Optimization in Networks with Actuated Controllers," paper presented on the 66th Annual Meeting of the Transportation Research Board, Washington, D.C., January 1987.
39. Akçelik, R., aaSIDRA User Guide, Akcelik and Associates Pty Ltd, Greythorn, Victoria, February 2002.
40. \_\_\_\_\_, Traffic Software Integrated System Volume III, CORSIM User's Guide, ITT Industries, Inc., Systems Division, Colorado Springs, CO, March 2001.
41. Chang, E.C. and C.J. Messer, PASSER II-90 Program User's Guide, Texas Transportation Institute, Texas A&M University, College Station, TX, June 1991.
42. Wallace, C.E. and K.G. Courage, PASSER II-90 User's Guide, Volume 3 of a series prepared for FHWA by COURAGE AND WALLACE, Gainesville, FL, December 1991.
43. Wallace, C.E., K.G. Courage and E.C.P. Chang, Methodology for Optimizing Signal Timing -- the M|O|S|T Reference Manual, Volume 1 of a series prepared for FHWA by COURAGE AND WALLACE, Gainesville, FL, December 1991.
44. Webster, F.V., "Traffic Signal Settings," Road Research Technical Report No. 39, London, United Kingdom, 1958.
45. Maddula, S., "Evaluation of Signalized Intersection Delay Using Video Image Detection," Masters Thesis, Department of Civil Engineering, University of Florida, August 1994.
46. Bonneson, J., "Modeling Queued Driver Behavior at Signalized Junctions," Transportation Research Record 1365, pp. 99-107, 1992.



# Index

## 9

95th Percentile Queue Length 242, 244

## A

Acknowledgements 4  
Actuated Estimation 37, 38  
Actuated Estimation Examples 381  
Actuated Estimation Methodology 199  
Actuated Screen (associated with record type 291) 135  
Adding or Deleting Intersections 53, 54  
Analysis Period 433  
Analysis Screen (associated with record types 1 1X, 10, 5X, and 9X) 104  
Analytical Model 431  
Appendix A  
    TRR Paper 415  
Arrival Type 205  
Artplan Import Methodology 206  
Associated Links 111, 116  
Attainability 206  
Automatic Flow Balancing 207  
Automatic Link Lengths 146, 147  
Average System Speed 213  
Average Travel Time 213  
Average Vehicle Spacing 111, 120

## B

Back of Queue Algorithm 213  
Bandwidth Optimization 293, 297  
Batch Processing 10, 11  
Behavior of the Enter Key 146, 147  
Bitmap Background 56, 64  
Bitmap Scaling Dialog 56, 70  
Bottlenecks 49  
Brute Force Optimization 28, 30  
Bus and Carpool Operations 48

## C

Calibration 431  
Complex Intersection Geometry 394  
Control Delay 434  
Convergence Threshold 310, 313  
CORSIM Import/Export/Processing/Optimization 14  
CORSIM Processing and Optimization  
    Examples 403  
CORSIM Processing Methodology 216  
Creating a Brand-New Input Data File 333  
Creating a New Input Data File by Importing  
    HCS Files 355  
Crossover 310, 312  
Cycle Length Optimization 299

## D

Degree of Saturation 224  
Delay and Level of Service 225  
Delete Scratch Files 317, 318  
Deterministic Model 432  
Direct CORSIM Optimization 329  
Dirty File Flag 51, 52  
Dirty Run Flag 52  
Disutility Index 230  
Double Cycling 231  
Dual-Ring Timing Screen (associated with record types 1X 18, 2Y, and 2Z) 91

## E

Editing Input Data (Sensitivity Analysis) 358  
Effective Green Time 232  
Efficiency 232  
Elitist Method 310, 313  
Empirical Model 431  
Excess Operating Cost 233  
Export Timing to CORSIM 216  
Extended Run 317, 319  
External Links 433

## F

Feeders Screen (associated with record type 28) 102  
Five Approaches or More 394  
Flow 233  
Force-Off and Maximum Green 233  
Formatted Report 17  
Fuel Consumption 234

## G

- Genetic Algorithm Optimization 310
- Genetic Algorithm Options 317
- Global Input Summary 10, 11
- Global Output Summary 10, 11
- Global Screen (associated with record types 1-10) 111
- GO-Pattern 239, 241
- Grouped Nodes 307

## H

- HCS Import Methodology 236
- HCS Import/Export 13
- Header Information Record 197
- Hill-Climb Optimization 293
- History of TRANSYT-7F 6
- Hotkeys 10, 12

## I

- Import CORSIM File 216
- Initial Timing Flag 289, 291
- Initial Timing Model 289
- IN-Pattern 239, 240
- Input Data Screen Overview 51
- Interference 238
- Internal Links 433
- Intersection Name Record 195
- Intersection Names Screen 145

## J

- Jam Spatial Headway 133, 134

## L

- Lane Configuration Screen (associated with record types 28, 29, and 291) 74
- Lane-by-Lane Analysis Example 400
- Level of Service (LOS) 225, 229
- License Agreement 1
- Link Balancing Bias 207, 212
- Link Name Record 196
- Link Numbering Scheme 71
- Link to HCS 56, 59
- Link-Wise Simulation 21, 22

## M

- Macroscopic Model 431
- Macroscopic Simulation Methodology 239
- Macroscopic Simulation of Traffic Flow 21
- Map Properties Dialog 56, 65

- Map View (associated with record types 70 and 71) 56

- Master Controller 111, 119
- Maximum Back of Queue 242
- Maximum Number of Generations 310, 313
- McTrans web site 19
- Microscopic Model 432
- Mid-Block Sources 47
- Multi-Cycle and Multi-Period Simulation 43
- Multi-Period Analysis Examples 387
- Multi-Period Optimization 326
- Mutation 310, 312

## N

- NEMA Phasing 91
- Network Partitioning 432
- New Features in Release 11 6, 9
- New TRANSYT Input File 53
- Node Input Summary 10, 11
- Node Output Summary 10, 11
- Number of Vehicles Discharged 233

## O

- Objective Function 434
- One-Touch CORSIM Animation 245
- Open with HCS 56, 59
- Operating Cost 247
- Optimal Spatial Headway 133, 134
- Optimization Node List 293
- Optimization Strategy 28
- Optimization Under Congested Conditions 306
- Original Cycle Length 104, 106
- OUT-Pattern 239, 241
- Output Quick Jump 10, 11
- Overlap Phasing 248

## P

- Pattern Initialization Cycle 239, 242
- Peak Hour Factor 433
- Pedestrians
  - Bicycles, and Exclusive Pedestrian Phases 47
- Percent of Vehicles Arriving on Green 266
- Performance Index 104, 108
- Permitted Movement Methodology 249
- Permitted Movements 40
- Permitted Screen (associated with record type 29) 136
- Permitted Shared Lanes Methodology 270, 272
- Phase Failures 432
- Phasing Sequence Optimization 319
- Platoon Dispersion 257
- Platooning Index 257, 260
- Platooning Moment 257, 260



- Population Size 310, 312
- Preferences Screen 146
- Profile View 261
- Progression Analysis Examples 362
- Progression Route Dialog 56, 68
- Proper Arteries for PROS 263, 264
- PROS 263
- PROS Optimization 304
- Punch Files 28, 36

## Q

- Queue Service Time and Green Extension Time 266
- Queue Spillback and Queue Spillover 267
- Queue Spillback and Queue Spillover Examples 371
- Queuing Capacity 267, 269
- Queuing Screen (associated with record types 28 and 291) 133

## R

- Random Number Seed 310, 314
- Recommended G.A. Parameter Values 316
- Record Type 0 156
- Record Type 1 157
- Record Type 10 165
- Record Type 18 167
- Record Type 1X 166
- Record Type 2 158
- Record Type 28 170
- Record Type 29 171
- Record Type 291 172
- Record Type 292 173
- Record Type 2Y 168
- Record Type 2Z 169
- Record Type 3 159
- Record Type 30 174
- Record Type 31 175
- Record Type 32 176
- Record Type 33 177
- Record Type 34 178
- Record Type 35 179
- Record Type 36 180
- Record Type 37 181
- Record Type 38 182
- Record Type 39 183
- Record Type 4 160
- Record Type 40 184
- Record Type 41 185
- Record Type 42 186
- Record Type 43 187
- Record Type 5 161
- Record Type 5X 188
- Record Type 6 162

- Record Type 60 189
- Record Type 61 191
- Record Type 7 163
- Record Type 70 192
- Record Type 71 193
- Record Type 8 164
- Record Type 9X 194
- Record Type Editor 12
- Record Type Format 149
- Red Light Running 432
- Reference Interval 84, 87
- References 437
- Report View 17
- Roundabout Control 46
- Route Screen (associated with record types 40-43 and 60-61) 122
- Route Title Record 190
- Run Title Record 155

## S

- Sample Data Files 19
- Satflow View Screen (associated with record types 28 and 292) 98
- Saturation Flow Rate Calculator (associated with record types 28 and 292) 142
- Scope and Qualities 5
- Sensitivity Analysis 358
- Sensitivity Parameters 269
- Shared Lanes and Lane-by-Lane Analysis 42
- Shared Lanes Methodology 270
- Sign Control 46
- Sign Control Examples 395
- Sign Control Methodology 274
- Simulation Methodology 239
- Simulation Model 431
- Sneakers 249, 255
- Spyglass Internal Data Viewer 275
- Step-Wise Simulation 21, 23
- Stochastic Model 432
- Stops 277
- Summary Output Reports 10, 11

## T

- Target Degree of Saturation Model 37
- Technical Support and Problem Reporting 20
- Throughput 279
- Time Full and Critical Link 280
- Time Jammed 280
- Time Period Quick Jump 10, 11
- Time-Space Diagram 280
- Timing Screen (associated with record types 1X 18, 2Y, and 2Z) 127

- Timing View Screen (associated with record types 1X, 18, 2Y, and 2Z) 84
- Total Operating Cost 284
- Total Travel and Total Travel Time 285
- Trademarks and Copyrights 4
- Traffic Screen (associated with record types 28 and 29) 80
- Traffic Signal Timing Elements and Characteristics 26
- Traffic Signal Timing Optimization 28
- Traffic-Actuated Control 37
- TRANSYT-7F Graphical User Interface (GUI) 10
- TRANSYT-7F Input Screens 51

## **U**

- Uncoordinated Operations 46
- Uncoordinated Operations Methodology 286
- Uncoordinated Optimization 327

## **V**

- Verification and Validation 434
- Volume-Density Control 37, 38

## **W**

- Webster's Method 299
- Weighting Screen (associated with record types 30-39) 140

## **Y**

- Yellow Trap 319, 323
- Yield Sign Control 46