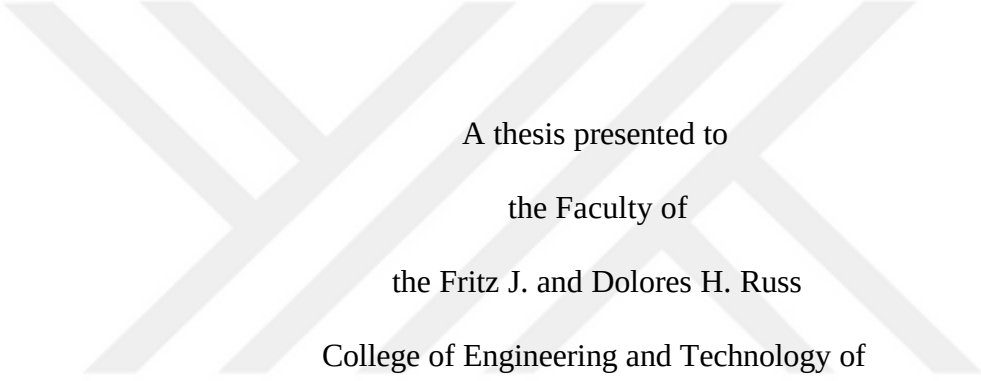


A SIMULATION APPROACH TO MODELING TRAFFIC IN CONSTRUCTION
ZONES



A thesis presented to
the Faculty of
the Fritz J. and Dolores H. Russ
College of Engineering and Technology of
Ohio University

In partial fulfillment
of the requirements for the degree
Master of Science

Erdinc Oner

November 2004

This thesis entitled
A SIMULATION APPROACH TO MODELING TRAFFIC IN CONSTRUCTION
ZONES

BY
ERDINC ONER

has been approved for
the Department of Industrial and Manufacturing Systems Engineering
and the Fritz J. and Dolores H. Russ College of Engineering and Technology by

Helmut T. Zwahlen

Russ Professor Emeritus of Department of Industrial and Manufacturing Systems
Engineering

Dennis Irwin

Dean, Fritz J. and Dolores H. Russ College of Engineering and Technology

ONER, ERDINC. M.S. November 2004. Industrial and Manufacturing Systems Engineering

A SIMULATION APPROACH TO MODELING TRAFFIC IN CONSTRUCTION ZONES (478 pp.)

Director of Thesis: Helmut T. Zwahlen

A simulation model has been designed to model the traffic when the number of lanes has been reduced due to construction work (bottleneck) using the ARENA simulation program. The ARENA simulation model consists of about 1750 modules (statements) and it uses the entity – transporter pairs to represent either a car or a truck with specific features assigned in the model. The model is developed to simulate two lane road traffic in same direction reduced to one lane through the construction zone. The model was designed to run on a PC. A minimum processor speed of 2 GHz and at least 256 MB of RAM is desirable to run the model. The ARENA Research Version 7.01 must be used for the simulation. The simulation model was run for a 24 hours time period for a weekday. The data used as input variables for the ARENA simulation model were collected during summer 2004 at the I-76 Westbound construction zone near Rootstown, Ohio as a part of the project entitled “Improved Work Zone Design Guidelines and Enhanced Model of Travel Delays in Work Zones” for Ohio Department of Transportation. This data was analyzed in order to be used as input and validation for the simulation model. The ARENA model could not be fully evaluated since the actual traffic volumes which were observed and collected were not high enough to cause queues in the lane reduction area (lane reduction from two lanes to one lane). For the relatively low traffic volume investigated in the real world the ARENA model, as well as the QuickZone model did not indicate any queues and delays.

Approved:

Helmut T. Zwahlen

Russ Professor Emeritus of Department of Industrial and Manufacturing Systems
Engineering

ACKNOWLEDGMENTS

First of all I would like to thank my advisor Dr. Helmut T. Zwaalen for his guidance and encouragement throughout my graduate study. His expert knowledge and advice guided me through this thesis, without which I would not have been able to get this point.

I would also like to thank to my thesis committee members Drs. Dušan N. Šormaz and Lloyd A. Herman for their suggestions and help.

Many thanks to colleagues Royston Lobo, Sripathi Maddisetty, and Prashanth Bejai for their assistance and help.

Special thanks to Sahika Vatan Korkmaz for being my best friend and for motivating and supporting me throughout my graduate study.

Lastly, I would like to express my love and gratitude to my parents for their never ending support and love. To my parents, Sükriye and Enver Öner, I dedicate this thesis.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	xi
1 INTRODUCTION	1
1.1 Statement of the Problem.....	3
1.2 Objective of the Study	3
1.3 Scope of Work	4
2 LITERATURE REVIEW	5
2.1 Work Zone Simulation Programs Evaluation Studies	5
2.2 Work Zone Simulation Program Results Evaluation Studies	12
3 METHODOLOGY	16
3.1 I-76 Westbound near Rootstown Construction Site.....	16
3.1.1 Description of the Work Zone Site Used in the Example.....	16
3.1.2 Data Collection	17
3.1.3 Data Analysis	20
3.1.3.1 Converting Hourly Traffic Counts to Cumulative Inter-arrival Time Distributions.....	26
3.2 Description and Design of ARENA (SIMAN) Simulation Program.....	51
3.2.1 Description of ARENA (SIMAN) Simulation Program.....	51
3.2.2 Design of the ARENA Traffic Simulation Model	52
3.2.2.1 Input Variables	52
3.2.2.1.1 Construction Zone Configuration	52
3.2.2.1.2 Inter-arrival Time Distribution (Entity Arrival)	54
3.2.2.1.3 Vehicle Types	57
3.2.2.1.4 Speed Profile	59
3.2.2.1.5 Car Following Behavior.....	62
3.2.2.1.6 Gap Acceptance for Merging and Lane Changing Behavior .	62

	vi
3.2.2.2 Flowchart of the ARENA Simulation Model	75
3.2.2.3 Output Variables	90
3.2.2.4 Limitations	90
3.3 Description and Design of QuickZone Delay Estimation Program.....	91
3.3.1 Inputs of QuickZone Delay Estimation Program.....	92
3.3.2 Outputs of QuickZone Delay Estimation Program.....	97
4 Results and discussion of results.....	100
4.1 Analysis and Discussion of ARENA Simulation Results	100
4.2 Comparison of ARENA with QuickZone	115
4.3 Discussion of Results	116
5 Conclusions	118
6 References	120
Appendix A. Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime Driving/Passing Lane	125
Appendix B. IATs as a Function of Volume (vphpl) on Driving Lane and Passing Lane during Daytime for Cumulative Percentage Values 1%, 2%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 95%, 98%, 99%, and 100% (Maximum)	141
Appendix C. Calculated Inter-arrival Times for 15 Minute Intervals for 24-hout Time Period	158
Appendix D. ARENA Simulation Model SIMAN Code.....	169

LIST OF FIGURES

Figure 1: Map of the Work Zone Site used in the Example	16
Figure 2: Trailer Locations Used in the I-76 Westbound Construction Zone Data Collection.....	18
Figure 3: Configuration of radar trailer and ORITE equipment for measuring traffic used in evaluation (adapted from Zwahlen et al. [22]).....	19
Figure 4: Comparison of Average Inter-arrival Times versus Number of Vehicles per Hour for Daytime and Nighttime Driving Lane Data	21
Figure 5: Comparison of Average Inter-arrival Times versus Number of Vehicles per Hour for Daytime and Nighttime Passing Lane Data	22
Figure 6: An Actual Cumulative IAT Distribution Observed during a 15 Minute Time Period (08/20/04 Friday, 8:15-8:30 AM)	27
Figure 7: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 469 vehicles/hour/driving lane	44
Figure 8: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 581 vehicles/hour/driving lane	45
Figure 9: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 698 vehicles/hour/driving lane	46
Figure 10: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 152 vehicles/hour/passing lane	47
Figure 11: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution	

(c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 419 vehicles/hour/passing lane	48
Figure 12: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 518 vehicles/hour/passing lane	49
Figure 13: Work Zone Configuration used in the Example Simulation.....	53
Figure 14: Hourly Traffic Counts for Driving Lane for 3 Days of Data	55
Figure 15: Hourly Traffic Counts for Passing Lane for 3 Days of Data.....	56
Figure 16: Comparison of Actual Speed Data for Driving Lane Collected on 08/20/2004 Friday with the Normal Distribution.....	60
Figure 17: Comparison of Actual Speed Data for Passing Lane Collected on 08/20/2004 Friday with the Normal Distribution.....	60
Figure 18: Gap Acceptance.....	63
Figure 19: MSSLC for the space between the leading vehicle and the merging vehicle versus relative speed between the lanes (adapted from Kanaris et al. [33])	64
Figure 20: MSSLC for the space between the following vehicle and the merging vehicle versus relative speed between the lanes (adapted from Kanaris et al. [33])	65
Figure 21: Comparison of the Van Aerde Car Following Model with the Field Data (adapted from Rakha and Crowther [34]).....	67
Figure 22: Comparison of Greenshields, Pipes, and Van Aerde Car Following Models (adapted from Rakha and Crowther [34]).....	68
Figure 23: Flowchart of the ARENA Traffic Simulation Model.....	79 to 89
Figure 24: Node Information (X-Horizontal Axis, Y-Vertical Axis)	92
Figure 25: Link Information	93
Figure 26: Temporal Distribution of Hourly Inbound Demand on I-76 Westbound based on the Hourly Vehicle Count Data for both Lanes Collected on 08/20/04.....	95
Figure 27: Daily Traffic Counts for the Links	96
Figure 28: Delay Graph for the Project (Whole Week).....	99

Figure 29: Comparison of Actual Number of Vehicles on Both Lanes and Simulated Number of Vehicles on Both Lanes as a Function of Time.....	101
Figure 30: Comparison of Actual Number of Vehicles on Driving Lane and Simulated Number of Vehicles on Driving Lane as a Function of Time.....	101
Figure 31: Comparison of Actual Number of Cars on Driving Lane and Simulated Number of Cars on Driving Lane as a Function of Time	102
Figure 32: Comparison of Actual Number of Trucks on Driving Lane and Simulated Number of Trucks on Driving Lane as a Function of Time	102
Figure 33: Comparison of Actual Number of Vehicles on Passing Lane and Simulated Number of Vehicles on Passing Lane as a Function of Time	103
Figure 34: Comparison of Actual Number of Cars on Passing Lane and Simulated Number of Cars on Passing Lane as a Function of Time	103
Figure 35: Comparison of Actual Number of Trucks on Passing Lane and Simulated Number of Trucks on Passing Lane as a Function of Time for	104
Figure 36: Observed Speed Flow Relationship on a San Diego Freeway [23]	106
Figure 37: Comparison of Average Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Interval for Driving Lane	107
Figure 38: Comparison of Maximum Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Driving Lane	108
Figure 39: Comparison of Minimum Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Driving Lane	108
Figure 40: Comparison of Average Wait Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Driving Lane	109
Figure 41: Comparison of Number of Vehicles Waiting in the Queue per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Driving Lane	109
Figure 42: Comparison of Average Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane	111
Figure 43: Comparison of Maximum Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane	111

Figure 44: Comparison of Minimum Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane	112
Figure 45: Comparison of Average Waiting Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane	112
Figure 46: Comparison of Maximum Waiting Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane	113
Figure 47: Comparison of Minimum Waiting Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane	113
Figure 48: Comparison of Average Number of Vehicle Waiting in Queue per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane.....	114
Figure 49: Comparison of Maximum Number of Vehicle Waiting in Queue per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane	114
Figure 50: Comparison of Minimum Number of Vehicle Waiting in Queue per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane.....	115

LIST OF TABLES

Table 1: Misses and Phantoms Observed on I-76 near Rootstown Construction Zone, Ohio (08/20/04 Friday – 10:52AM-11:32AM)	24
Table 2: Length Classification of Vehicles by Trailer Compared to Video Record for August 20, 2004	25
Table 3: Hyperbolic Fit Formulae used in Excel Spreadsheet for Determining Cumulative IATs for Selected Percentiles for I-76 Westbound Driving Lane (X is hourly traffic count in vphpl and Y is cumulative IAT in seconds.).....	29
Table 4: Hyperbolic Fit Formulae used in Excel Spreadsheet for Determining Cumulative IATs for Selected Percentiles for I-76 Westbound Passing Lane (X is hourly traffic count in vphpl and Y is cumulative IAT in seconds.).....	30
Table 5: IATs in Seconds based on Hyperbolic Formulae Developed for Driving Lane (Min 184 vehicles/hour, Max 782 vehicles/hour)	31
Table 6: IATs in Seconds based on Hyperbolic Formulae Developed for Passing Lane (Min 66 vehicles/hour, Max 666 vehicles/hour)	32
Table 7: Comparison of Average IATs Generated using IAT Distribution with the Actual Average IATs	34
Table 8: Corrected IATs in Seconds based on Hyperbolic Formulae Developed for Driving Lane (Min 184 vehicles/hour, Max 782 vehicles/hour)	35
Table 9: Corrected IATs in Seconds based on Hyperbolic Formulae Developed for Passing Lane (Min 66 vehicles/hour, Max 666 vehicles/hour).....	36
Table 10: Cumulative IATs Calculated using the OU Fitting Distribution.....	39
Table 11: Cumulative Percentage Values used for Negative Exponential Distribution in OU Fitting Distribution Comparison Graph (Figure 9 a, b)	40
Table 12: Cumulative Percentage Values used for Normal Distribution in OU Fitting Distribution Comparison Graph (Figure 9 c, d)	41
Table 13: Cumulative Percentage Values used for Pearson Type III Distribution in OU Fitting Distribution Comparison Graph (Figure 9 e, f)	43
Table 14: Percentages of Passenger Cars and Trucks for Driving Lane and Passing Lane according to OU Video Record.....	58

Table 15: Vehicle Speeds used for the Transporters in the ARENA Simulation Model..	61
Table 16: Minimum Safe Space for Lane Changing before the Lane Closure Taper (adapted from Kanaris et al. [32])	66
Table 17: Required Gaps for Lane Changing Maneuver Derived from Van Aerde Car Following Model (adapted from Rakha and Crowther [33])	69
Table 18: Required Space for Lane Changing when the Merging Vehicle Speed is Greater or equal than the Desired Lane Speed.....	71
Table 19: Required Space for Lane Changing when the Merging Vehicle Speed is less than the Desired Lane Speed.....	72
Table 20: Required Space for Lane Changing when there are Less than 3 Vehicles waiting in the Queue at the Lane Closure Taper.....	73
Table 21: Required Space for Lane Changing when there are Less than 5 Vehicles waiting in the Queue at the Lane Closure Taper.....	74
Table 22: Required Space for Lane Changing when there are Less than 10 Vehicles waiting in the Queue at the Lane Closure Taper.....	74
Table 23: Required Space for Lane Changing when there are More than 10 Vehicles waiting in the Queue at the Lane Closure Taper.....	75
Table 24: Hourly Vehicle Counts and Hourly Demand Factors for 08/20/2004	94
Table 25: Output Summary Table	97
Table 26: Comparison of Number of Vehicles observed in the Field during Data Collection and the Number of Vehicles Obtained from Simulation Output.....	100

1 INTRODUCTION

A simulation study is concerned with the building of a model for a problem rather than directly working with the problem itself. Successful simulation study needs comprehensive and multidisciplinary knowledge and experience [1]. The main purpose of a traffic simulation model in construction zones is to estimate the delay times for the drivers and determine bottlenecks, which cause delays for the drivers. Reduced guidance, dense traffic, merges at lane reductions and entrance ramps are the main causes of traffic delays in construction work zones.

Traffic simulation models are used to replicate the operations of actual traffic systems. These simulation models are very powerful tools to predict the characteristics of traffic flow in different conditions. Accurately modeling the traffic in work zones will prevent the bottlenecks in the work zones, and it will reduce the delay times.

Traffic simulation models are divided into two main categories: microscopic simulation models and macroscopic simulation models [2]. Microscopic models simulate the traffic flow by using the behavior of the individual vehicles or characteristics of the drivers. Microscopic models help us to predict the actual traffic flow better than the macroscopic models. In microscopic models, car following behavior, lane-changing behavior, acceleration and deceleration behaviors are included in the model individually. On the other hand, macroscopic traffic simulation models regard traffic flow as a continuum or as a stream of fluid.

Traffic modeling is basically a queueing system. Queueing situation always will occur when a service facility is not capable to serve all the arrivals at some point in time. In traffic simulation, vehicles entering the system are the entities arriving and the road

section is the service facility, which has a service distribution related with the speeds of the incoming vehicles. The inter-arrival time probability density function can be determined by measuring the exact times that the vehicles enter the work zone. The service time probability density function can be determined using the speed profile of the vehicles in the work zone and delays associated with the merger and constricted lanes.

There are basically two types of queueing models; deterministic and probabilistic models [3]. In deterministic models there are no probability distributions associated with the arrival of events or service times. The arrivals of events with the constant rate are known. However, in stochastic models a probability distribution is associated with the arrival of events and the service times. In traffic and in most queueing situations the entities arrive to the system with a probabilistic inter-arrival time (IAT) distribution and the service times (ST) also have a probabilistic distribution varying for each entity. Furthermore the output variables of the queueing systems, length of the queue and the delay time for the entities are also probabilistic measures. The behavior of queues, number of entities waiting in the queue, and delay times cannot be solved with a straight forward mathematical approach unless IAT and ST are very well behaved mathematical distributions (e.g. Negative Exponential, Erlang, or Hyper-Poisson).

Monte Carlo Simulation, run for many thousand times, can provide reliable answers, which are very specific and unique, for determining system countermeasures. Other methods, like deterministic simulations where arrivals and service times occur at periodic time intervals cannot accurately capture the behavior of such queueing systems.

1.1 Statement of the Problem

Construction work zones on heavily traveled highways can cause problems for the motorists. Construction work zones may cause travel time delays, bottlenecks, and even accidents. Slowing and merging traffic is the main cause of the problems in construction work zones. Better planning of construction work zones before the starting of construction projects would decrease the problems associated with the work zone. In most of the construction work zones with high hourly traffic volumes, paved shoulders or medians are used to avoid the reduction of lanes. If the queue lengths and the delay times for the traffic in the construction work zone would be known in advance, then, the decision of whether or not a paved shoulder or median should be utilized for the traffic could be made with more accuracy. If no excessive queues and waiting times would be forecasted the cost of providing the extra temporary paved lanes could be avoided and a large amount of costs could be saved. A simulation approach could forecast the lengths of queues and the extent of delay times and alternative construction work zone solutions could be explored without affecting the motoring public. Simulation has the potential to help identify the outcomes of various configurations for construction work zone projects without actually affecting the motoring public.

1.2 Objective of the Study

The objective of this study is to develop a traffic simulation model using ARENA simulation program to estimate the expected delays and queues using a probabilistic inter-arrival time distribution in construction zones and to investigate the capabilities and

limitations of the ARENA simulation program for modeling traffic when the number of lanes are reduced.

Another objective is to use a procedure to measure the traffic in a real world situation and to process the time stamped traffic data in such a way that inter-arrival time distributions can be used in the ARENA simulation model.

Furthermore, we also want to compare the results of the ARENA traffic simulation model with the QuickZone Delay Estimation Program, which uses a deterministic queueing modeling approach.

1.3 Scope of Work

The first step of this study is to find out the available construction work zone simulation programs available in the public domain and literature and find out what the advantages and disadvantages of these simulation programs are.

The second step of this study is to conduct a literature review to find out traffic behavior through the construction work zones.

The third step of this study is to develop a simulation model for the construction work zone using the ARENA 7.01 simulation program by Rockwell Software.

The fourth step is to analyze measured traffic in a real world environment and obtain probabilistic inter-arrival time distributions for different traffic volumes during a day in order to validate the simulation program.

The fifth step is to compare the ARENA simulation program output with the QuickZone delay estimation program output.

2 LITERATURE REVIEW

2.1 Work Zone Simulation Programs Evaluation Studies

QuickZone work zone delay estimation program, which was developed by The Federal Highway Administration (FHWA) in cooperation with Mitretek Systems [4], uses a deterministic inter-arrival time distribution for the estimation of the output parameters. It uses a deterministic queueing model to simulate traffic going through work zones to determine when there would be traffic backups. It is a tool intended for highway engineers to determine how to set up a work zone to minimize traffic disruption. QuickZone provides four primary outputs—a delay graph, a travel behavior summary, an amortized delay and construction cost graph, and a summary table. Detailed description of the QuickZone delay estimation program is given in section 3.3.

In a study by Maze and Kamyab [2], a work zone simulation model was developed using ARENA simulation program. The model developed to provide the delay estimation when a lane closure occurs and to visually demonstrate the forecasted delay. Lane changing and car following algorithms were included in the simulation model. The authors investigated the effects of the slow moving vehicles and late mergers using the developed simulation model. They compared the results of the developed simulation model, traffic flow rate, speed, and delay time with the actual data. They found that the outputs of the model establish a level of confidence that the model is capable of simulating the conditions of the work zone. The results of their study showed that with traffic volumes less than 700 vehicles per hour, it is unlikely to observe delay at the lane closures. At traffic volumes of 950 vehicles per hour, it is more likely to observe delay in

the construction work zones. The authors were contacted in order to get more information about the details of the ARENA simulation model, but no information was obtained about the architecture of the model.

Al-Kaisy, Stewart and Van Aerde [5] performed a simulation study to examine the capacity and the operational performance at freeway diverge areas. In order to understand the traffic behavior at freeway diverge areas, they used microscopic traffic simulation model INTEGRATION. Car following behavior and lane changing behavior are the most important features of INTEGRATION for simulating freeway operations. Using the user specified free flow speed, speed at capacity, capacity and jam density single regime speed flow density relationship is determined in the model. Another important feature of the program is that it includes both mandatory and discretionary lane changing behavior. Mandatory lane changing occurs when a driver must change the lane because of the ending lane or closed lane, and discretionary lane changing occurs when the other lane provides better driving conditions. Al-Kaisy et al. developed their model at an exit ramp. They analyzed different deceleration lane lengths at exit ramps and their effects on capacity and operational performance at freeway diverge areas. .

Benekohal and Abu-Lebdeh [6] performed a variability analysis, using the stochastic traffic simulation model TRAF-NETSIM outputs. In the study the authors used batch means and replications to assess variability in the measure of effectiveness calculations of NETSIM. They used average delay, average speed, and vehicle trips as the measures of effectiveness parameters. The outputs of the simulation model are analyzed by the statistical methods such as batch means method, replication method, and correlation among batches method. The batch means method is performed by running the

program for one long replication and dividing it into smaller batches. Statistics are collected and using the variability within the batches, a confidence interval is build for the model. Another way of building confidence interval is replication method. In this method multiple independent replications having same roadway and traffic conditions are run for the model and statistics on the system performance are collected. Statistics for the batches may be treated as stationary time series data, when a long run is divided into batches. Plot of the mean value for each batch against time will help to determine whether the time series is stationary. The authors proposed an interval calculation method and compared the results of the simulation model with the results they calculated using the proposed interval calculation method. Their study showed that with the proposed interval calculation method they could build confidence intervals for the measure of effectiveness values.

Bloomberg and Dale [7] compared microscopic simulation models VISSIM and CORSIM by using them in a study for designing alternatives in Seattle, Washington. CORSIM was developed by FHWA to analyze freeways, urban streets, and corridors or networks. VISSIM was developed at the University of Karlsruhe, Germany to analyze functionally classified roadways and public transportation operations. In order to compare these two simulation models, the authors applied the models to analyze the alternatives for the project performed by Washington State Department of Transportation. Six different scenarios were analyzed using the models. The authors find differences in the car following logic, network-coding process, gap acceptance, modeling of signals, animation features and output data of the models. The network coding process is different between the two models. CORSIM uses a link-node structure wherein the user defines the

attributes such as speed, lane configurations, and traffic control devices to the links and nodes. In VISSIM use of nodes is eliminated, the model relies on links and connectors which allow the user to match the network geometry to field conditions. The car following model in CORSIM sets a desired amount of headway distance between the vehicles. Vehicles in the model seek to maintain the minimum allowed headway distance while they are not exceeding the maximum allowed speed. Speed, acceleration, and status of each vehicle are recomputed in every second by CORSIM. VISSIM like CORSIM uses an interval based simulation approach. VISSIM simulates traffic flow by moving driver-vehicle units through a network. Stochastic distributions are used to replicate individual driver vehicle unit behavior and dynamic headway. Gap acceptance feature in CORSIM is adapted using 10 different driver types. Variable gap acceptance is assigned to each driver considering the current available gap and a personal gap acceptance value. Bloomberg and Dale concluded that both simulation models are acceptable for modeling traffic, but they recommended that the modelers should use more than one simulation model to make more accurate recommendations.

Makigami and Nakanishi [8] developed a macroscopic simulation model to investigate the traffic flow in the bottleneck sections of an expressway during peak period. They collected the traffic flow data by video recording, aerial photography, and measuring travel times and running speeds. Estimation of capacity of bottleneck sections and measurement of spot running speed is calculated using video recording method. Aerial photography method gave the measurement of traffic density and pursuit of behavior of congested area. Using these traffic data the capacity of the highway section is calculated. Mathematical simulation model is developed using the capacity

information. The results of the developed simulation model are compared to the actual data collected. The results of the simulation model did not show significant difference from the actual traffic data collected.

Memmott and Dudek [9] developed a model entitled Queue and User Evaluation of Work Zones (QUEWZ). The purpose of the model is to determine the effects of different lane closure strategies in work zones. Memmott and Dudek found limitations in several of the methods previously developed by traffic engineers and software engineers to measure the costs associated with work zone delays. Most of the simulation models used average daily traffic volume for simulation, but Memmott and Dudek used hourly traffic volumes in their model. The traffic pattern can have a large effect on the speeds and queues throughout the day and using average daily volume might cause misleading results. Usually, the traffic will not arrive at a work zone in a uniform pattern and therefore, the average daily traffic value can misrepresent the actual arrival rates for different hours of the day. This effect is evident during rush hour when typically more delays occur than at times associated with less traffic volume. Delay or travel time costs, vehicle running costs, speed-change cycling costs, and accident costs resulting from restricted capacity through a work zone can be determined using the model developed by Memmott and Dudek.

Morales and Paniati [10] conducted a simulation study to analyze the effectiveness of traffic simulation model, ROADSIM, at two-lane roads. Roadsim is a reprogrammed version of an earlier developed model TWOWAF. The model recalculates the position of the vehicles in 1-second intervals considering the effects of the roadway geometry, traffic control, driver preferences, vehicle type and performance

characteristics, and passing opportunities based on the oncoming traffic. TWOWAF logic was modified to include the car following logic and vehicle generation logic which emits vehicles onto the simulated roadway at each end. This new model is called Roadsim. Morales and Paniati compared the results of the simulation model Roadsim, measures of effectiveness values, with the actual data collected from the two-lane rural road in Virginia. The geometric conditions (grade of the road) and traffic conditions (mean speed, percentage of the trucks) are studied and it is found that ROADSIM generates appropriate results for simulating the traffic flow at two lane rural roads when it is compared to the actual traffic flow data.

Nakanishi et al. [11] developed a macroscopic traffic simulation system, which also includes the microscopic simulation features. The simulation model MITRAM (Road Traffic Simulation System with the Microscopic Model for Analyzing Traffic Jam in the Broad Areas) is used for the congested real time road traffic simulation. The microscopic features of the model like decision-making process of the motorists are modeled according to the fuzzy theory. They developed a fuzzy model for simulating the behavior of the vehicle, which reflects the microscopic features of the traffic in the simulation.

Robles and Janson [12] implemented a dynamic traffic simulation model (DYMOD) to the I-25/HOV corridor at southeast of Denver to predict the traffic conditions during incidents. Robles and Janson used loop detectors to collect the actual traffic data. Using the collected data and developed configuration they simulated the traffic flow at I-25/HOV corridor. Using the DYMOD they simulated lane-blocking accidents and estimated the accident delays. The study showed that the simulation of

incidents provide an advantage to plan alternative routes for accidents and reduce delay times. In addition, the authors concluded that DYMODO could be used for planning traffic during construction projects.

Schonfeld and Chien [13] developed a model to find the optimal work zone lengths for two-lane highways. They explain that highway maintenance is very expensive and the delay costs of users can actually exceed the maintenance expenditures by highway agencies. If flow exceeds the capacity of the open lanes, then queues start forming, which increases travel times and resulting user costs. Schonfeld and Chien also note that the increase in average travel time is also proportional to the length of the work zone. As a general rule, longer work zones cause longer user delays. The study also indicates that agencies have tried to develop guidelines for sizing zones but better guidelines must be created to effectively integrate agency and user goals.

Benz, Fenno, and Voigt [14] analyzed the advantages of traffic modeling in reconstruction projects. They analyzed the I-45 Pierce Elevated reconstruction in Houston. At the start of construction project the alternatives are analyzed to finish the project at minimum time. They used macroscopic simulation model FREFLO, a component of the CORFLO simulation model to analyze the construction zone. The alternatives are analyzed for weekdays and weekends and the results are used to determine the liquidated damage costs to be placed on the contractor. During the construction project the data are collected for evaluating and identifying the bottlenecks in the traffic. The collected data showed that there was not much traffic delays and increase on the travel time of the customers. The identification of the routes and phases in

the construction project before the construction started helped to prevent bottlenecks and delays.

2.2 Work Zone Simulation Program Results Evaluation Studies

Rao and Owen [15] proposed a multistage validation procedure for the high-fidelity traffic simulation models. The multistage validation procedure is consisted of two approaches; conceptual validation and operational validation. In conceptual validation, model survey and model walkthrough methods are applied to the model. The operational validation is divided into two stages. The first one is qualitative approach, in which performance measures are compared graphically and by animation. The second approach is the quantitative approach. The first test in quantitative approach is comparing two means; the second test is the non-parametric approach in which Kolmogorov-Smirnov test, two dimensional two sample test, and one sample t-test are applied; the third test is the parametric approach, in which error analysis using autoregressive-integrated-moving-average (ARIMA) is performed. The proposed analysis approach provides higher confidence level in the validation of simulation models.

Roupail and Tiwari [16] state that estimating capacity and level of service (LOS) at freeway construction zones is essential for the planning and scheduling of work zone traffic control. They conclude that traffic speed through a lane closure depends on the following features. Geometric features such as lane configuration, grades, curves, lane width, lateral clearance, sight distance, and proximity to ramps. Traffic features such as flow rates and (heavy) truck occurrence. Traffic control features such as signing, arrow

boards, and flaggermen. Work activity features such as location, crew size, equipment type noise, dust, and length of work zone.

Carr [17] presents a model for predicting the Construction Congestion Cost (CO^3) associated with work zones. He states that three types of delays can be present at work zones. The first one is the speed delays, which is the difference in time to travel through the work zone before and after the construction. The second one is the backup delays, which is the time that vehicles must wait to enter the work zone because of the reduced capacity within the area. The work zone delay is the sum of these two delay types. And the last one is the diversion delay, which is the difference in time to travel another path around the work zone. The presented model by Carr calculates these different kinds of delays, which can be observed at work zones.

Carter, Rakha, and Van Aerde [18] analyzed the differences in traffic flow measures between freeway lanes. The main cause of the variability between the lanes is the requirement for passing to the left lane for faster vehicles and slower vehicles stay on the right lane. Another source of this variability is the presence of trucks. And the third factor causing the variability in the traffic flow between the lanes is the presence of entrance and exit ramps. The vehicles have to decelerate on exit ramps and the vehicles should be in the shoulder lane to exit. In case of the entrance ramps, the vehicles have to accelerate to reach the posted speed limit on the freeway and they should be in the shoulder. Carter, Rakha, and Van Aerde analyzed the traffic flow at Queen Elizabeth Way in Ontario. The authors suggested that different lane flows have different impacts on the microscopic simulation models. The speed shows variability according to the lanes

and they also found that the day of the week does not have a significant effect on the traffic flow measures variability.

Chronopoulos and Wang [19] developed a traffic simulation model through parallel processing. Chronopoulos and Wang applied the Lax method (explicit) and the Euler method (implicit) to simulate the traffic data collected from Minnesota. Entrance and exit ramp sections of the freeway are also simulated. The results of the methods used in the simulation were found acceptable when they are compared to the actual data collected from the experiment site. They concluded that the Lax and Euler methods could be used in the solution of a traffic flow continuum model.

Dudek et al. [20] completed capacity studies at nine work zones on Texas and Oklahoma freeways. They concluded that individual work zones have characteristics such as the grade, automobile mix, presence of entrance and exit ramps, and number of lane closures that affect the overall traffic flow. They also concluded that it is important to estimate the impact of a lane closure and take appropriate measures to minimize traffic delays.

Lemessi [21] developed a car following and lane changing micro-simulation model of a two-lane road section using SLX (Simulation Language with Extensibility). In car following algorithm, the author defines a desired speed for each driver, when the distance between a vehicle and its leader is greater than a pre-defined driver-specific critical distance, the driver tries to maintain its desired speed. When the distance between two consecutive vehicles falls below the critical distance the following vehicle either changes the lane or brakes. In the following mode the driver follows the leading vehicle, trying to maintain zero speed difference between the leading vehicle and the following

vehicle. In the developed model, lane changing mode occurs in three cases; during free driving mode, vehicle perceives a slower vehicle and changes lane without braking; during the braking mode, the driver stops braking and changes lane; during the following mode, driver changes the lane. In modeling lane changing behavior of the vehicles, minimum required headway distance is assumed as 50 meters.



3 METHODOLOGY

3.1 I-76 Westbound near Rootstown Construction Site

3.1.1 Description of the Work Zone Site Used in the Example

The construction work zone chosen to simulate was Ohio State Job Number 534.03, also identified as POR-76-9.50 on construction drawings. It was a bridge repair and pavement resurfacing job on I-76 in Rootstown and Edinburg Townships in Portage County. It extends from State Route 14 on the east to State Route 44 on the west. The map of the construction work zone site is given in Figure 1. In the summer of 2004, both lanes of eastbound traffic are crossed over to the westbound direction and the westbound traffic is reduced to a single lane for about 1.1 miles. The speed limit on I-76 was 65 miles per hour (MPH) before the construction zone, and it was 55 MPH through the construction zone.



Figure 1: Map of the Work Zone Site used in the Example

3.1.2 Data Collection

Data used in the simulation study were collected as part of the “Improved Work Zone Design Guidelines and Enhanced Model of Travel Delays in Work Zones” project for Ohio Department of Transportation. The data were collected at I-76 construction work zone in the westbound direction near Rootstown using microwave radar detectors. These detectors use a microwave radar beam as the means of detection that is reflected by passing traffic. Total of nine trailers equipped with microwave radar units were deployed at the construction zone. The locations of the trailers are shown in Figure 2.

Microwave radar equipped trailers collect time-stamped individual vehicle data including exact time of passing, speed, vehicle length, and classification. The data used in the example was collected between the dates 08/20/2004 (Friday) and 08/22/2004 (Sunday). Three days of data was collected in order to reduce the affects of day to day variability in the traffic flow behavior. In addition to the trailer records, traffic was recorded on video at all trailer locations for 30 minutes in order to compare and validate trailer records.

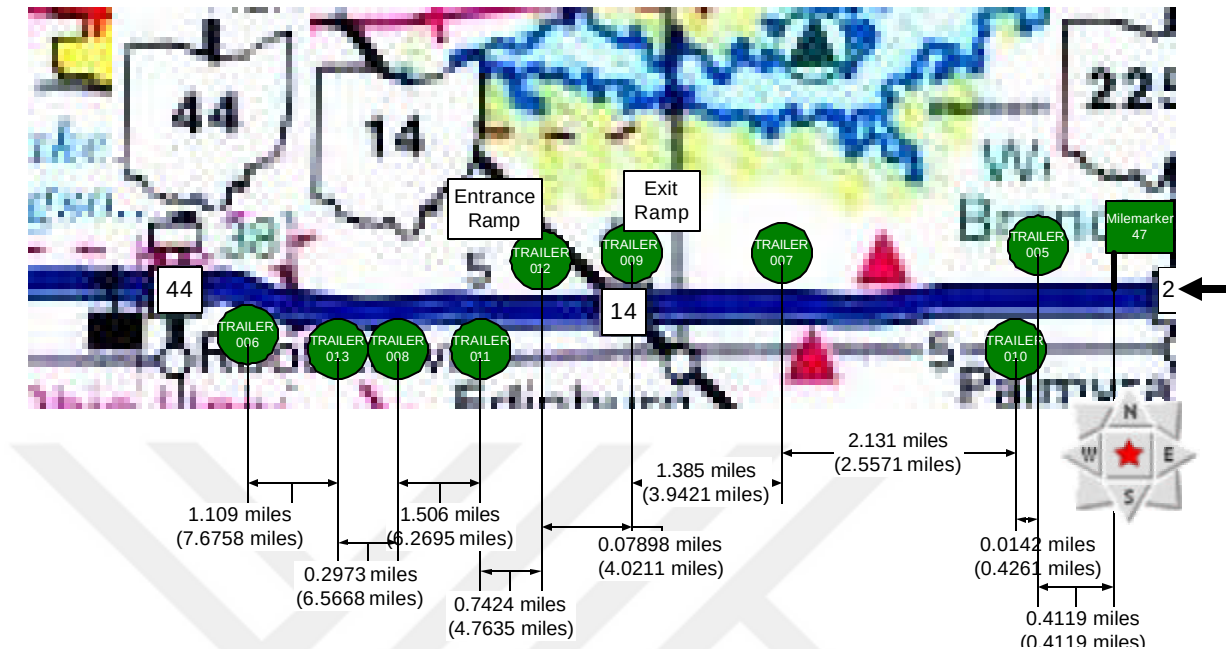


Figure 2: Trailer Locations Used in the I-76 Westbound Construction Zone Data Collection

In the example, the trailer data collected from the fifth location was used. This location was about 1.5 miles before the start of the lane closure taper, which was reflecting the free flow characteristics of the traffic. The trailer data from location 5 was used for generating the cumulative probability distributions for the inter-arrival times, identifying the percentages of the vehicle types, and assigning the initial speeds for the vehicles. The trailer at location 6 was used for adjusting the speeds of the vehicles in the work zone.

The data collected by the microwave radar trailers include time to the nearest millisecond, lane of traffic, and a set of data from the first sensor: a timestamp in 2.5-ms time increments, duration of the radar image in 2.477-ms time increments, a moving average speed based on the last 16 vehicles in mph, and a vehicle class [22]. Next the

same set of data for the second sensor, then an average of the two running average speeds in mph, vehicle length in feet, and per vehicle speeds for each sensor in mph. All speeds are rounded to whole mph values, and all lengths to even feet values.

The trailer measurements were validated by measuring traffic separately for approximately half an hour at each trailer location, as shown in Figure 3. The traffic was videotaped with a time-stamped video. Later the videotape was analyzed and vehicles in each lane were correlated with data records downloaded from the trailer.

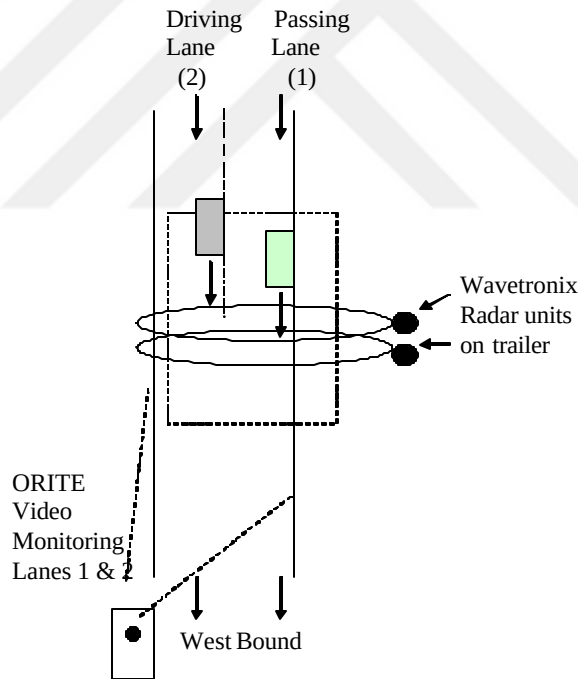


Figure 3: Configuration of radar trailer and ORITE equipment for measuring traffic used in evaluation (adapted from Zwahlen et al. [22])

3.1.3 Data Analysis

The first step of the data analysis was separating daytime and nighttime traffic count data for the work zone in order to identify the difference in traffic volume during daytime and nighttime. Time from sunrise until sundown was assumed as daytime. The work zone data was collected between the dates 08/20/2004 and 08/22/2004 and the daytime was identified as the times between 6:45 AM to 8:15 PM. Three days of data were used in the study. The summary of the three days of data for driving lane and passing lane for nighttime and daytime durations are given in Appendix A. Data collected by the trailer was divided into 15 minute time intervals and analysis was performed with the 15 minute interval data sets. For each 15 minute interval, vehicle counts, average speeds, and average inter-arrival times were computed. Microwave radar sensor equipped trailers provide an exact time stamp when a vehicle passes by the trailer. The difference between two consecutive time stamps gives the inter-arrival time for those two vehicles. After the counts were calculated for 15 minutes, each of the 15 minute counts were multiplied by 4 in order to get the hourly vehicle counts.

The analysis of daytime and nighttime IAT distributions showed that there was not much difference in the IAT distributions between day and night. For similar hourly traffic counts, almost the same average inter-arrival times were observed for daytime and nighttime traffic data. In Figure 4, the traffic volume versus average inter-arrival time comparison for daytime and nighttime are both given for driving lane for 3 days of data. Figure 5 shows the comparison for passing lane. The difference in the hourly traffic counts can be easily observed in the figures as it was expected. The figure also shows that

the average inter-arrival time for a given time interval is dependent on the number of vehicles per time interval. Average inter-arrival time for daytime driving lane traffic was 6.28 seconds with a standard deviation of 2.31 seconds. Average inter-arrival time for nighttime driving lane was 17.98 seconds with a standard deviation of 9.94 seconds.

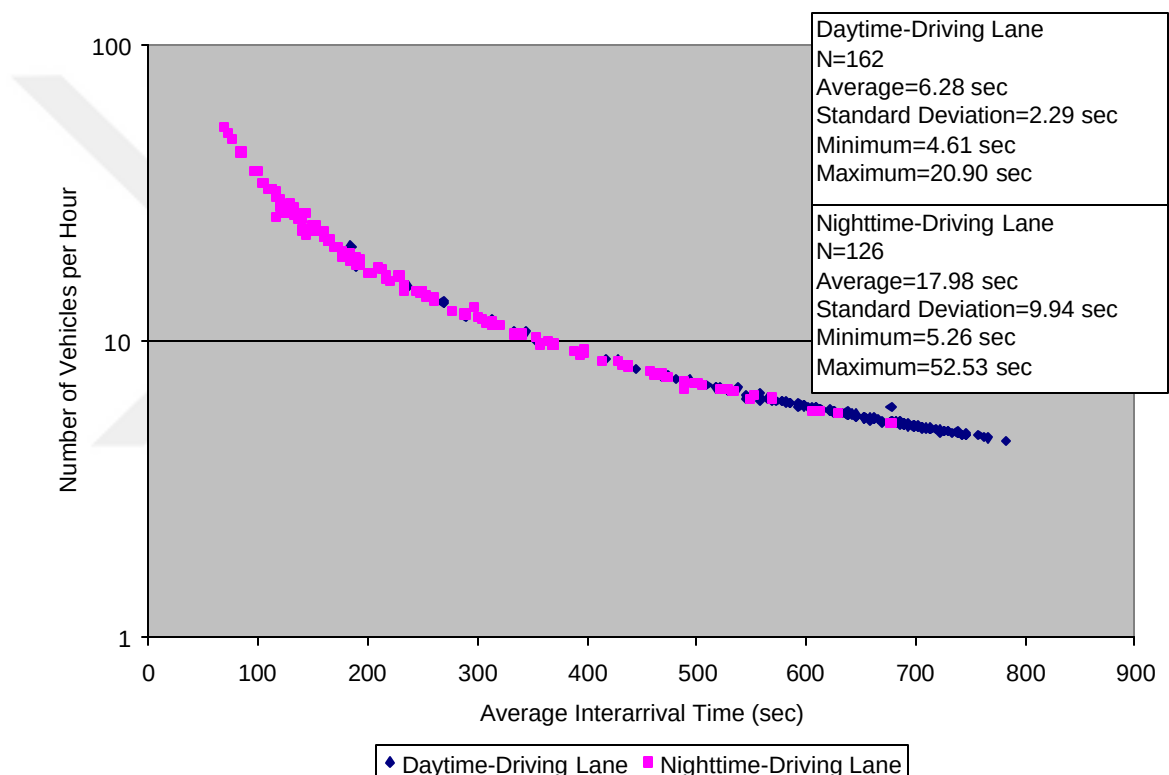


Figure 4: Comparison of Average Inter-arrival Times versus Number of Vehicles per Hour for Daytime and Nighttime Driving Lane Data

Average inter-arrival time for daytime passing lane traffic was 12.07 seconds with a standard deviation of 8.70 seconds. Average inter-arrival time for nighttime passing lane was 135.04 seconds with a standard deviation of 229.16 seconds. The difference in the average inter-arrival times was due to low traffic counts during nighttime.

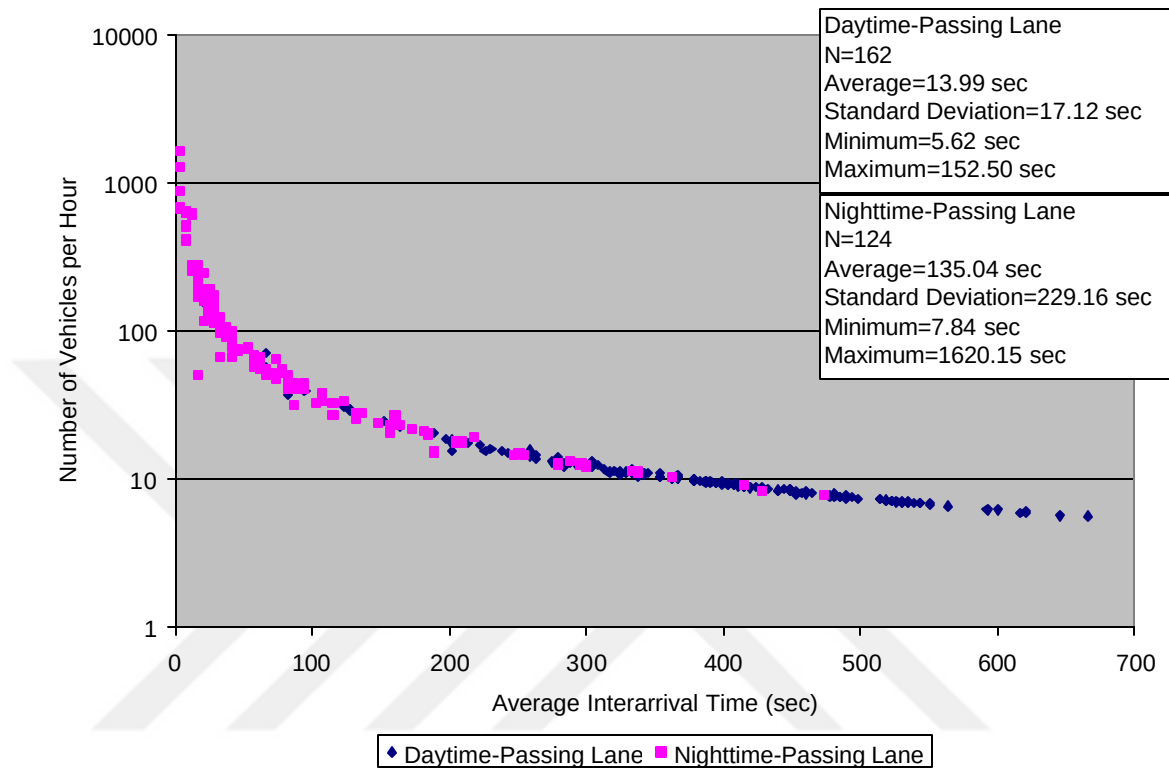


Figure 5: Comparison of Average Inter-arrival Times versus Number of Vehicles per Hour for Daytime and Nighttime Passing Lane Data

As mentioned earlier, the traffic was videotaped with a time-stamped video. Later, the videotape was analyzed and vehicles in each lane were correlated with data records downloaded from the trailer. The recorded data were documented in a Microsoft Excel spreadsheet with the time a vehicle passed the trailer to the nearest second and vehicle type.

The downloaded text file from the trailer was imported into Microsoft Excel, and the videotaped data was entered into a separate worksheet in the same Excel file. Videotaped data records were matched against the radar trailer data, and misses (a vehicle

observed on video but not detected by the trailer) and phantoms (a vehicle detected by the trailer but not observed on video) were identified.

The microwave radar trailer system may sometimes miss vehicles, particularly a small car obscured by a large truck in an intervening lane, and it may sometimes register phantom vehicles from extraneous radar echoes, e.g. from a truck in an adjacent lane. These phantoms and misses are tabulated for the trailer at location 5 in Table 1. The net error is also tabulated. This is the number of misses minus the number of phantoms, thus a positive value represents an undercount by the system (more misses than phantoms). Using the net error values multiplication factors were determined for adjusting the hourly vehicle counts. In Appendix A, the last columns of the tables show the adjusted number of vehicles per hour, which is the product of the hourly vehicle counts and the multiplication factors given in Table 1 for each lane.

Table 1: Misses and Phantoms Observed on I-76 near Rootstown Construction Zone, Ohio (08/20/04 Friday – 10:52AM-11:32AM)

Overall (N=767)				%?=Misses-Phantoms	Multiplication Factor
16	0.02086	2.09%	Phantoms	1.30%	1.0130
26	0.0339	3.39%	Misses		
Driving Lane (N=456)				%?=Misses-Phantoms	Multiplication Factor
8	0.01754	1.75%	Phantoms	0.22%	1.0022
9	0.01974	1.97%	Misses		
Passing Lane (N=321)				%?=Misses-Phantoms	Multiplication Factor
8	0.0249	2.49%	Phantoms	2.80%	1.0280
17	0.0530	5.30%	Misses		

The trailer is also designed to measure the length of passing vehicles and to classify them into length bins. The system has been set up to use three length bins – class 0 is 0-20 feet, class 1 is 21-40 feet, and class 2 is 41 feet and above. Lengths of the vehicles were not measured per each vehicle, but from the videotaped the vehicles were grouped into large trucks (semis, all expected to be over 40 feet) and everything else or “cars”, with some identified as motorcycles. A comparison of the videotaped length categories to the trailer length and classification values from the trailer for August 20 is given in Table 2.

**Table 2: Length Classification of Vehicles by Trailer Compared to Video Record for
August 20, 2004**

		Driving Lane					
		Wavetronix Vehicle Class			total	percent correct	percent incorrect
		<=20ft	21-40ft	>40ft			
OU Vehicle Record	car	263		2	265	99.25%	0.75%
	truck	40		133	173	76.88%	23.12%
	total	303		135	438		
	percent clearly correct	86.80%		98.52%	90.41%		
	percent clearly incorrect	13.20%		1.48%	9.59%		
		Passing Lane					
		Wavetronix Vehicle Class			total	percent correct	percent incorrect
		<=20ft	21-40ft	>40ft			
OU Vehicle Record	car	261		1	262	99.62%	0.38%
	truck	17		8	25	32.00%	68.00%
	total	278		9	287		
	percent clearly correct	93.88%		88.89%	93.73%		
	percent clearly incorrect	6.12%		11.11%	6.27%		
		Both Lanes					
		Wavetronix Vehicle Class			total	percent correct	percent incorrect
		<=20ft	21-40ft	>40ft			
OU Vehicle Record	car	524		3	527	99.43%	0.57%
	truck	57		141	198	71.21%	28.79%
	total	581		144	725		
	percent clearly correct	90.19%		97.92%	91.72%		
	percent clearly incorrect	9.81%		2.08%	8.28%		

3.1.3.1 Converting Hourly Traffic Counts to Cumulative Inter-arrival Time

Distributions

IAT data were generated from the location 5 trailer at I-76 Westbound construction work zone near Rootstown with two lanes of traffic. Three days of traffic data were taken for the analysis and there was uninterrupted traffic flow at the trailer location. The average speed for driving lane was 65.3 mph and that for passing lane was 70.4 mph. The average speeds for each 15 minute interval during daytime for driving and passing lane are given in Appendix A. The speeds given in the tables were collected by the microwave radar trailers. The system records the moving average speed for the last 16 vehicles in the output. The average of two sensors moving average speeds were used in this study.

Vehicles were recorded in the data file with a time stamp with a precision of 0.025 second. At every fifteen minutes, a line listing the number of vehicles observed in that period was inserted in the data file. There were 162 periods for the data collection location. For each period, a table and a graph for the cumulative IAT distribution were established. An example of a graph is shown in Figure 6.

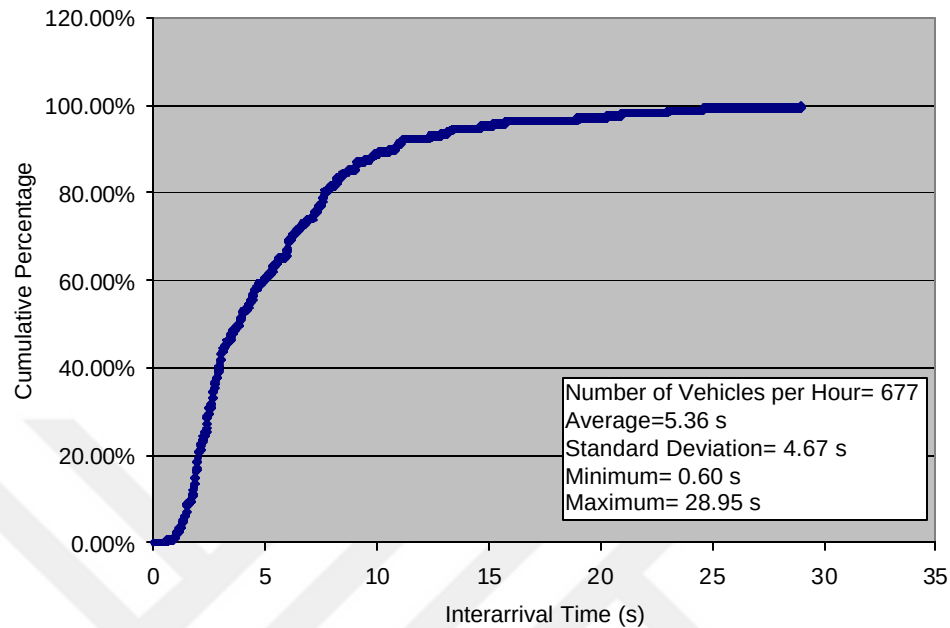


Figure 6: An Actual Cumulative IAT Distribution Observed during a 15 Minute Time Period (08/20/04 Friday, 8:15-8:30 AM)

For each time period a mean and standard deviation of the IATs were determined. For each lane, three time periods were randomly selected and withdrawn for later validation of the model. Additionally, three time periods for the passing lane were removed because they constituted outliers. Hourly flow rates, average IATs, average speeds, and standard deviations for the time intervals used in this study are shown in Appendix A. For each of the marked periods the number of vehicles passing was reported; this number was multiplied by 4 and then by the multiplication factors given in Table 1 according to the lane of the vehicles to obtain a traffic flow rate in vehicles per hour per lane (vphpl).

The cumulative value of 0% was assigned an IAT of 0.1 seconds. From all the time periods in a given set, for the 1% cumulative value, the IATs in seconds were

determined and plotted as a function of the hourly traffic count. A hyperbolic least-squares fit was made to determine the mathematical relationship between the IAT and the hourly traffic count for the 1% cumulative IAT. The same procedure was used to determine the mathematical relationships for the cumulative percentiles of 2%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 95%, 98%, 99%, and 100% (taken as the maximum IAT recorded). Graphs of the IATs as a function of volume at each percentile level are shown in Appendix B, both for driving and passing lanes.

The percentile values were then rearranged into a table of cumulative IATs with each percentile forming a column and each hourly traffic count forming a row. From these tables, graphs of cumulative IATs versus hourly traffic counts were plotted using Microsoft Excel.

For each percentile, inter-arrival time versus hourly traffic count data were fitted using a hyperbolic fit of the form $y = (a/x) + b$. The hyperbolic fits for determining cumulative IAT distributions for given percentiles for driving and passing lane are given in Table 3 and Table 4 respectively. Each of these fits may be used to compute an IAT at that cumulative percentile for a given hourly traffic count. The R^2 values for each fit equation are also shown in Table 3 and Table 4.

Table 3: Hyperbolic Fit Formulae used in Excel Spreadsheet for Determining Cumulative IATs for Selected Percentiles for I-76 Westbound Driving Lane (X is hourly traffic count in vphpl and Y is cumulative IAT in seconds.)

Cumulative Percentage	Model	R ²
0%	0.1	*
1%	$Y = 122.06/X + 0.4947$	0.1824
2%	$Y = 183.83/X + 0.5566$	0.3272
5%	$Y = 198.45/X + 0.8047$	0.3352
10%	$Y = 469.05/X + 0.7137$	0.6722
20%	$Y = 892.65/X + 0.6247$	0.8410
30%	$Y = 1254.82/X + 0.6556$	0.9091
40%	$Y = 1701.76/X - 0.6734$	0.9248
50%	$Y = 2322.24/X + 0.5729$	0.9442
60%	$Y = 3208.26/X + 0.2549$	0.9542
70%	$Y = 4295.84/X - 0.1387$	0.9641
80%	$Y = 5390.20/X + 0.0199$	0.9642
90%	$Y = 7592.15/X - 0.1768$	0.9167
95%	$Y = 10848.2/X - 2.2824$	0.9180
98%	$Y = 12050.26/X - 0.3884$	0.8842
99%	$Y = 12842.42/X + 0.8547$	0.8293
100% (max)	$Y = 13495.82/X + 6.3496$	0.6329

* IAT value for 0% was arbitrarily set to 0.1s.

Table 4: Hyperbolic Fit Formulae used in Excel Spreadsheet for Determining Cumulative IATs for Selected Percentiles for I-76 Westbound Passing Lane (X is hourly traffic count in vphpl and Y is cumulative IAT in seconds.)

Cumulative Percentage	Model	R ²
0%	0.1	*
1%	$Y = 12.22/X + 0.4753$	0.0301
2%	$Y = 29.33/X + 0.5211$	0.0931
5%	$Y = 57.77/X + 0.5776$	0.2743
10%	$Y = 281.99/X + 0.1617$	0.3636
20%	$Y = 443.98/X + 0.1633$	0.6756
30%	$Y = 762.81/X - 0.0794$	0.7104
40%	$Y = 1266.43/X - 0.3614$	0.8680
50%	$Y = 1976.05/X - 0.6918$	0.8916
60%	$Y = 3166.15/X - 1.3412$	0.9036
70%	$Y = 4388.64/X - 1.0848$	0.9378
80%	$Y = 6357.94/X - 1.0673$	0.9319
90%	$Y = 9498.70/X - 0.2287$	0.8455
95%	$Y = 10960.47/X + 5.3074$	0.8692
98%	$Y = 11411.96/X + 14.6193$	0.8042
99%	$Y = 11656.48/X + 21.5875$	0.7491
100% (max)	$Y = 12419.46/X + 32.9255$	0.6723

* IAT value for 0% was arbitrarily set to 0.1s.

Using the hyperbolic fit distributions, a spreadsheet was created that allows a user to type in a volume level in vphpl and extract cumulative distribution function values at 1%, 2%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 95%, 98%, 99%, and 100%. Using the values at these percentiles a cumulative density function for IATs is generated. A table of IAT distribution at cumulative percentile levels as a function of hourly traffic counts for driving lane is given in Table 5 and for passing lane is given in Table 6.

Table 5: IATs in Seconds based on Hyperbolic Formulae Developed for Driving Lane (Min 184 vehicles/hour, Max 782 vehicles/hour)

Number of Vehicles per hour	Cumulative Percentage																
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%
150	0.1	1.31	1.78	2.13	3.84	6.58	9.02	12.02	16.05	21.64	28.50	35.95	50.44	70.04	79.95	86.47	96.32
200	0.1	1.11	1.48	1.80	3.06	5.09	6.93	9.18	12.18	16.30	21.34	26.97	37.78	51.96	59.86	65.07	73.83
250	0.1	0.98	1.29	1.60	2.59	4.20	5.67	7.48	9.86	13.09	17.04	21.58	30.19	41.11	47.81	52.22	60.33
300	0.1	0.90	1.17	1.47	2.28	3.60	4.84	6.35	8.31	10.95	14.18	17.99	25.13	33.88	39.78	43.66	51.34
350	0.1	0.84	1.08	1.37	2.05	3.18	4.24	5.54	7.21	9.42	12.14	15.42	21.52	28.71	34.04	37.55	44.91
400	0.1	0.80	1.02	1.30	1.89	2.86	3.79	4.93	6.38	8.28	10.60	13.50	18.80	24.84	29.74	32.96	40.09
450	0.1	0.77	0.97	1.25	1.76	2.61	3.44	4.46	5.73	7.38	9.41	12.00	16.69	21.82	26.39	29.39	36.34
500	0.1	0.74	0.92	1.20	1.65	2.41	3.17	4.08	5.22	6.67	8.45	10.80	15.01	19.41	23.71	26.54	33.34
550	0.1	0.72	0.89	1.17	1.57	2.25	2.94	3.77	4.80	6.09	7.67	9.82	13.63	17.44	21.52	24.20	30.89
600	0.1	0.70	0.86	1.14	1.50	2.11	2.75	3.51	4.44	5.60	7.02	9.00	12.48	15.80	19.70	22.26	28.84
650	0.1	0.68	0.84	1.11	1.44	2.00	2.59	3.29	4.15	5.19	6.47	8.31	11.50	14.41	18.15	20.61	27.11
700	0.1	0.67	0.82	1.09	1.38	1.90	2.45	3.10	3.89	4.84	6.00	7.72	10.67	13.22	16.83	19.20	25.63
750	0.1	0.66	0.80	1.07	1.34	1.81	2.33	2.94	3.67	4.53	5.59	7.21	9.95	12.18	15.68	17.98	24.34
800	0.1	0.65	0.79	1.05	1.30	1.74	2.22	2.80	3.48	4.27	5.23	6.76	9.31	11.28	14.67	16.91	23.22
Model	0.1	$y = 122.06/x + 0.4947$	$y = 183.83/x + 0.5566$	$y = 198.45/x + 0.8047$	$y = 469.05/x + 0.7137$	$y = 892.65/x + 0.6247$	$y = 1254.82/x + 0.6556$	$y = 1701.76/x + 0.6734$	$y = 2322.24/x + 0.5729$	$y = 3208.26/x + 0.2549$	$y = 4295.84/x - 0.1387$	$y = 5390.20/x + 0.0199$	$y = 7592.15/x - 0.1768$	$y = 10848.2/x - 2.2824$	$y = 12050.26/x - 0.3884$	$y = 12842.42/x + 0.8547$	$y = 13495.82/x + 6.3496$

Table 6: IATs in Seconds based on Hyperbolic Formulae Developed for Passing Lane (Min 66 vehicles/hour, Max 666 vehicles/hour)

Number of Vehicles per hour	Cumulative Percentage																
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%
50	0.1	0.72	1.11	1.73	2.94	9.04	15.18	24.97	38.83	61.98	86.69	126.09	189.75	224.52	242.86	254.72	281.31
100	0.1	0.60	0.81	1.16	1.77	4.60	7.55	12.30	19.07	30.32	42.80	62.51	94.76	114.91	128.74	138.15	157.12
150	0.1	0.56	0.72	0.96	1.38	3.12	5.01	8.08	12.48	19.77	28.17	41.32	63.10	78.38	90.70	99.30	115.72
200	0.1	0.54	0.67	0.87	1.19	2.38	3.73	5.97	9.19	14.49	20.86	30.72	47.26	60.11	71.68	79.87	95.02
250	0.1	0.52	0.64	0.81	1.07	1.94	2.97	4.70	7.21	11.32	16.47	24.36	37.77	49.15	60.27	68.21	82.60
300	0.1	0.52	0.62	0.77	0.99	1.64	2.46	3.86	5.90	9.21	13.54	20.13	31.43	41.84	52.66	60.44	74.32
350	0.1	0.51	0.60	0.74	0.94	1.43	2.10	3.26	4.95	7.70	11.45	17.10	26.91	36.62	47.22	54.89	68.41
400	0.1	0.51	0.59	0.72	0.89	1.27	1.83	2.80	4.25	6.57	9.89	14.83	23.52	32.71	43.15	50.73	63.97
450	0.1	0.50	0.59	0.71	0.86	1.15	1.62	2.45	3.70	5.69	8.67	13.06	20.88	29.66	39.98	47.49	60.52
500	0.1	0.50	0.58	0.69	0.84	1.05	1.45	2.17	3.26	4.99	7.69	11.65	18.77	27.23	37.44	44.90	57.76
550	0.1	0.50	0.57	0.68	0.81	0.97	1.31	1.94	2.90	4.42	6.89	10.49	17.04	25.24	35.37	42.78	55.51
600	0.1	0.50	0.57	0.67	0.80	0.90	1.19	1.75	2.60	3.94	6.23	9.53	15.60	23.57	33.64	41.01	53.62
650	0.1	0.49	0.57	0.67	0.78	0.85	1.09	1.59	2.35	3.53	5.67	8.71	14.38	22.17	32.18	39.52	52.03
700	0.1	0.49	0.56	0.66	0.77	0.80	1.01	1.45	2.13	3.18	5.18	8.02	13.34	20.97	30.92	38.24	50.67
Model	0.1	$y = 12.22/x + 0.4753$	$y = 29.33/x + 0.5211$	$y = 57.77/x + 0.5776$	$y = 116.77/x + 0.6024$	$y = 443.98/x + 0.1633$	$y = 762.81/x - 0.0794$	$y = 1266.43/x - 0.3614$	$y = 1976.05/x - 0.6918$	$y = 3166.15/x - 1.3412$	$y = 4388.64/x - 1.0848$	$y = 6357.94/x - 1.0673$	$y = 9498.70/x - 0.2287$	$y = 10960.47/x + 5.3074$	$y = 11411.96/x + 14.6193$	$y = 11656.48/x + 21.5875$	$y = 12419.46/x + 32.9255$

The average IATs were calculated for the given number of vehicles per hour in Table 5 and Table 6 using the cumulative IAT distribution values. The comparison of the average IATs calculated using the IAT distribution values and hourly vehicle count values are given in Table 7. The average IATs based on volume were calculated by dividing 1 hour (3600 sec) time period by the number of vehicles per hour counts. The results showed that there was a time difference between the average IATs with a range of $\pm 5\%$. The average IATs based on hourly traffic volumes were divided by the average IATs calculated using the IAT distribution and a correction factor for the distribution IATs were generated for different number of vehicles per hour values. Then these correction factors were used to adjust the IATs given in Table 5 and Table 6 for each cumulative percentage value.

Using the correction factors a table of corrected IAT distribution at cumulative percentile levels as a function of hourly traffic counts for driving lane and passing lane were calculated. It was assumed that the distribution shape remains the same, but the IATs at given percentages need to be adjusted to a value that the mean concide with the hourly rates. The corrected IATs are given in Table 8 for driving lane and in Table 9 for passing lane.

Table 7: Comparison of Average IATs Generated using IAT Distribution with the Actual Average IATs

Number of vehicles per Hour	Driving Lane			Passing Lane		
	Average IAT for Fit Distribution (sec)	Average IAT based on Volume (sec)	Correction Factor	Average IAT for Distribution (sec)	Average IAT based on Volume (sec)	Correction Factor
50				68.62	72.00	1.05
100				34.56	36.00	1.04
150	22.93	24.00	1.05	23.21	24.00	1.03
200	17.27	18.00	1.04	17.53	18.00	1.03
250	13.88	14.40	1.04	14.13	14.40	1.02
300	11.61	12.00	1.03	11.86	12.00	1.01
350	10.00	10.29	1.03	10.24	10.29	1.00
400	8.78	9.00	1.02	9.02	9.00	1.00
450	7.84	8.00	1.02	8.07	8.00	0.99
500	7.09	7.20	1.02	7.32	7.20	0.98
550	6.47	6.55	1.01	6.70	6.55	0.98
600	5.95	6.00	1.01	6.18	6.00	0.97
650	5.52	5.54	1.00	5.75	5.54	0.96
700	5.15	5.14	1.00	5.37	5.14	0.96
750	4.82	4.80	1.00			
800	4.54	4.50	0.99			

The IATs for the number of vehicles which are not listed in the tables, linear interpolation is used to calculate the IATs.

Table 8: Corrected IATs in Seconds based on Hyperbolic Formulae Developed for Driving Lane (Min 184 vehicles/hour, Max 782 vehicles/hour)

Number of Vehicles per hour	Cumulative Percentage																
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%
150	0.10	1.37	1.87	2.23	4.02	6.88	9.44	12.58	16.80	22.65	29.83	37.63	52.79	73.31	83.68	90.51	100.82
200	0.10	1.15	1.54	1.87	3.19	5.30	7.22	9.57	12.70	16.98	22.24	28.11	39.38	54.15	62.39	67.81	76.95
250	0.10	1.02	1.34	1.66	2.69	4.35	5.89	7.76	10.23	13.58	17.69	22.40	31.33	42.66	49.62	54.20	62.61
300	0.10	0.93	1.21	1.52	2.35	3.72	5.00	6.56	8.59	11.31	14.65	18.59	25.97	35.01	41.11	45.12	53.05
350	0.10	0.87	1.11	1.41	2.11	3.27	4.36	5.70	7.42	9.69	12.49	15.87	22.14	29.54	35.03	38.64	46.21
400	0.10	0.82	1.04	1.33	1.93	2.93	3.89	5.05	6.54	8.48	10.86	13.83	19.27	25.45	30.47	33.77	41.08
450	0.10	0.78	0.98	1.27	1.79	2.66	3.51	4.55	5.85	7.53	9.60	12.24	17.03	22.27	26.93	29.99	37.08
500	0.10	0.75	0.94	1.22	1.68	2.45	3.22	4.14	5.30	6.78	8.59	10.97	15.25	19.73	24.09	26.97	33.88
550	0.10	0.73	0.90	1.18	1.59	2.27	2.97	3.81	4.85	6.16	7.76	9.94	13.79	17.65	21.78	24.49	31.25
600	0.10	0.70	0.87	1.14	1.51	2.13	2.77	3.54	4.48	5.64	7.07	9.07	12.57	15.92	19.85	22.43	29.06
650	0.10	0.68	0.84	1.11	1.44	2.00	2.60	3.30	4.16	5.21	6.49	8.34	11.54	14.46	18.21	20.68	27.21
700	0.10	0.67	0.82	1.09	1.38	1.90	2.45	3.10	3.89	4.83	5.99	7.72	10.66	13.21	16.81	19.19	25.61
750	0.10	0.65	0.80	1.06	1.33	1.81	2.32	2.93	3.65	4.51	5.56	7.17	9.90	12.12	15.60	17.89	24.23
800	0.10	0.64	0.78	1.04	1.29	1.73	2.20	2.78	3.45	4.23	5.18	6.70	9.23	11.18	14.54	16.76	23.01

Table 9: Corrected IATs in Seconds based on Hyperbolic Formulae Developed for Passing Lane (Min 66 vehicles/hour, Max 666 vehicles/hour)

Number of Vehicles per hour	Cumulative Percentage																
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%
50	0.10	0.76	1.16	1.82	3.08	9.49	15.92	26.20	40.74	65.04	90.96	132.30	199.10	235.58	254.83	267.27	295.18
100	0.10	0.62	0.85	1.20	1.84	4.79	7.86	12.81	19.86	31.58	44.58	65.11	98.70	114.91	134.09	143.90	163.65
150	0.10	0.58	0.74	1.00	1.43	3.23	5.18	8.36	12.91	20.44	29.13	42.72	65.24	81.04	93.78	102.68	119.66
200	0.10	0.55	0.69	0.89	1.22	2.45	3.83	6.13	9.43	14.87	21.41	31.54	48.52	61.71	73.58	81.99	97.55
250	0.10	0.53	0.65	0.82	1.09	1.98	3.03	4.79	7.35	11.54	16.79	24.83	38.49	50.09	61.43	69.52	84.19
300	0.10	0.52	0.63	0.78	1.00	1.66	2.49	3.91	5.97	9.32	13.71	20.37	31.81	42.34	53.29	61.17	75.21
350	0.10	0.51	0.61	0.75	0.94	1.44	2.11	3.27	4.98	7.74	11.51	17.18	27.04	36.80	47.45	55.16	68.74
400	0.10	0.50	0.59	0.72	0.89	1.27	1.82	2.80	4.24	6.56	9.86	14.79	23.47	32.64	43.05	50.62	63.83
450	0.10	0.50	0.58	0.70	0.85	1.14	1.60	2.43	3.67	5.64	8.59	12.94	20.69	29.39	39.61	47.06	59.97
500	0.10	0.49	0.57	0.68	0.82	1.03	1.42	2.14	3.21	4.91	7.57	11.46	18.47	26.79	36.84	44.18	56.84
550	0.10	0.49	0.56	0.68	0.80	0.95	1.28	1.90	2.83	4.31	6.74	10.25	16.65	24.66	34.56	41.81	54.24
600	0.10	0.48	0.55	0.65	0.77	0.88	1.16	1.70	2.53	3.82	6.05	9.25	15.14	22.88	32.65	39.81	52.05
650	0.10	0.48	0.55	0.64	0.75	0.82	1.05	1.53	2.26	3.40	5.46	8.40	13.87	21.37	31.02	38.10	50.16
700	0.10	0.47	0.54	0.63	0.74	0.76	0.97	1.39	2.04	3.05	4.96	7.67	12.77	20.07	29.61	36.62	48.52

Validation results were created by entering in the observed hourly traffic count for the validation data and comparing the output generated by the spreadsheet to the actual observed cumulative IAT distribution generated from three periods of data previously set aside.

In addition to the comparison of the generated IATs with the actual data, the fitting of the generated IATs were also compared with the mathematical distributions available in the literature for IAT calculation. May [24] in his Traffic Flow Fundamentals book, classified the time headway distributions (Inter-arrival Time Distributions) into; random headway state, constant headway state, and intermediate headway state.

Negative exponential distribution is the mathematical distribution that represents the random inter-arrival times. In his study, May compared the field data with the negative exponential distribution. He found that negative exponential distribution does not reflect the actual data characteristics quite well. Negative exponential distribution was best for fitting at low flow levels.

Normal distribution is the mathematical distribution that reflects the constant headway state (constant inter-arrival times). The comparison of two distribution, actual field data distribution and generated normal distribution, showed that the two distributions were quite different. Normal distribution fitted the data best for high traffic flow rates.

For the analysis of intermediate headway state, May used the Pearson type III distribution as an example of the generalized mathematical model approach. The

comparison of the Pearson type III distribution and actual data distribution showed that the two distributions were about the same both for low and high flow rates.

An example calculation procedure for the comparison graphs is given below for 698 vehicles per hour per driving lane data.

The data used for the comparison graphs was set aside before the IAT distribution calculations. The data from 08/22/04 Sunday between 12:45 and 13:00 was used for the comparison. The 15 minute vehicle count for this set was 174 vehicles. This number first multiplied by 4 to get number of vehicles per hour and then it was multiplied by the corresponding correction factor for adjusting phantoms and misses. The adjusted number of vehicles per hour for the data found as 698 vehicles/hour/driving lane. The average IAT was 5.20 seconds with the standard deviation of 3.88 seconds. The minimum observed IAT for this 15 minute interval was 0.48 seconds and the maximum was 22.98 seconds. Histogram data with cumulative percentage values were calculated for the actual data set using MS Excel spreadsheet.

For the OU fitting distribution, the adjusted number of vehicles per hour per lane was used. The IATs for cumulative percentage values 0%, 1%, 2%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 95%, 98%, 99%, and 100% (maximum) were calculated by linear interpolation using Table 8 corrected IATs table for driving lane. The values given in Table 10 were calculated for the IATs using the IAT distribution generated.

Table 10: Cumulative IATs Calculated using the OU Fitting Distribution

Number of Vehicles per Hour Per Driving Lane = 698	
Cumulative Percentage	IAT (sec)
0.01%	0.10
1%	0.67
2%	0.82
5%	1.09
10%	1.39
20%	1.90
30%	2.45
40%	3.11
50%	3.90
60%	4.85
70%	6.01
80%	7.74
90%	10.70
95%	13.26
98%	16.87
99%	19.25
100%	25.68

The cumulative probabilities for the given IATs using negative exponential probability density function were calculated using MS Excel spreadsheet. The formula used for the calculation is shown in (1).

$$f(t) = l \times e^{-lt} \quad (1)$$

where,

t = IAT for which the probability is investigated (x ≥ 0.1 second, the minimum x value (IAT) was taken as 0.1)

l = 0.1922 reciprocal of the mean of the IATs for 15 minute time interval where no. of vphpl was 698)

The values used in Figure 9 a, b are given below in Table 11.

Table 11: Cumulative Percentage Values used for Negative Exponential Distribution in OU Fitting Distribution Comparison Graph (Figure 9 a, b)

Cumulative Percentage	IAT (sec)
0.01%	0.05
1%	0.11
2%	0.27
5%	0.55
10%	1.16
20%	1.86
30%	2.66
40%	3.61
50%	4.77
60%	6.26
70%	8.37
80%	11.98
90%	15.59
95%	20.35
98%	23.96
100%	83.85

For the normal distribution, the MS Excel spreadsheet function was used for the calculation. MS Excel Normal Distribution function calculates the cumulative probability function, which is the integral from negative infinity to x in the formula (2).

$$f(x, m, s) = \frac{1}{\sqrt{2\pi} s} e^{-\left(\frac{(x-m)^2}{2s^2}\right)} \quad (2)$$

x = IAT for which cumulative probability is investigated (x ≥ 0.1 second, the minimum x value (IAT) was taken as 0.1 seconds)

$m = 5.20$ seconds ($\infty < m < \infty$) (average of the IATs for 15 minute time interval where no. of vphpl was 698)

$s = 3.88$ seconds ($s > 0$) (standard deviation of the IATs for 15 minute time interval where no. of vphpl was 698)

The values used in Figure 9 c, d are given below in Table 12.

Table 12: Cumulative Percentage Values used for Normal Distribution in OU Fitting Distribution Comparison Graph (Figure 9 c, d)

Cumulative Percentage	IAT (sec)
8.98%	0.1
13.91%	1
20.44%	2
28.50%	3
37.82%	4
47.92%	5
58.15%	6
67.86%	7
76.48%	8
83.64%	9
89.21%	10
93.26%	11
96.03%	12
97.79%	13
98.84%	14
99.43%	15
99.73%	16
99.88%	17
99.95%	18
99.98%	19
99.99%	20
100.00%	21

The cumulative probabilities for the given IATs using Pearson Type III distribution were calculated using Matlab. The probability density function for the Pearson Type III Distribution is given in (3).

$$f(t) = \frac{1}{\Gamma(K)} [l(t-a)]^{K-1} e^{-l(t-a)} \quad (3)$$

where,

t = IAT for which the probability is investigated ($t \geq 0.1$)

l = parameter that is a function of the mean time headway and the two user specified parameters, K and a . ($l = \frac{K}{\bar{t} - a} = 0.258$, where $\bar{t}$ (average of the sample) = 5.20 seconds for 15 minute time interval where no. of vphpl was 698)

K = user selected parameter between 0 and ∞ that affects the shape of the distribution ($\hat{K} = \frac{\bar{t} - a}{s} = 1.3777$, where, $\bar{t}$ (average of the sample) = 5.20 seconds, s (standard deviation of the sample) = 3.88 seconds, for 15 minute time interval where no. of vphpl was 698)

$a = 0.1$ user selected parameter greater than or equal to zero that affects the shift of the distribution (IATs less than this value will have a 0 probability)

e = constant parameter, 2.71828

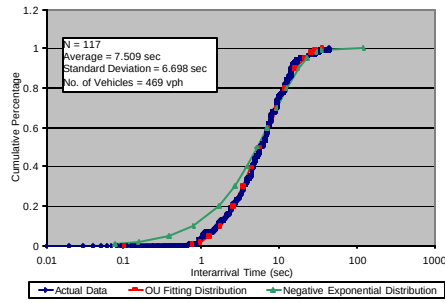
$\Gamma(K)$ = gamma function, equivalent to $\int_0^\infty e^{-x} x^{K-1} dx$ for all $K \geq 0$

Cumulative IAT values used in Figure 9 e, f are given below in Table 13. The cumulative probabilities were obtained by numerically integrating the $f(t)$, equation (3) using Δt of 0.10 seconds (Matlab trapezoidal rule).

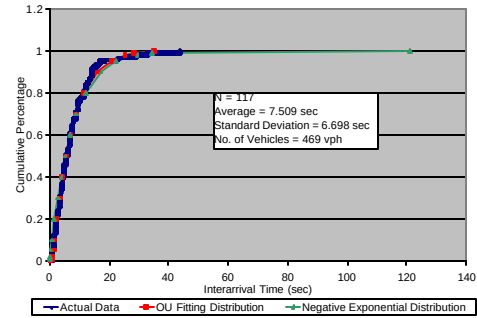
Table 13: Cumulative Percentage Values used for Pearson Type III Distribution in OU Fitting Distribution Comparison Graph (Figure 9 e, f)

Cumulative Percentage	IAT (sec)	Cumulative Percentage	IAT (sec)	Cumulative Percentage	IAT (sec)
0.40%	0.1	94.05%	13.5	99.58%	27.5
4.75%	0.5	94.69%	14	99.61%	28
11.50%	1	95.77%	15	99.63%	28.5
18.60%	1.5	96.22%	15.5	99.64%	29
25.60%	2	96.63%	16	99.66%	29.5
32.32%	2.5	96.98%	16.5	99.67%	30
38.64%	3	97.30%	17	99.68%	30.5
44.53%	3.5	97.58%	17.5	99.69%	31
49.96%	4	97.83%	18	99.70%	31.5
54.95%	4.5	98.05%	18.5	99.71%	32
59.50%	5	98.25%	19	99.72%	32.5
63.64%	5.5	98.42%	19.5	99.73%	33
67.40%	6	98.58%	20	99.73%	33.5
70.80%	6.5	98.72%	20.5	99.74%	34
73.86%	7	98.84%	21	99.74%	34.5
76.62%	7.5	98.95%	21.5	99.75%	35
79.11%	8	99.04%	22	99.75%	35.5
81.34%	8.5	99.13%	22.5	99.75%	36
83.34%	9	99.20%	23	99.75%	36.5
85.13%	9.5	99.27%	23.5	99.76%	37
86.73%	10	99.33%	24	99.76%	37.5
88.16%	10.5	99.38%	24.5	99.76%	38
89.44%	11	99.42%	25	99.76%	38.5
90.59%	11.5	99.46%	25.5	99.76%	39
91.61%	12	99.50%	26	99.77%	39.5
92.52%	12.5	99.53%	26.5	100.00%	40
93.33%	13	99.56%	27		

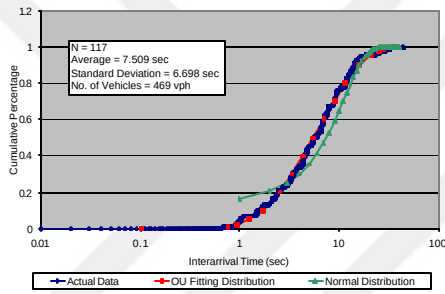
The comparison graphs for diving lane are given in Figure 7, Figure 8, and Figure 9, and the graphs for passing lane are given in Figure 10, Figure 11, and Figure 12.



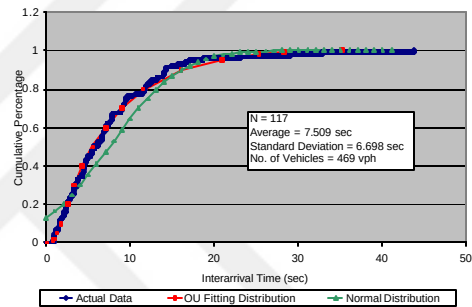
a



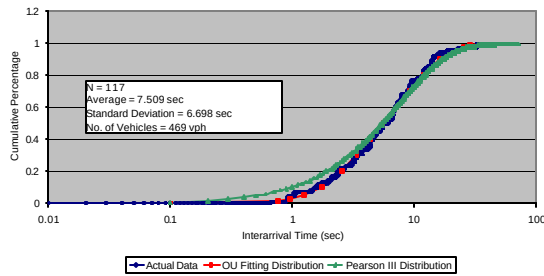
b



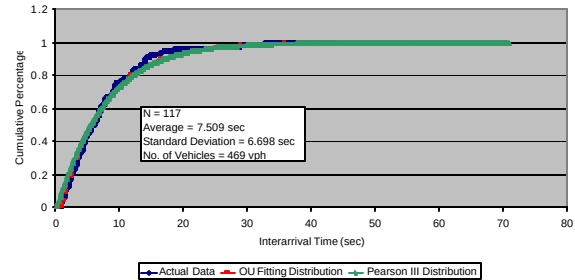
c



d

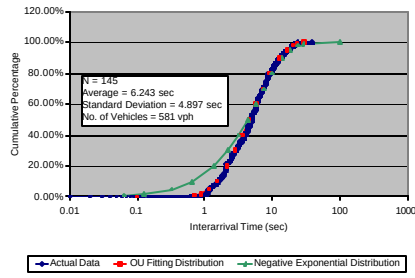


e

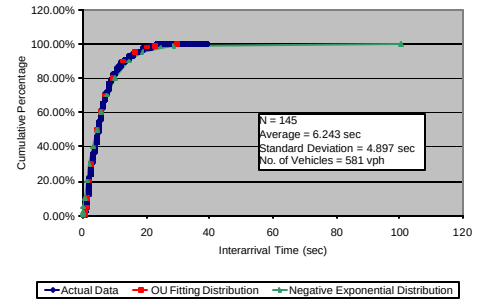


f

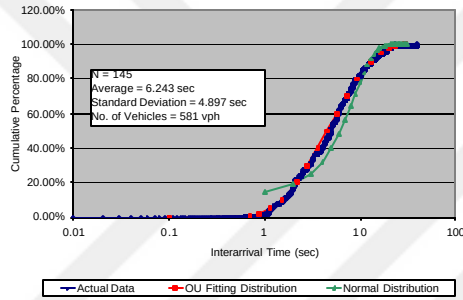
Figure 7: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 469 vehicles/hour/driving lane



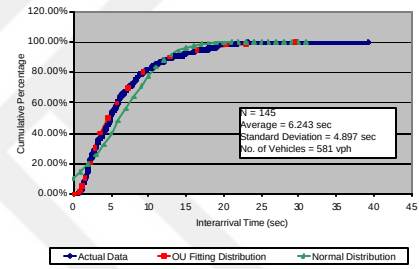
a



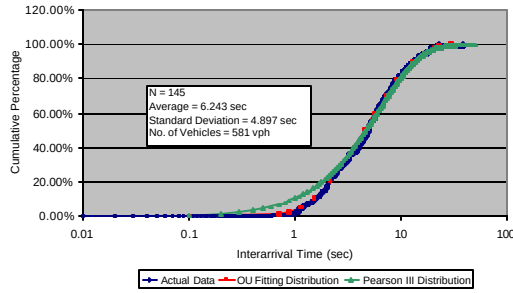
b



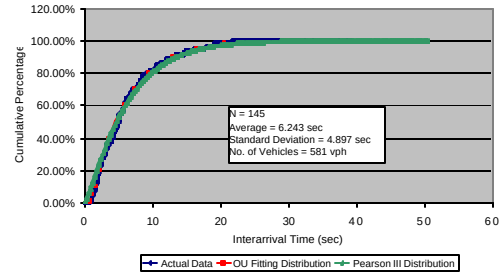
c



d

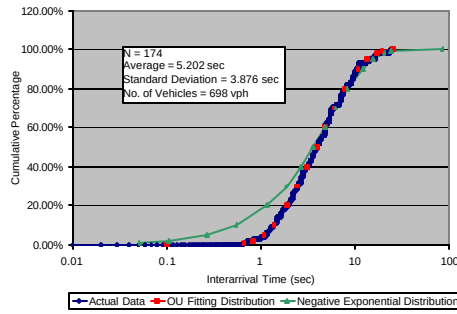


e

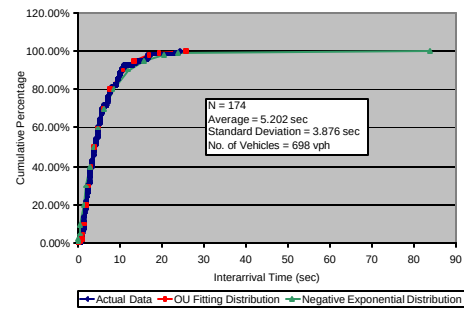


f

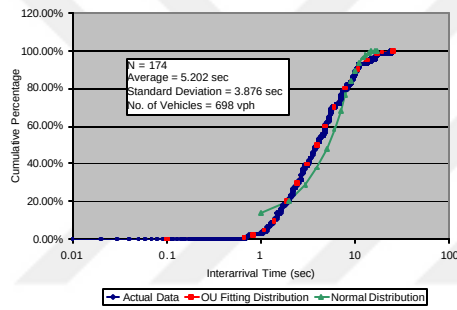
Figure 8: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 581 vehicles/hour/driving lane



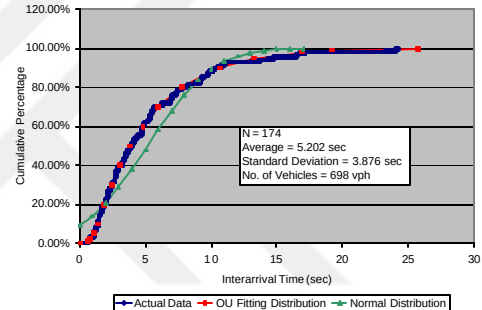
a



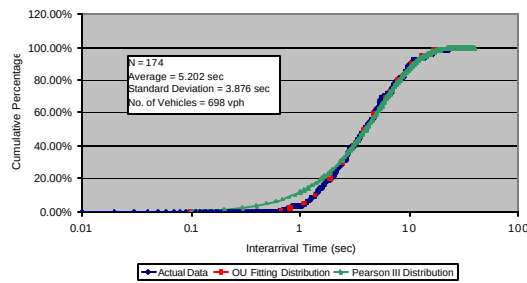
b



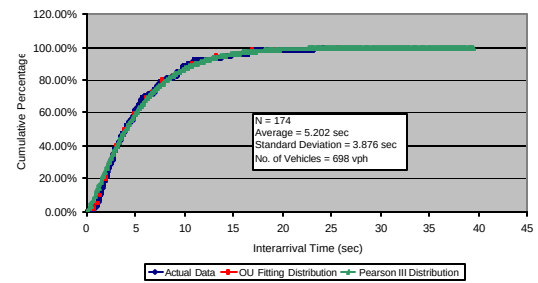
c



d



e



f

Figure 9: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 698 vehicles/hour/driving lane

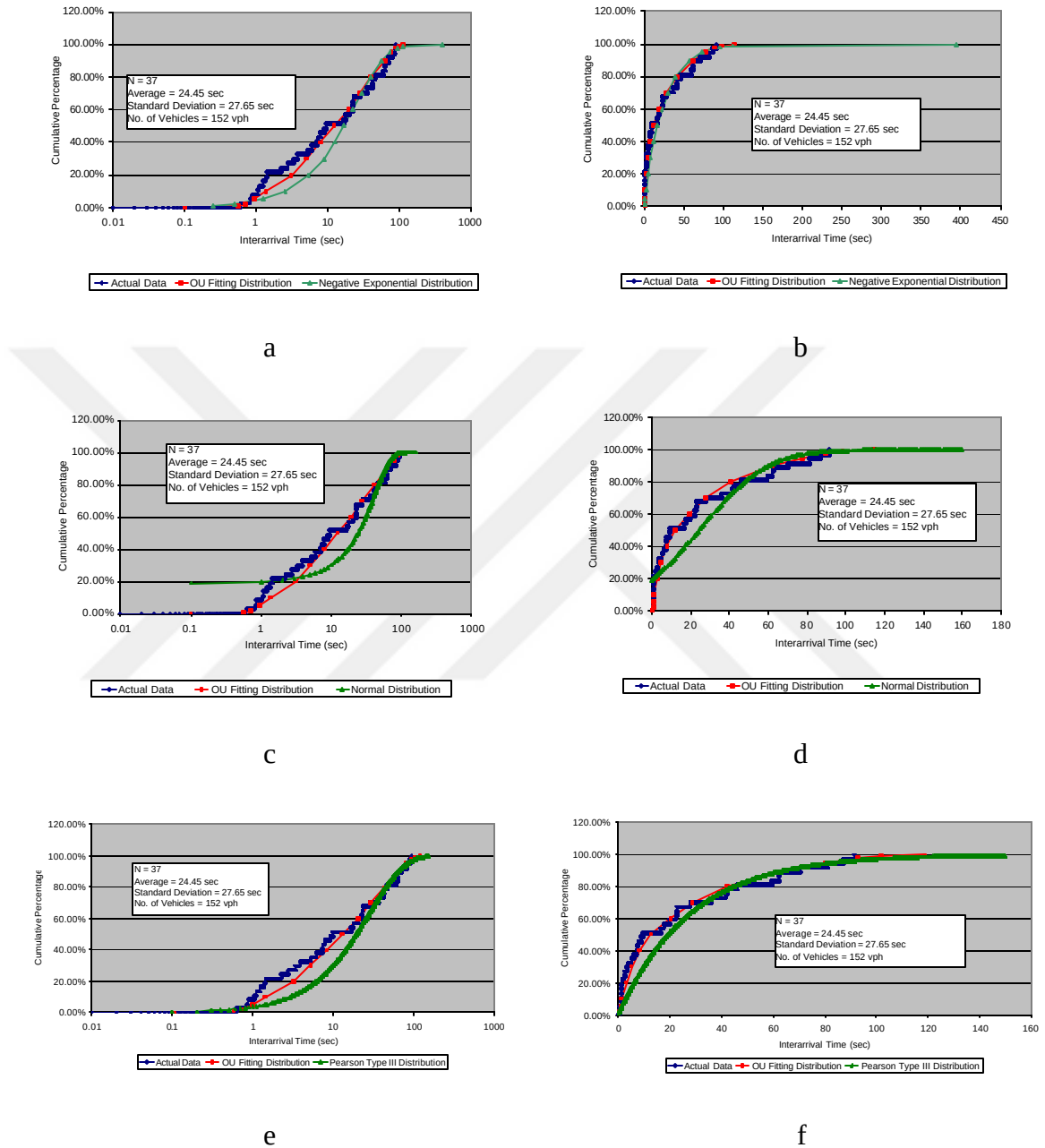


Figure 10: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 152 vehicles/hour/passing lane

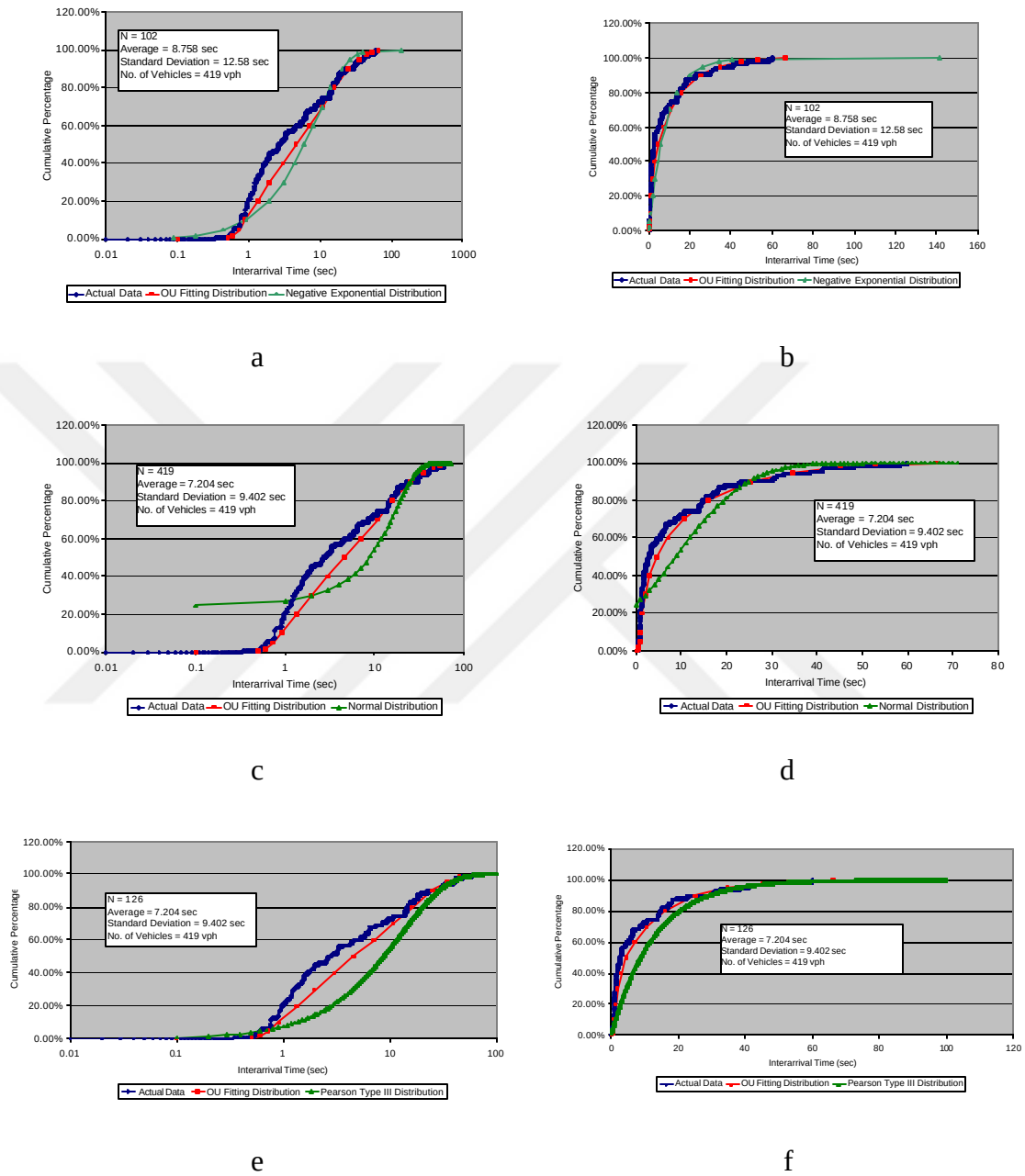
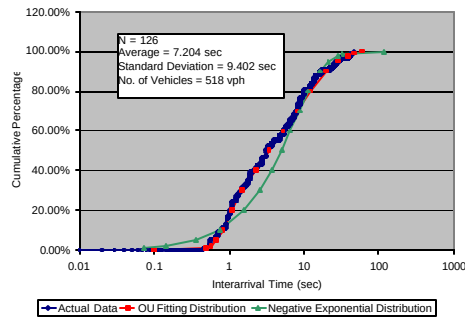
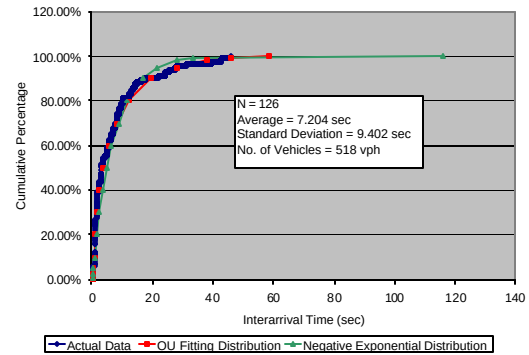


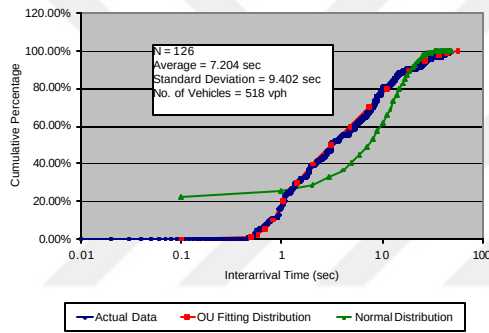
Figure 11: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 419 vehicles/hour/passing lane



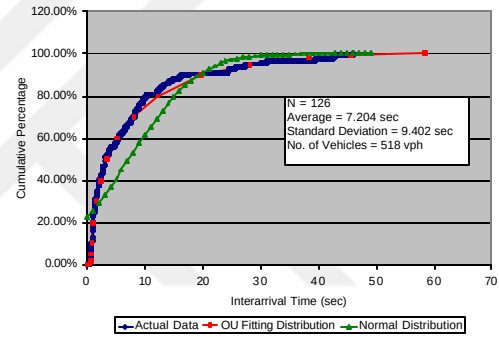
a



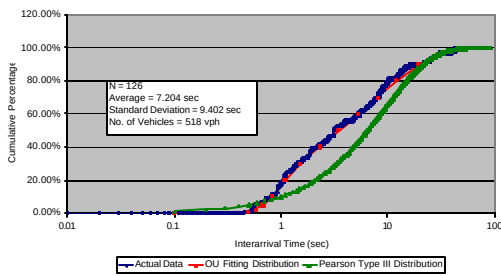
b



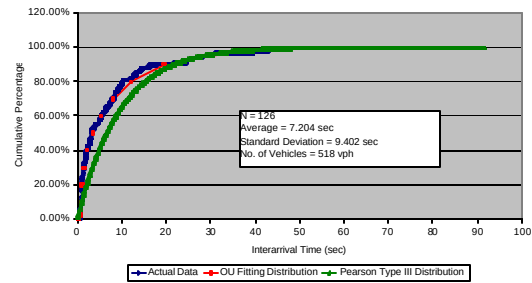
c



d



e



f

Figure 12: Comparison of Actual Inter-arrival Times, OU Fitting Distribution, and Negative Exponential Distribution (a (Logarithmic Scale), b), Normal Distribution (c (Logarithmic Scale), d), Pearson Type III Distribution (e (Logarithmic Scale), f) for 518 vehicles/hour/passing lane

The comparison of the proposed cumulative IAT distribution fit the actual data better than the three distributions given for the IATs. It should also be noted that the given mathematical distributions provide probability tables from which you can get the probability of given inter-arrival time. In the proposed method for determining cumulative IAT distribution one can only enter the hourly traffic volume and get the inter-arrival times for the given cumulative percentage values.

The validation shows that this method produces fairly accurate cumulative IAT distributions in the hourly traffic count range the data was taken in. The cumulative IAT distributions show a fairly close hyperbolic relationship between higher percentile values and hourly traffic counts, as shown by the higher R^2 values in Table 3 and Table 4. As expected, a similar hyperbolic relationship holds between the average IAT and the hourly traffic counts, as seen in Appendix A. The conversion approach presented here, using a least squares fit to get the best relationship between cumulative IATs and hourly traffic counts and implemented in an easy to use Excel spreadsheet works quite well. The observed relatively strong hyperbolic relationships between the IAT averages and the hourly traffic counts indicate that even under fairly different traffic flows with all their randomness, a robust relationship appears to exist between the average IAT and the hourly traffic count.

Additional work using a representative sample of other sites will be required to demonstrate that this conversion approach is generally valid.

3.2 Description and Design of ARENA (SIMAN) Simulation Program

3.2.1 Description of ARENA (SIMAN) Simulation Program

ARENA simulation software research version 7.01 by Rockwell Automation was used to model the traffic flow in construction zone. ARENA software is designed to model queues. The software program takes the inter-arrival time probability density functions and service time probability density functions as inputs [26]. And the ARENA software can model multiple lanes as multiple queues.

The vehicles entering the work zone was simulated using the transporter module of ARENA. The transporter module of ARENA allows the programmer to enter the speed distribution for the vehicles. The inter-arrival times of the entities (vehicles) entering the construction work zones were determined according to the field data collected. The vehicle type assignment (i.e. car or truck) along with the lengths were made at the point of entry. The vehicles were represented with the entity-transporter pairs in the system. When the vehicles entered the system, the speed distribution functions were assigned to the vehicles using transporters.

Another important issue in traffic simulation was the headway and the spacing between the vehicles. Headway is defined as the time between successive vehicles as they pass a point on a lane or roadway, using a common reference point on both vehicles and spacing is defined as the distance between successive vehicles in a traffic stream, as measured from front bumper to front bumper [23]. In order to specify the headways and spacing between the vehicles, initially, the headway and spacing were determined according to the arrival time of the vehicles. The difference in vehicle arrival times

determined the headways and spacing. Additional length was added to the length of the vehicles to maintain the minimum spacing during simulation.

The travel time (delay time) in the work zone was determined according to the speed profiles of the vehicles in the work zone.

The model was evaluated using the data collected in the field. The use of real world data allowed us to compare the outputs of the model and to analyze the accuracy of its outputs.

About 1750 statements were used in the program to model traffic. Most of the modules used at the stations for lane changing behavior were used with small changes in the statements. One simulation run (replication) took about 30 minutes on a PC with 2.8 GHz processor and 1.0 GB RAM.

3.2.2 Design of the ARENA Traffic Simulation Model

3.2.2.1 Input Variables

The traffic data was entered into ARENA simulation program by using number of variables which defined the system. According to these input variables the program computed the output variables. The following variables were entered in order to get the output variables.

3.2.2.1.1 Construction Zone Configuration

The model was developed for two lane traffic configuration. Construction work was performed on the passing lane of the road and the passing lane was closed for 5600 feet. The map of the construction zone is given in Figure 1. Vehicles were simulated

beginning from 700 feet in advance of the first “Left Lane Closed Ahead” sign. Then the second and third lane closure signs were placed 1200 feet and 1700 feet after the beginning of the simulation. After 500 feet from the third left lane closure sign, the lane closure taper was begun. The length of the lane closure taper was 150 feet. One lane road (after the left lane merge) shifted to the shoulder at 2300 feet with 300 feet taper. After the merging of left lane and shifting of right lane to the shoulder, one lane road in construction zone was simulated for 350 feet. After 3000 feet from the beginning of the simulation start point, vehicles left the system. The dimensioning and the placement of the traffic signs and drums are given in Figure 13.

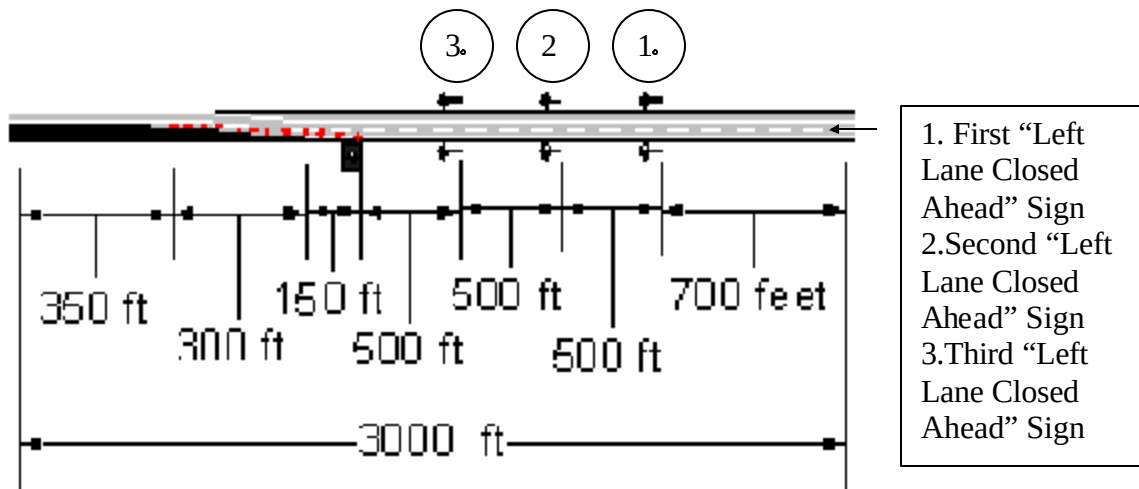


Figure 13: Work Zone Configuration used in the Example Simulation

The construction zone in ARENA simulation program was modeled by using the intersections and links elements of the program [26]. Intersections were defined at the changes in the construction zone. And the links defined the distances between the intersections. These two elements defined the network system in the simulation.

Originally the system was simulated for 14000 feet. It was starting 7950 feet before the left lane closure taper and ending 6050 feet after the taper. The simulation run length using the original distances took 3 – 5 hours for 1 replication. In order to decrease the simulation runtime, the distances for the construction zone were reduced. The reduction of the distances would not cause any loss of data, since the objective of the model was to identify the queue length and the waiting time at the lane closure taper.

3.2.2.1.2 Inter-arrival Time Distribution (Entity Arrival)

Vehicle entry to the system was modeled using the create module of ARENA simulation program. Actual inter-arrival times of the vehicles at the beginning of the work zone were collected in the field. As mentioned earlier in section 3.1.3.1 [page 26] the inter-arrival times of the vehicles were analyzed and a spreadsheet for determining the cumulative percentages of inter-arrival times for given hourly traffic counts was established. Using the spreadsheet, inter-arrival times of vehicles for 24-hour time period for a weekday were determined using the actual count data from 08/20/2004 Friday.

Hourly vehicle counts were gathered from the field data collected. As mentioned earlier, 3 days of data was collected in the field. Figure 14 and Figure 15 shows the hourly vehicle counts for 3 days of data, both for driving and passing lanes. From the

graphs it can be easily observed that the daytime hourly vehicle counts are greater than the nighttime counts.

In addition, it was observed that the counts for the data collected on 08/20/2004 were greater than the hourly vehicle counts of the other days. The hourly vehicle counts for the day with the maximum number of vehicles per hour 08/20/2004 was selected for the simulation example. The output variables of the simulation model are the queue length and the waiting time at the lane closure taper, and it was expected to be greater for the time periods with high hourly vehicle counts.

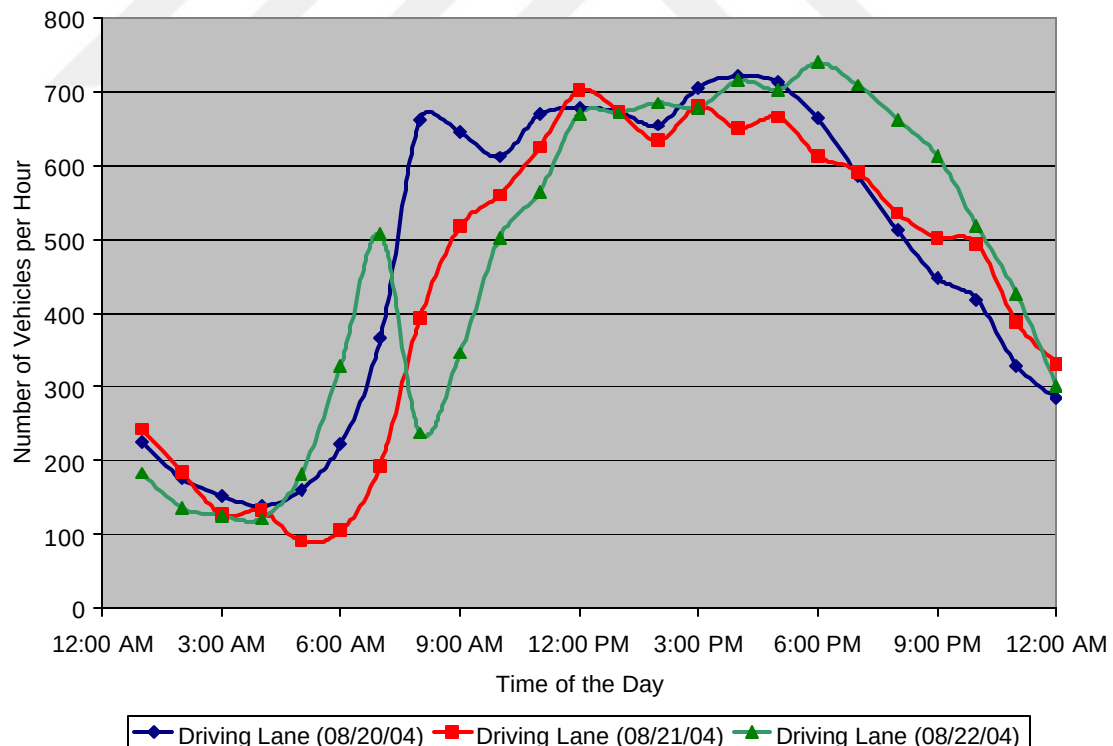


Figure 14: Hourly Traffic Counts for Driving Lane for 3 Days of Data

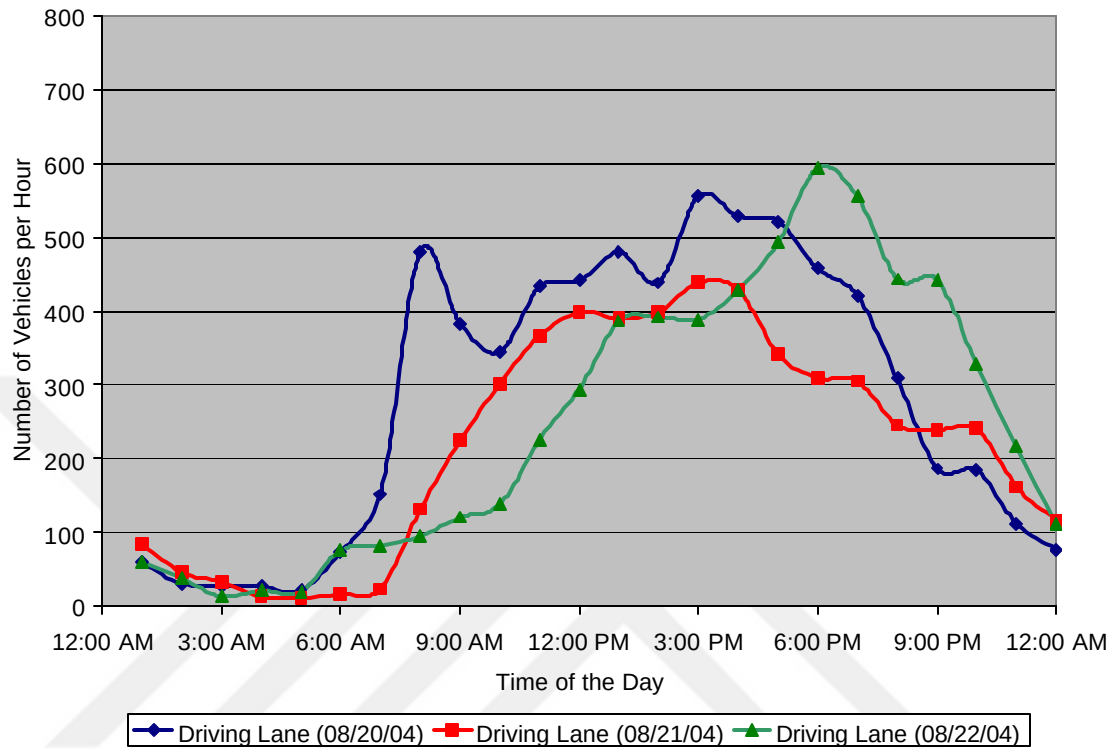


Figure 15: Hourly Traffic Counts for Passing Lane for 3 Days of Data

In Appendix C adjusted hourly vehicle counts for 08/20/2004 (Friday) daytime data and their corresponding inter-arrival times for given cumulative percentages are presented for driving lane and passing lane. Inter-arrival times for each cumulative percentage values were calculated according to the corrected IAT tables given in Table 8 and Table 9. The IATs for the vehicle counts which are not given in the table were calculated using a linear interpolation.

The inter-arrival times and their cumulative percentage values given for each 15 minute interval in Appendix C were entered to the create module of ARENA. Continuous probability distribution function was used for the arrival rates. The inter-arrival times for

the vehicles for the create module were entered with the given expression using the inter-arrival times and their cumulative percentages for corresponding number of vehicles per hour per lane.

CONT(0.0001,0.1,0.01,0.71,0.02,0.88,0.05,1.16,0.1,1.54,0.2,2.20,0.3,2.88,0.4,3.68,0.5,4.68,0.6,5.93,0.7,7.46,0.8,9.56,0.9,13.25,0.95,16.91,0.98,20.93,0.99,23.57,1,30.23)

3.2.2.1.3 Vehicle Types

Vehicles entered to the system as entities and they are transported in the system with guided vehicles transport option of the ARENA simulation program. Two different vehicle types were defined in the ARENA simulation model; passenger cars and trucks. The percentages of trucks and passenger cars were determined using the field data collected. It was assumed that the percentages of trucks and cars remain the same for the 24 hour simulation time period. Using the OU video record data presented in Table 2, the percentages of trucks and cars were determined which are given in Table 14, both for driving and passing lane. These percentages were assigned to the vehicles at the beginning of the simulation run.

Table 14: Percentages of Passenger Cars and Trucks for Driving Lane and Passing Lane according to OU Video Record

Percentages		
Vehicle Type	Driving Lane	Passing Lane
Passenger Car	91%	9%
Truck	61%	39%

In the simulation model, passenger cars were simulated using the guided vehicle module of ARENA. Vehicle length, acceleration and deceleration rates, speed at the beginning of the simulation can be assigned to this module.

The passenger car guided vehicle module variables were adopted from Traffic Engineering Handbook [23]. The default acceleration and deceleration rates of the guided vehicles in ARENA were used for the passenger cars. The lengths of passenger cars in the model were assumed to be 16 feet. In addition to this 16 feet length, 4 feet was added to the length of the passenger cars to prevent the collision of the vehicles in stopped traffic.

In the model trucks were also simulated using the guided vehicle module of ARENA. The default acceleration and deceleration rates of the guided vehicles were used in ARENA for the trucks. Trucks in the model were assumed to be 60 feet in the model. In addition to the 60 feet vehicle length, 5 feet was added to the length of the trucks to prevent the collision of the vehicles in stopped traffic.

3.2.2.1.4 Speed Profile

The speeds of the vehicles were assigned using a probability density function (pdf) at the beginning of the simulation. Different flow speeds were assigned to each vehicle with the function. These speeds were assigned as the desired speeds of the vehicles during simulation. If the vehicles need to decelerate or accelerate according to the conditions of the roadway, the default acceleration and deceleration rates are used in the simulation. Using a pdf for determining the speed profile also allowed us to simulate different driver types.

The most commonly used mathematical distributions for representing speed profile are the normal, log-normal, and composite distributions [24]. The normal distribution was used in the ARENA simulation model to simulate the traffic. The cumulative frequency graphs of the actual data and the normal distribution using actual data parameters are given in Figure 16 for driving lane and in Figure 17 for passing lane.

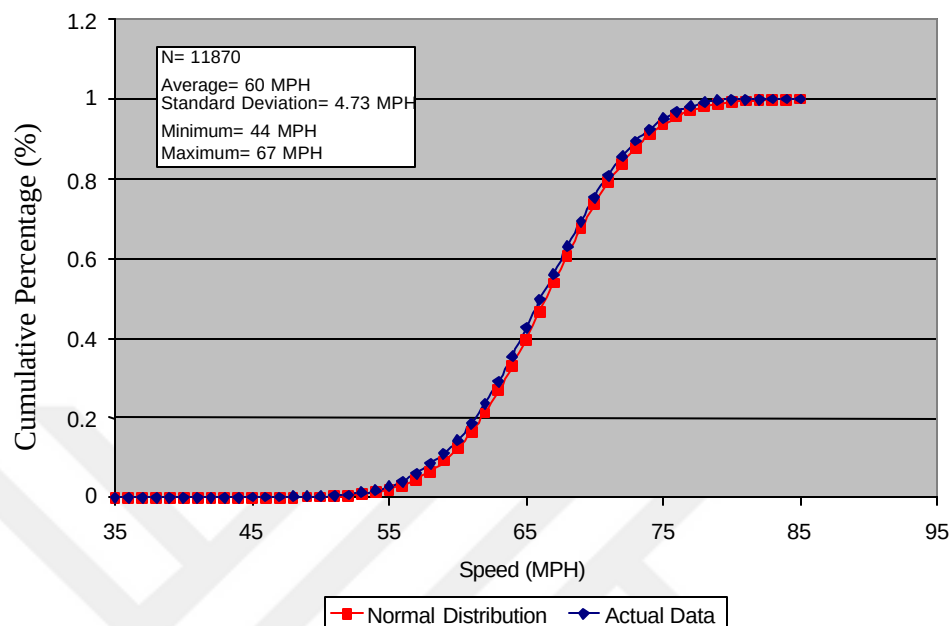


Figure 16: Comparison of Actual Speed Data for Driving Lane Collected on 08/20/2004 Friday with the Normal Distribution

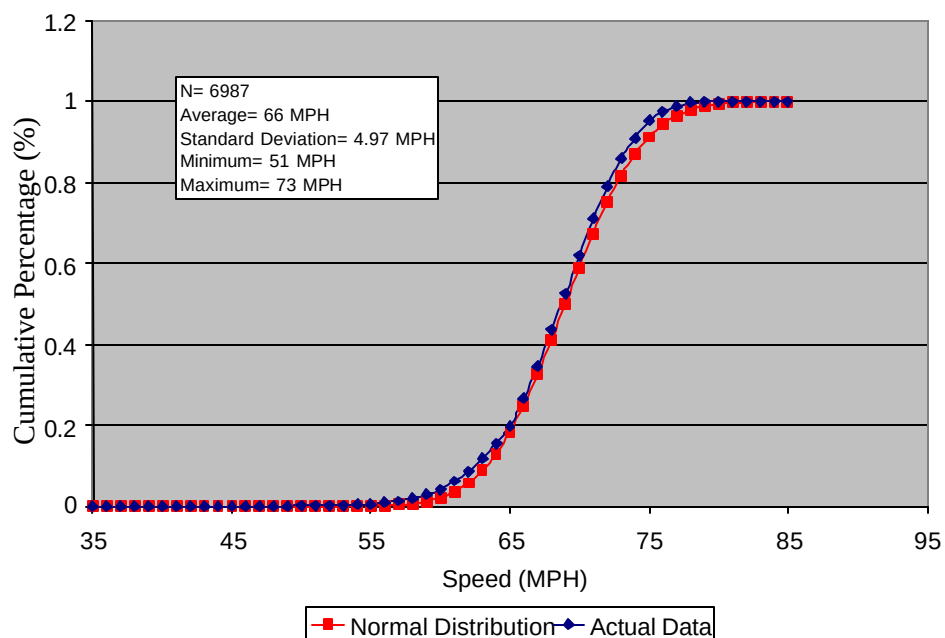


Figure 17: Comparison of Actual Speed Data for Passing Lane Collected on 08/20/2004 Friday with the Normal Distribution

The mean and the standard deviation for the speeds at the beginning of the work zone were calculated using the field data and they were entered into the model. In addition to assigning speeds to the vehicles at the beginning of the simulation for free flow conditions, speeds were also assigned to the vehicles at the beginning of the taper according to the average and standard deviations calculated using the field data. In Table 15, the assigned speeds for the vehicles before the construction zone and at the construction zone are given for both driving and passing lane. As mentioned earlier the microwave radar trailers provide the moving average speed of last 16 vehicles recorded in the output. The averages and standard deviations given below are the moving average speed of 16 vehicles. The same distributions were used for the 24 hour simulation time period.

Table 15: Vehicle Speeds used for the Transporters in the ARENA Simulation Model

	At the Beginning of Simulation		At First Merging Sign	
	Average (feet/second)	Standard Deviation (feet/second)	Average (feet/second)	Standard Deviation (feet/second)
Driving Lane	88	3.47	75	10.27
Passing Lane	97	3.67	84	8.80

3.2.2.1.5 Car Following Behavior

The car following behavior of the vehicles is dependent on the gap acceptance of the following vehicles. The distance between two consecutive vehicles provides safe following distance for the vehicles. The safe distance is dependent on the speed of the vehicles, length of the vehicles, deceleration and acceleration rates of the vehicles, and response time of the drivers.

In ARENA simulation program, the guided vehicle property of the transporters module automatically adjusts the distance between the transporters according to their acceleration and deceleration rates and speeds. The model prevents the collision between the transporters. However, in the simulation model minimum following distance is also specified for the vehicles. The car following distance between the vehicles is not less than 50 ft when the vehicles are in free flow conditions. The car following distance is not less than 25 ft for the jam density conditions, and when there are vehicles waiting in the queue for merging the minimum distance between the vehicles is not less than 4 feet for passenger cars and not less than 5 feet for trucks. The safe distances between two vehicles when they are stopping are integrated into the model using additional lengths for the vehicle length attribute of the transporters as it was mentioned earlier. The distance between two vehicles was not less than these minimum distances.

3.2.2.1.6 Gap Acceptance for Merging and Lane Changing Behavior

Gap acceptance is the main factor affecting the lane changing behavior. When a termination of a lane occurs, the vehicles on that lane have to stop until they can find

sufficient gap on the traveling lane. The acceptable gap for changing lanes is dependent on the speed of the vehicle which will change lane, and the speed of the vehicles on the other lane. The distance (gap) between two vehicles is calculated using the link and zone properties of ARENA simulation program. Lengths are assigned to the links and each link is composed of zones which have different lengths. According to the speeds of the vehicles, the model checks the occupancy of the links and zones for the needed gap and if there is an available distance for changing lanes, the vehicles change their lanes. The required gap is also dependent on the type of the vehicles. There are two types of vehicles simulated in the model; passenger cars and trucks. Their length is added to the required gap for lane changing. Figure 18 shows the placement of the vehicles in the gap acceptance calculations.

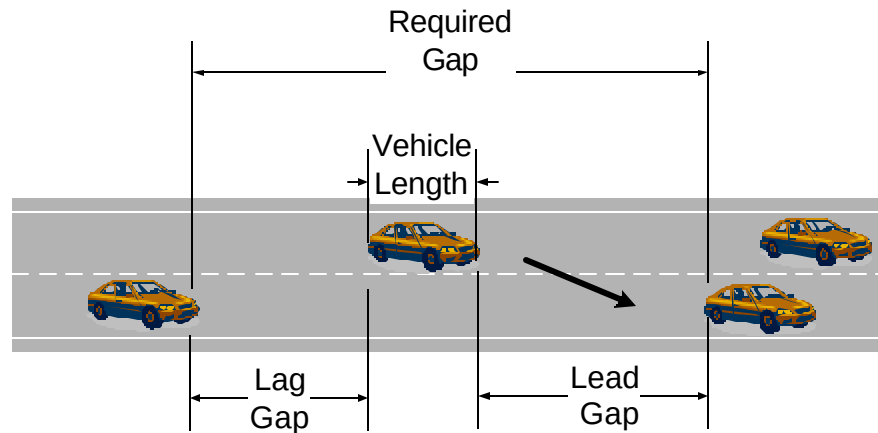


Figure 18: Gap Acceptance

Kanaris et al. [32] defined a minimum safety spacing during lane changing (MSSLC) between the leading vehicle and following vehicle on the desired travel lane

for the lane changing vehicle. He stated that the distance between two vehicles should not be less than MSSLC to avoid collisions. MSSLC includes the probability of emergency braking conditions in order to simulate the real traffic conditions in detail. The researchers calculated the MSSLC for different conditions.

In the ARENA simulation model, MSSLC calculated using the assumption that acceleration rates and deceleration rates for the lane changing vehicle, leading vehicle, and following vehicle are the same.

The gaps required for lane changing were derived from the figures given below. In Figure 19 the space required for leading vehicle gap is given according to the difference in speeds between the two vehicles.

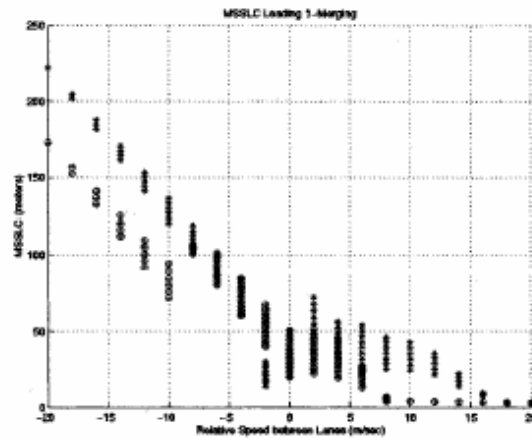


Figure 19: MSSLC for the space between the leading vehicle and the merging vehicle versus relative speed between the lanes (adapted from Kanaris et al. [32])

In Figure 20 the space required for following vehicle gap is given according to the speed differences between two vehicles.

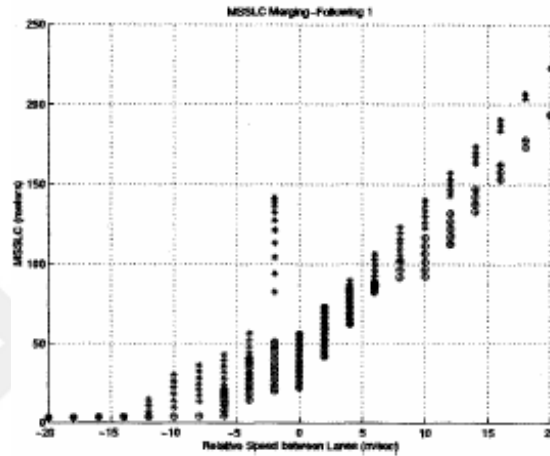


Figure 20: MSSLC for the space between the following vehicle and the merging vehicle versus relative speed between the lanes (adapted from Kanaris et al. [32])

The values derived from the graphs were added to the vehicle lengths and the minimum required gaps for lane changing were calculated. Table 16 shows the required distances for the passenger cars and trucks. Required distances for lane changing given in Table 16 are the distances used before the end of the lane. There are no vehicles waiting in the queue for lane changing. The MSSLCs derived from the graphs presented by Kanaris et al. did not match with the field data collected at I-76 Westbound since in real life conditions lane changing does not occur at safe space distances all the time.

**Table 16: Minimum Safe Space for Lane Changing before the Lane Closure Taper
(adapted from Kanaris et al. [32])**

	Minimum Safe Spacing for Lane Changing (ft)					
Speed Difference (Speed on Adjacent Lane – Merging Vehicle) (ft/s)	Passenger Car				Truck	
	Lead Gap	Vehicle Length	Lag Gap	Total	Vehicle Length	Total
Speed Difference < -50 ft/s	200 ft	20 ft	850 ft	1070 ft	65 ft	1115 ft
-50 < Speed Difference ≤ -35 ft/s	160 ft	20 ft	540 ft	720 ft	65 ft	765 ft
-35 < Speed Difference ≤ -15 ft/s	100 ft	20 ft	450 ft	570 ft	65 ft	615 ft
-15 < Speed Difference ≤ 15 ft/s	20 ft	20 ft	180 ft	220 ft	65 ft	265 ft
15 < Speed Difference ≤ 35 ft/s	20 ft	20 ft	130 ft	170 ft	65 ft	215 ft
35 < Speed Difference ≤ 50 ft/s	20 ft	20 ft	80 ft	120 ft	65 ft	165 ft
50 < Speed Difference	20 ft	20 ft	30 ft	70 ft	65 ft	115 ft

Rakha and Crowther [33] compared three most known car following models in the literature; Greenshields, Pipes, and Van Aerde car following models. Greenshields is a single regime macroscopic car following model which uses two parameters for determining the car following distance. Free speed and the capacity or the jam density is used for determining the car following distances in this model. Pipes car following model is a two regime microscopic model, which uses the free speed, jam density, and driver sensitivity factor. Van Aerde car following model uses four parameters; free speed, speed at capacity, jam density, and capacity. The authors compared these 3 car following

models and then compared the Van Aerde model's results with the real world data. It is shown that the Van Aerde car following model gives the best fitting distribution to the real headway distance data. In Figure 21, comparison of Van Aerde car following model with the field data is given. In Figure 22, the comparison of three models fitting distributions are given.

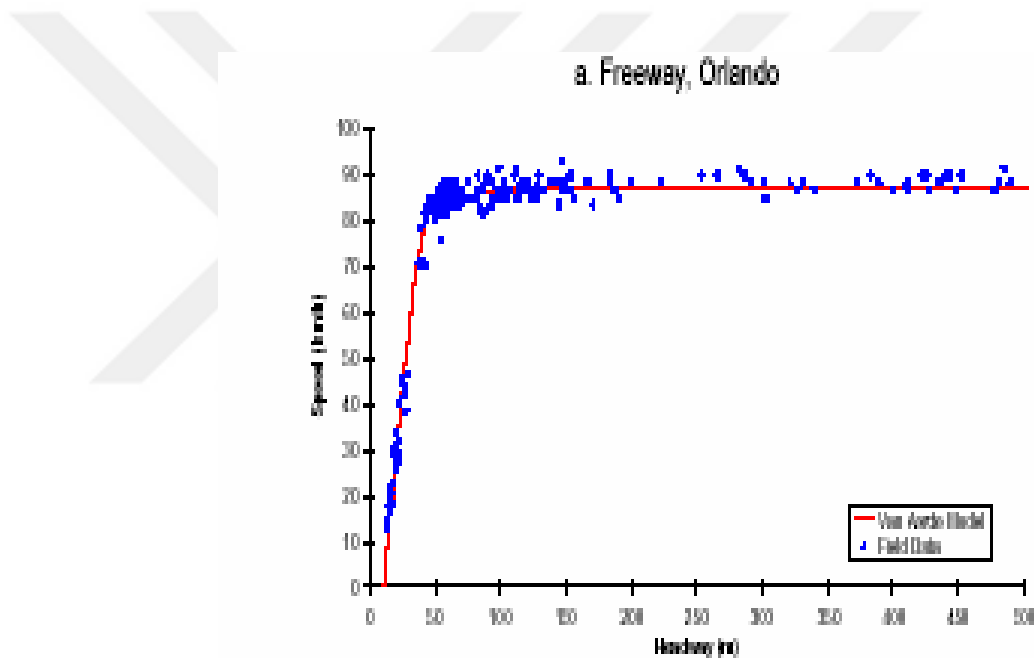


Figure 21: Comparison of the Van Aerde Car Following Model with the Field Data
(adapted from Rakha and Crowther [33])

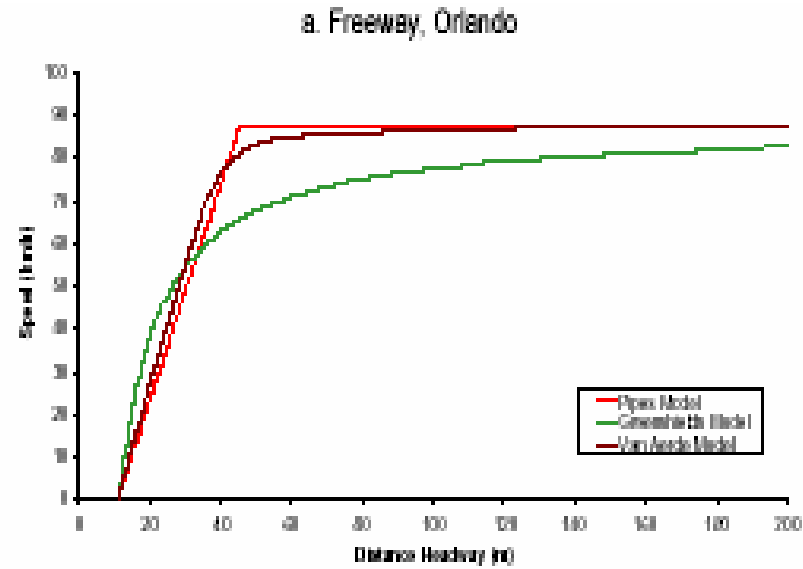


Figure 22: Comparison of Greenshields, Pipes, and Van Aerde Car Following Models (adapted from Rakha and Crowther [33])

Using Figure 21, comparison of Van Aerde model with the field data graphs, required gaps for lane changing were generated. Required gaps are given in Table 17. The speeds of the leading vehicle, following vehicle, and the merging vehicle are assumed to be the same.

Table 17: Required Gaps for Lane Changing Maneuver Derived from Van Aerde Car Following Model (adapted from Rakha and Crowther [33])

	Required Gaps for Lane Changing (ft)					
Speed	Passenger Car				Truck	
	Lead Gap	Vehicle Length	Lag Gap	Total	Vehicle Length	Total
Speed ≤ 10 ft/s	40 ft	20 ft	40 ft	100 ft	65 ft	150 ft
$10 \text{ ft/s} < \text{Speed} \leq 35 \text{ ft/s}$	65 ft	20 ft	65 ft	150 ft	65 ft	200 ft
$35 \text{ ft/s} < \text{Speed} \leq 55 \text{ ft/s}$	100 ft	20 ft	100 ft	225 ft	65 ft	275 ft
$55 \text{ ft/s} < \text{Speed} \leq 75 \text{ ft/s}$	130 ft	20 ft	130 ft	275 ft	65 ft	325 ft
$75 \text{ ft/s} < \text{Speed}$	165 ft	20 ft	165 ft	350 ft	65 ft	400 ft

Lane changing behavior is one of the most important features of a traffic simulation model. There are two types of lane changing behavior in microscopic traffic simulation; mandatory and discretionary lane changing behaviors.

Mandatory lane changing is defined as changing the lane when there is a termination of lane. When there is a termination of the lane, the vehicles start looking for the availability of the space between two consecutive vehicles on the next lane when they see the closed lane traffic sign. From that point vehicles compute the distance available between two consecutive vehicles on the next lane. The vehicles maintain their traveling lane until they find available lane changing space on the next lane. If there is no available space for lane changing, the vehicles travel on their lane until the lane ends, and they stop at the end of the lane. They wait until they find available space for lane changing.

Other type of lane changing is the discretionary lane changing behavior. Discretionary lane changing occurs when passing over a low speed vehicle or when yielding another merging vehicle. When there is an available space for lane changing, the vehicle which will change the lane checks the speed of the leading vehicle on the other lane. If the speed of the leading vehicle is greater than the lane changing vehicle, the lane changing vehicle accelerates during merging and after it changes the lane it adjusts its speed according to the car following behavior. If the speed of the leading vehicle is less than the lane changing vehicle, it decelerates until it reaches the speed of the leading vehicle during merging. After it completes merging, it adjusts its speed according to the car following behavior.

Discretionary lane changing behavior is not included in this simulation model. Vehicles in the construction zone simulation example model only have the mandatory lane changing behavior.

Lane changing distances are dependent on the gap acceptance of the drivers. The literature showed that the speeds of the vehicles determine the car following distances of the drivers and the lane changing distances are determined according to these car following distances. In addition to the speed of the vehicles, the relative speed of the vehicles on different lanes affect the distance required for lane changing. Using the derived lane changing distances from Kanaris et al. [32] and Rakha and Crowther [33], the required lane changing space for the traffic simulation model was derived. If the speed of the merging vehicle is greater than or equal to the speeds of the leading and following vehicles on the other lane the values given in Table 18 are used for lane

changing space. If the speed of the merging vehicle is less than the leading and the following vehicles on the other lane the values given in Table 19 are used for lane changing.

Table 18: Required Space for Lane Changing when the Merging Vehicle Speed is Greater or equal than the Desired Lane Speed

	Required Gaps for Lane Changing (ft)					
Speed	Passenger Car				Truck	
	Lead Gap	Vehicle Length	Lag Gap	Total	Vehicle Length	Total
Speed ≤ 10 ft/s	40 ft	20 ft	5 ft	65 ft	65 ft	110 ft
10 ft/s < Speed ≤ 35 ft/s	65 ft	20 ft	10 ft	95 ft	65 ft	140 ft
35 ft/s < Speed ≤ 55 ft/s	100 ft	20 ft	20 ft	140 ft	65 ft	185 ft
55 ft/s < Speed ≤ 75 ft/s	130 ft	20 ft	20 ft	170 ft	65 ft	215 ft
75 ft/s < Speed	165 ft	20 ft	30 ft	215 ft	65 ft	260 ft

Table 19: Required Space for Lane Changing when the Merging Vehicle Speed is less than the Desired Lane Speed

	Required Gaps for Lane Changing (ft)					
Speed	Passenger Car				Truck	
	Lead Gap	Vehicle Length	Lag Gap	Total	Vehicle Length	Total
Speed ≤ 10 ft/s	5 ft	20 ft	40 ft	65 ft	65 ft	110 ft
10 ft/s $<$ Speed ≤ 35 ft/s	10 ft	20 ft	65 ft	95 ft	65 ft	140 ft
35 ft/s $<$ Speed ≤ 55 ft/s	20 ft	20 ft	100 ft	140 ft	65 ft	185 ft
55 ft/s $<$ Speed ≤ 75 ft/s	20 ft	20 ft	130 ft	170 ft	65 ft	215 ft
75 ft/s $<$ Speed	30 ft	20 ft	165 ft	215 ft	65 ft	260 ft

Another important decision in determining the required space for the lane changing is the stopped traffic conditions. When the vehicles traveling on the lane which will be closed, could not find enough space, they stop at the lane closure taper. The number of vehicles waiting in the queue for lane changing affects the drivers' gap acceptance behavior. When there are vehicles waiting in the queue, drivers are more likely to change their lanes even if there is smaller space available on the other lane. It was assumed that the merging at the stopped traffic conditions occurs in a first come first serve basis. The vehicle at the beginning of the queue merges first all the time.

In Table 20 minimum space used in the simulation model for lane changing when there are less than 3 vehicles waiting in the queue are given. The minimum space was calculated with the assumption that when there are vehicles waiting in the queue they are

willing to change lanes with smaller gaps, which is less than their travel distance in one second, when there are no vehicles waiting.

In Table 21 required space for lane changing when there are less than 5 but more than 3 vehicles waiting in the queue are given. In Table 22 required space for lane changing when there are less than 10 but more than 5 vehicles waiting in the queue are given. In Table 23 required space for lane changing when there are more than 10 vehicles waiting in the queue is given.

Table 20: Required Space for Lane Changing when there are Less than 3 Vehicles waiting in the Queue at the Lane Closure Taper

	Required Gaps for Lane Changing (ft)					
Speed	Passenger Car				Truck	
	Lead Gap	Vehicle Length	Lag Gap	Total	Vehicle Length	Total
Speed ≤ 10 ft/s	5 ft	20 ft	30 ft	55 ft	65 ft	95 ft
10 ft/s $<$ Speed ≤ 35 ft/s	10 ft	20 ft	45 ft	75 ft	65 ft	120 ft
35 ft/s $<$ Speed ≤ 55 ft/s	20 ft	20 ft	55 ft	95 ft	65 ft	140 ft
55 ft/s $<$ Speed ≤ 75 ft/s	20 ft	20 ft	65 ft	105 ft	65 ft	150 ft
75 ft/s $<$ Speed	30 ft	20 ft	90 ft	140 ft	65 ft	185 ft

Table 21: Required Space for Lane Changing when there are Less than 5 Vehicles waiting in the Queue at the Lane Closure Taper

	Required Gaps for Lane Changing (ft)					
Speed	Passenger Car				Truck	
	Lead Gap	Vehicle Length	Lag Gap	Total	Vehicle Length	Total
Speed ≤ 10 ft/s	5 ft	20 ft	20 ft	45 ft	65 ft	90 ft
10 ft/s $<$ Speed ≤ 35 ft/s	10 ft	20 ft	35 ft	65 ft	65 ft	110 ft
35 ft/s $<$ Speed ≤ 55 ft/s	20 ft	20 ft	45 ft	85 ft	65 ft	130 ft
55 ft/s $<$ Speed ≤ 75 ft/s	20 ft	20 ft	55 ft	95 ft	65 ft	140 ft
75 ft/s $<$ Speed	30 ft	20 ft	80 ft	130 ft	65 ft	175 ft

Table 22: Required Space for Lane Changing when there are Less than 10 Vehicles waiting in the Queue at the Lane Closure Taper

	Required Gaps for Lane Changing (ft)					
Speed	Passenger Car				Truck	
	Lead Gap	Vehicle Length	Lag Gap	Total	Vehicle Length	Total
Speed ≤ 10 ft/s	5 ft	20 ft	10 ft	35 ft	65 ft	80 ft
10 ft/s $<$ Speed ≤ 35 ft/s	10 ft	20 ft	25 ft	55 ft	65 ft	100 ft
35 ft/s $<$ Speed ≤ 55 ft/s	20 ft	20 ft	35 ft	75 ft	65 ft	120 ft
55 ft/s $<$ Speed ≤ 75 ft/s	20 ft	20 ft	45 ft	85 ft	65 ft	130 ft
75 ft/s $<$ Speed	30 ft	20 ft	70 ft	120 ft	65 ft	165 ft

Table 23: Required Space for Lane Changing when there are More than 10 Vehicles waiting in the Queue at the Lane Closure Taper

	Required Gaps for Lane Changing (ft)					
Speed	Passenger Car				Truck	
	Lead Gap	Vehicle Length	Lag Gap	Total	Vehicle Length	Total
Speed ≤ 10 ft/s	5 ft	20 ft	5 ft	30 ft	65 ft	75 ft
10 ft/s $<$ Speed ≤ 35 ft/s	10 ft	20 ft	15 ft	45 ft	65 ft	90 ft
35 ft/s $<$ Speed ≤ 55 ft/s	20 ft	20 ft	25 ft	65 ft	65 ft	110 ft
55 ft/s $<$ Speed ≤ 75 ft/s	20 ft	20 ft	35 ft	75 ft	65 ft	120 ft
75 ft/s $<$ Speed	30 ft	20 ft	60 ft	110 ft	65 ft	155 ft

3.2.2.2 Flowchart of the ARENA Simulation Model

Traffic in the simulation model was simulated for two-lane road reduced to one lane at I-76 Westbound construction work zone near Rootstown, Ohio. The flowchart of the ARENA simulation model is given in Figure 23. The SIMAN model and experimental codes of the model are given in Appendix D.

The model starts with the creation of entities. The arrival of the entities (vehicles) are simulated by the create modules in ARENA as described in section 3.2.2.1.2 (page 54). The arrival rate is a discrete distribution dependent on the number of vehicles per hour per lane. Separate ARENA create modules are used for driving lane and passing lane for each 15 minute time interval.

After the arrival of the entities into the system according to the lanes, vehicle types are assigned according to the percentages given in Table 14. According to the vehicle type, the entities request transporters as cars or trucks. Four different transporter sets are defined in the model; car for driving lane, truck for driving lane, car for passing lane, and truck for passing lane. Initial speeds of the transporters are assigned as 95 feet per second. And the lengths of the transporters are assigned as described in vehicle types section.

After entities are assigned to different types of transporters, they request the corresponding transporters to the first station defined. Entities are paired with the transporters and then the speeds of the transporters are assigned according to a normal distribution. Two different speed distributions are assigned in the model, one for passing lane and one for driving lane. Assigning different speeds, which were determined according to the field data, incorporated the driver types in the model. Some of the transporters have higher speeds and some has lower speeds. The transporters then transferred to the next station, which is the first left lane closure warning sign. Until the first sign transporters do not change lanes.

Transporters with entities move through the system using the links determined with the SIMAN elements modules. For each location of interest, a station is assigned and these stations are assigned with the intersections, and then the intersections are connected with the links. Transporters move on these links, which represents the roadway.

After arriving to the station at the first lane closure sign, the speeds of the transporters are reduced because of the construction zone speed limits. The new speeds are assigned with a normal distribution function using the values given in Table 15. Then the transporters on the passing lane start looking for the available conditions for lane changing before they reach to the end of the passing lane. They first check the speed of the transporter on the driving lane to determine the needed space for lane changing. According to the difference in the speeds of the transporter on passing lane and driving lane, the needed space for the transporter on the passing lane is determined. The space is also dependent on the vehicle type. If the speed of the transporter on driving lane is greater than the speed of the transporter on the passing lane, the transporter on the passing lane requires more lag gap for lane changing, else it requires more lead gap. The required space for lane changing according to the difference in speeds of passing lane transporter and driving lane transporter are given in Table 16. If the required space is available, transporter with the entity on the passing lane advances to the next station on the driving lane, else it continues on passing lane to the next station. Same procedure continues at the stations, which are 1250 feet, 1000 feet, 900 feet, 800 feet, 700 feet, 600 feet, 500 feet, 450 feet, 400 feet, 350 feet, 300 feet, 250 feet, 200 feet, 150 feet, 100 feet, and 50 feet in advance of the passing lane closure taper for the transporters on the passing lane. The transporters on the driving lane move through the road by adjusting their speeds according to the merging traffic from the passing lane when needed.

When the transporters on the passing lane did not merge to the driving lane until the lane closure taper, they first check the number of vehicles waiting at the lane closure taper station on passing lane. If there are:

- 3 or less vehicles waiting at the taper, they require the space values given in Table 20 for lane changing.
- More than 3 and 5 or less vehicles waiting, they require the space values given in Table 21 for lane changing.
- More than 5 and 10 or less vehicles waiting, they require the space values given in Table 22 for lane changing.
- More than 10 vehicles waiting, they require the space values given in Table 23 for lane changing.

If there is no space available for merging, the transporters stop and wait at the taper until they find the required space for lane changing.

The transporters on the driving lane continue on driving lane by adjusting their speeds when needed because of the merging transporters from passing lane.

After the taper, the road becomes a one-lane road and all the vehicles on driving lane and passing lane continue on this one-lane road until they leave the system 750 feet after the lane closure taper.

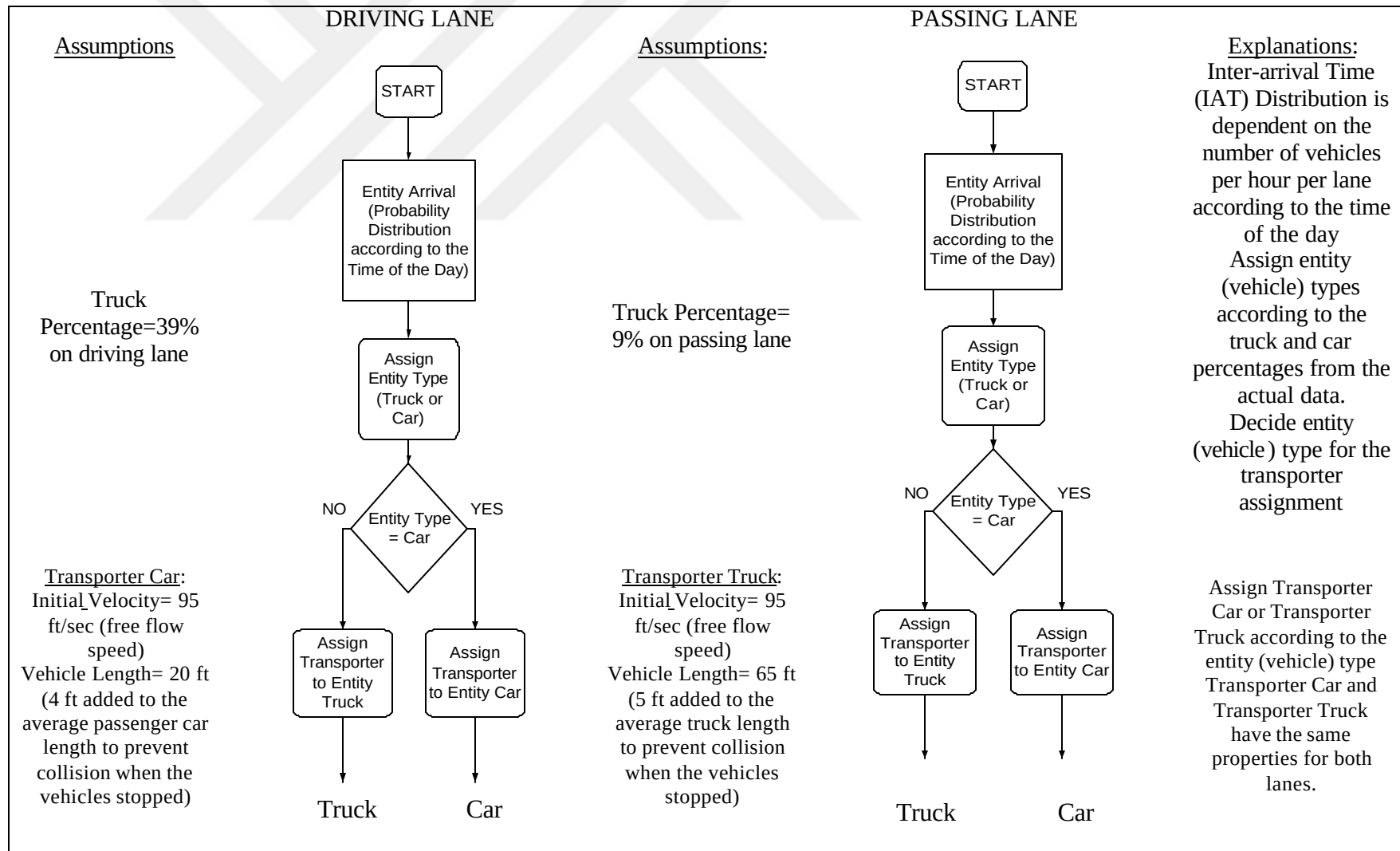


Figure 23: Flowchart of the ARENA Traffic Simulation Model

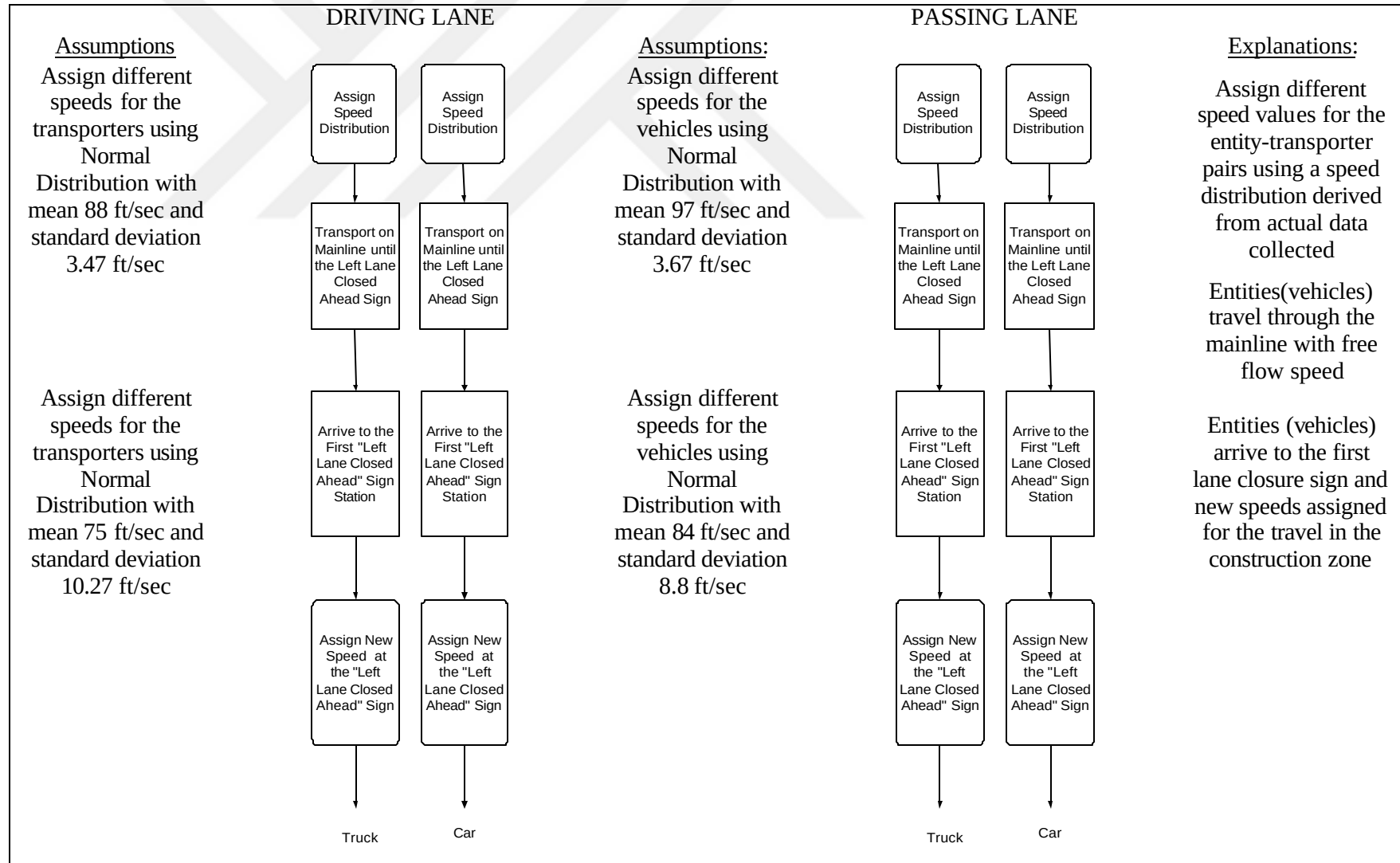
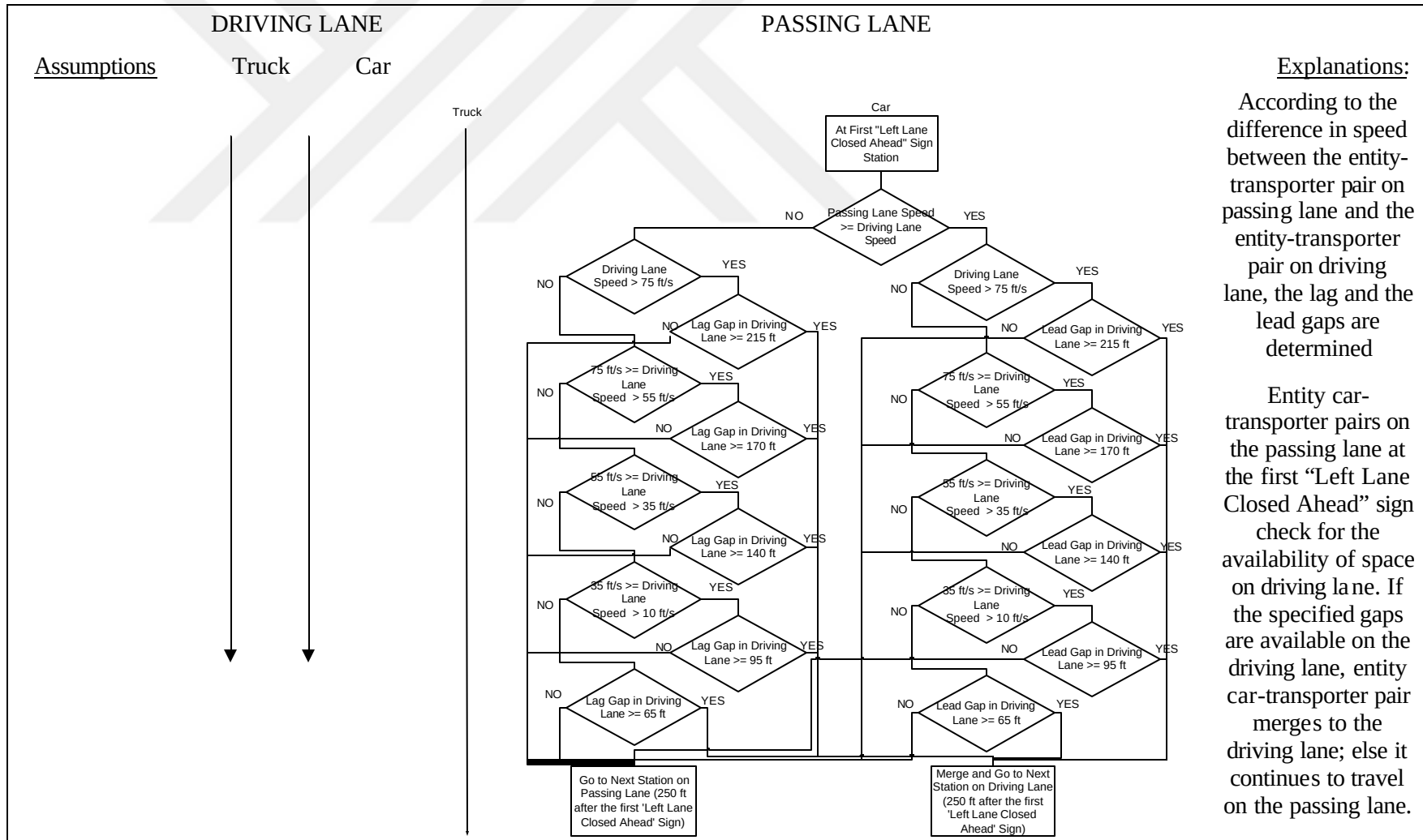


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

Explanations:

According to the difference in speed between the entity-transporter pair on passing lane and the entity-transporter pair on driving lane, the lag and the lead gaps are determined

Entity car-transporter pairs on the passing lane at the first "Left Lane Closed Ahead" sign check for the availability of space on driving lane. If the specified gaps are available on the driving lane, entity car-transporter pair merges to the driving lane; else it continues to travel on the passing lane.

Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

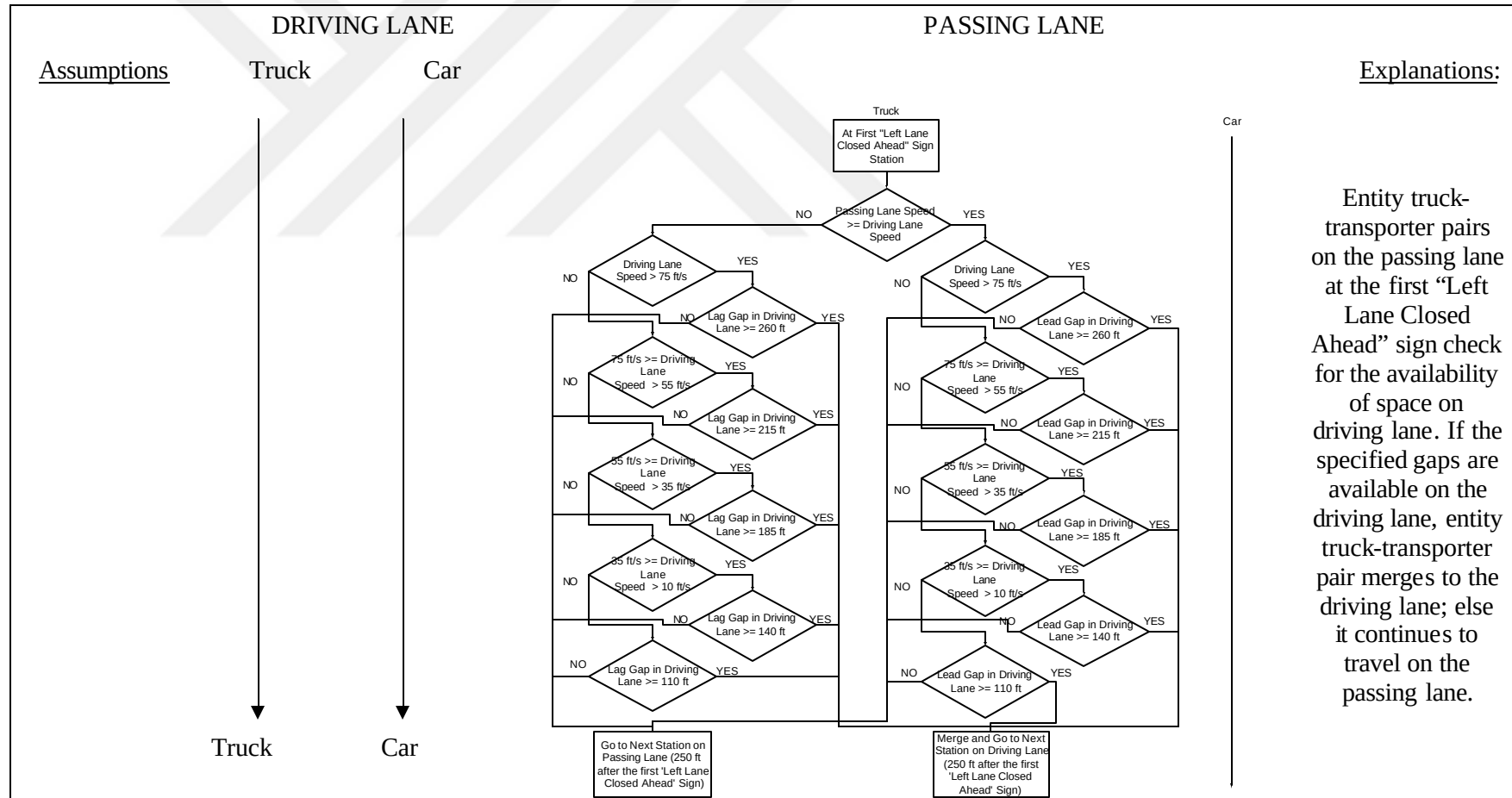


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

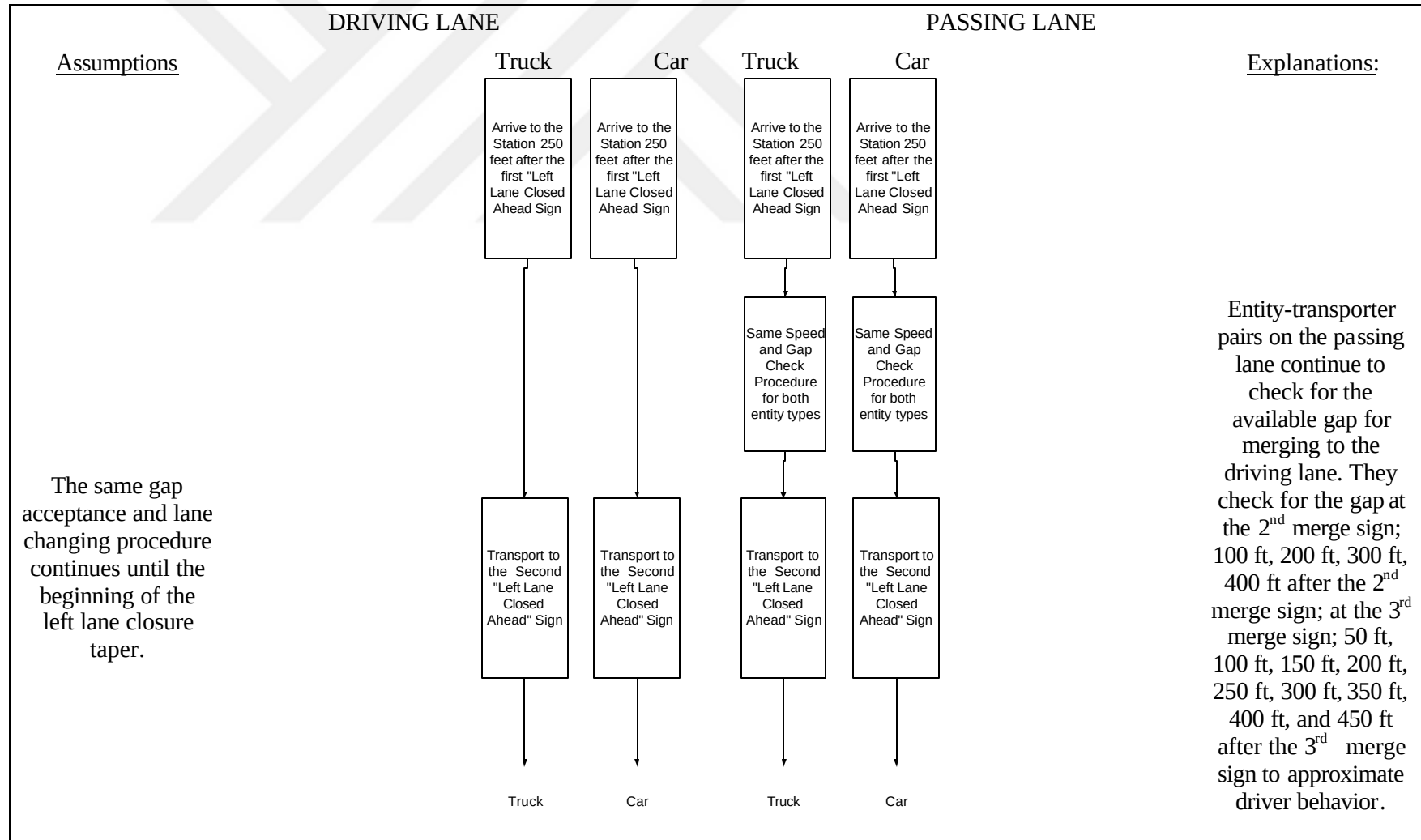


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

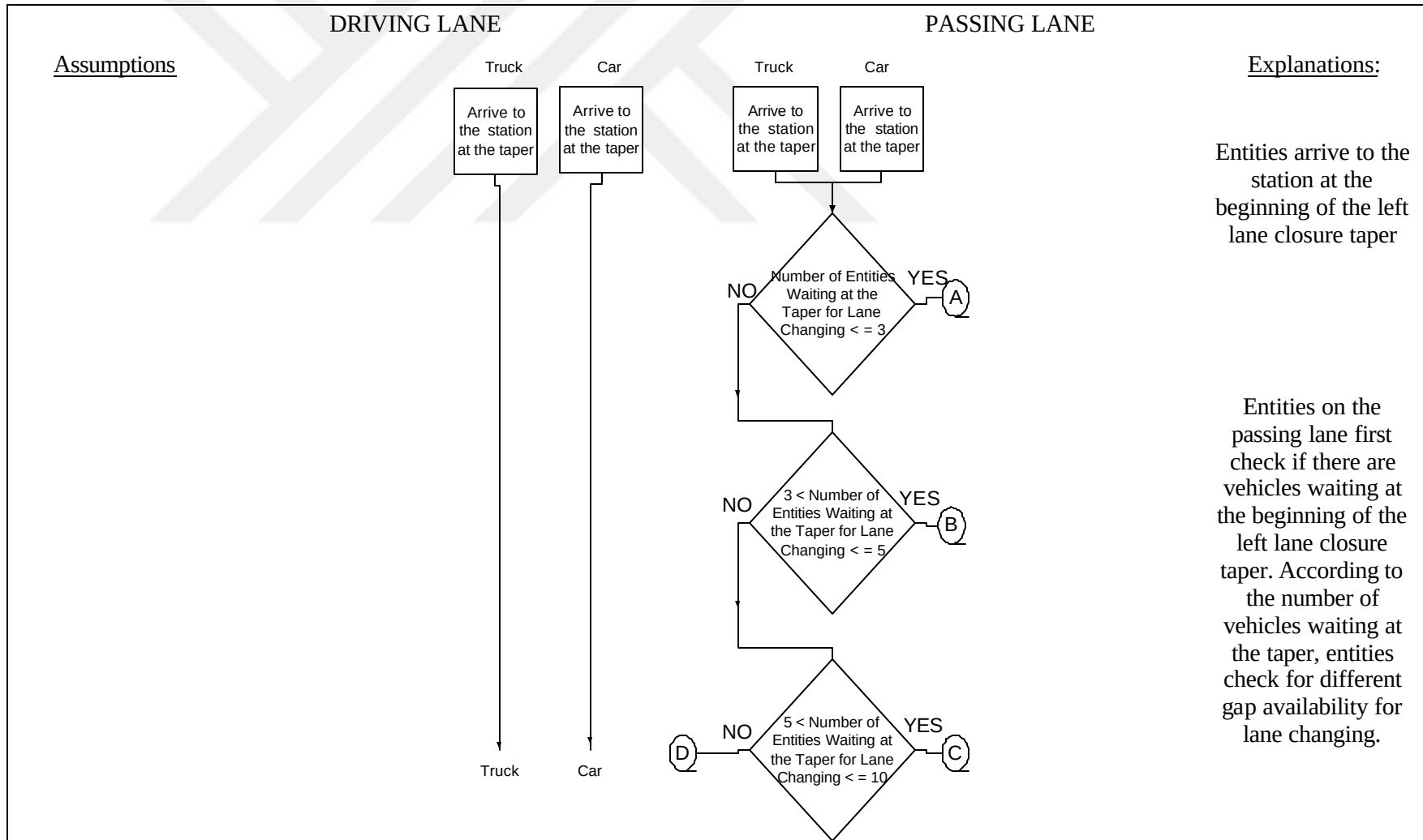


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

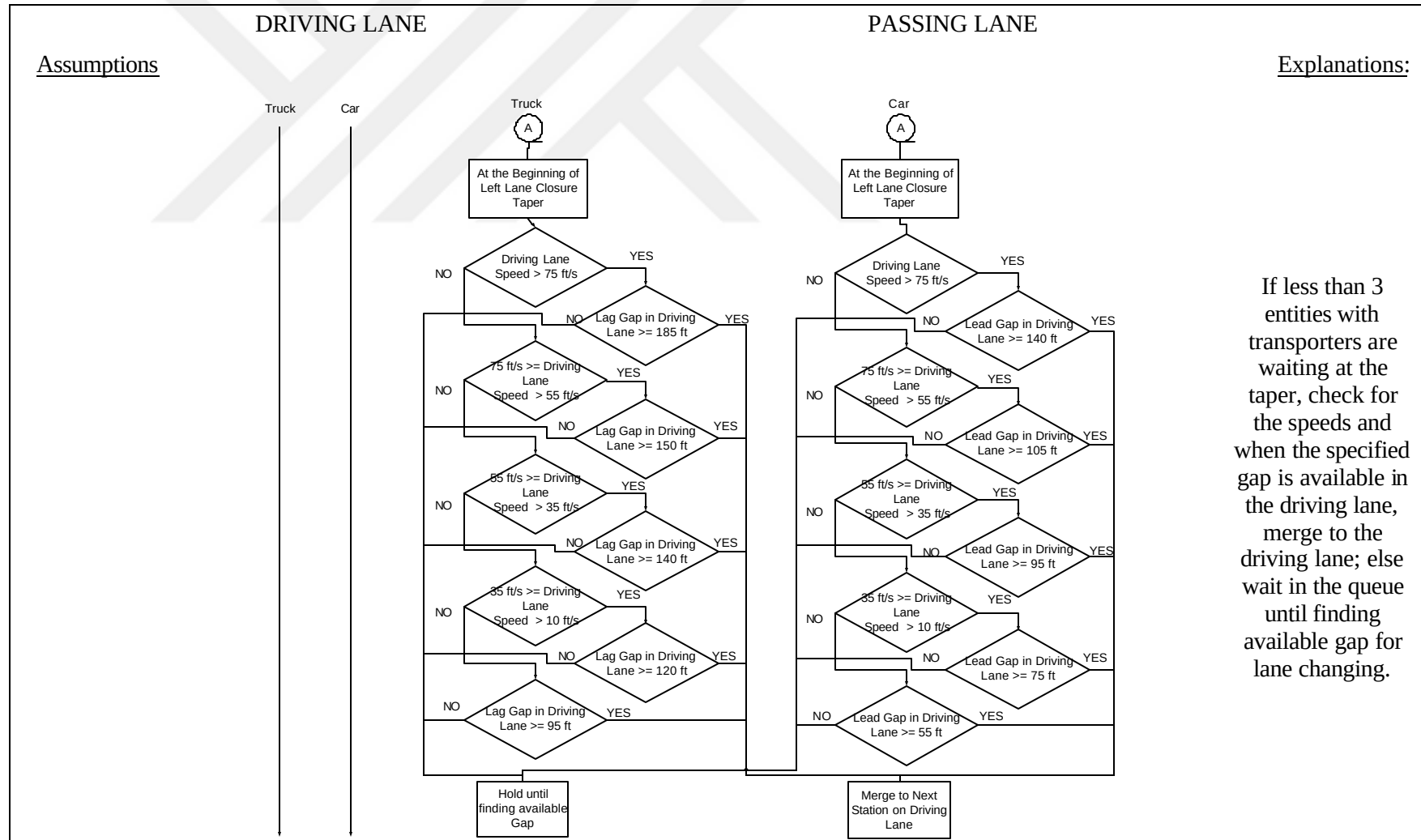


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

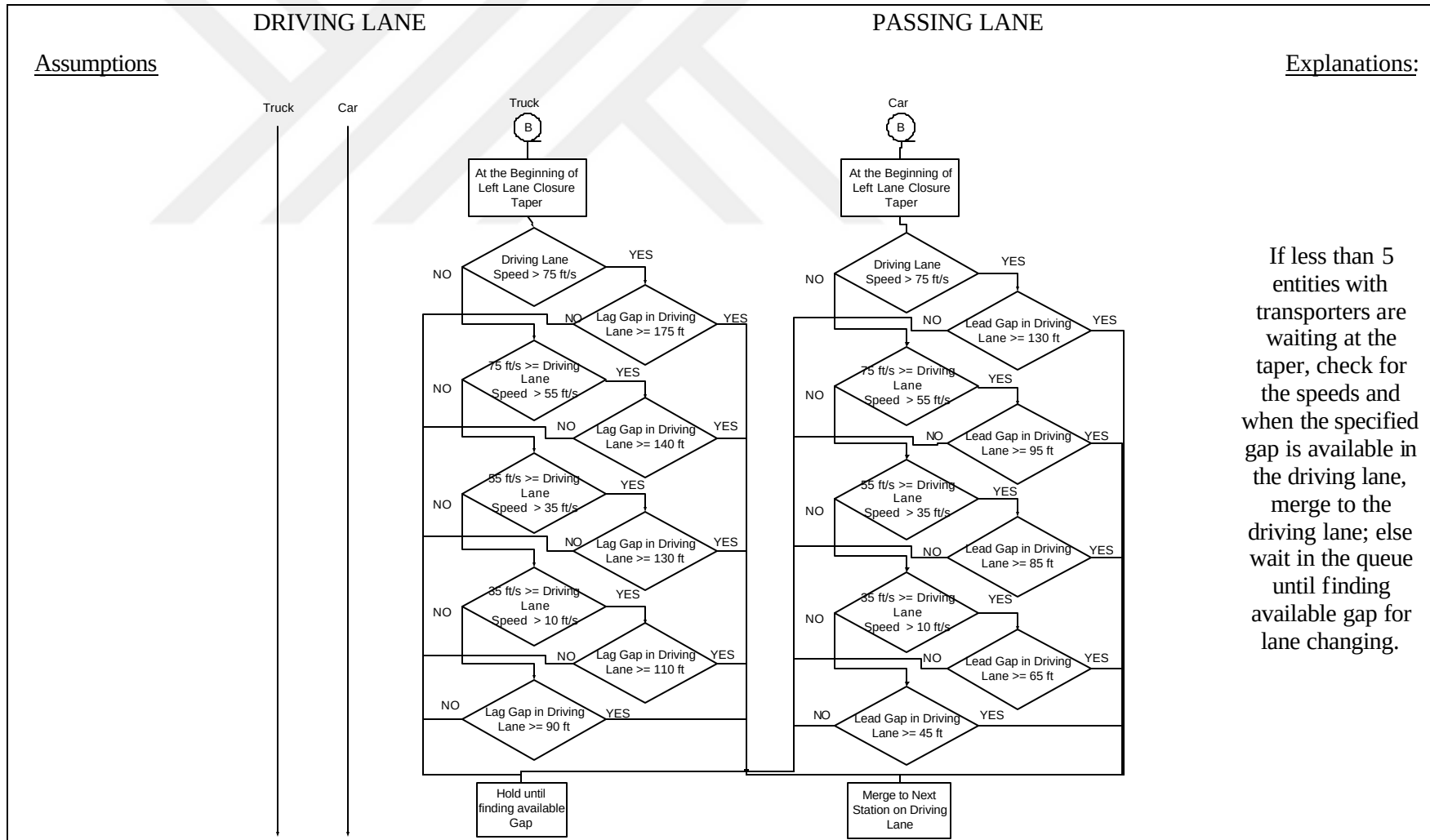


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

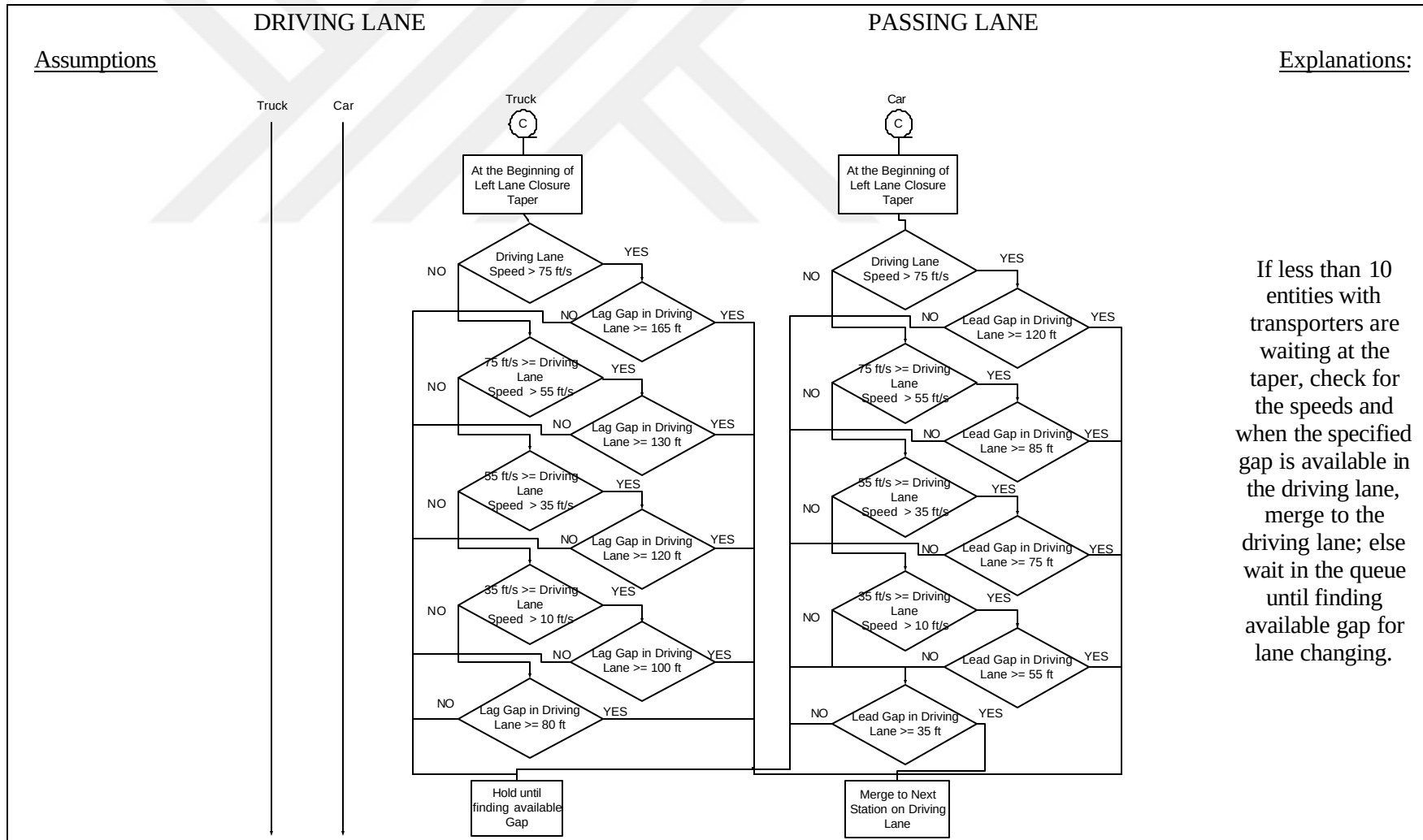


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

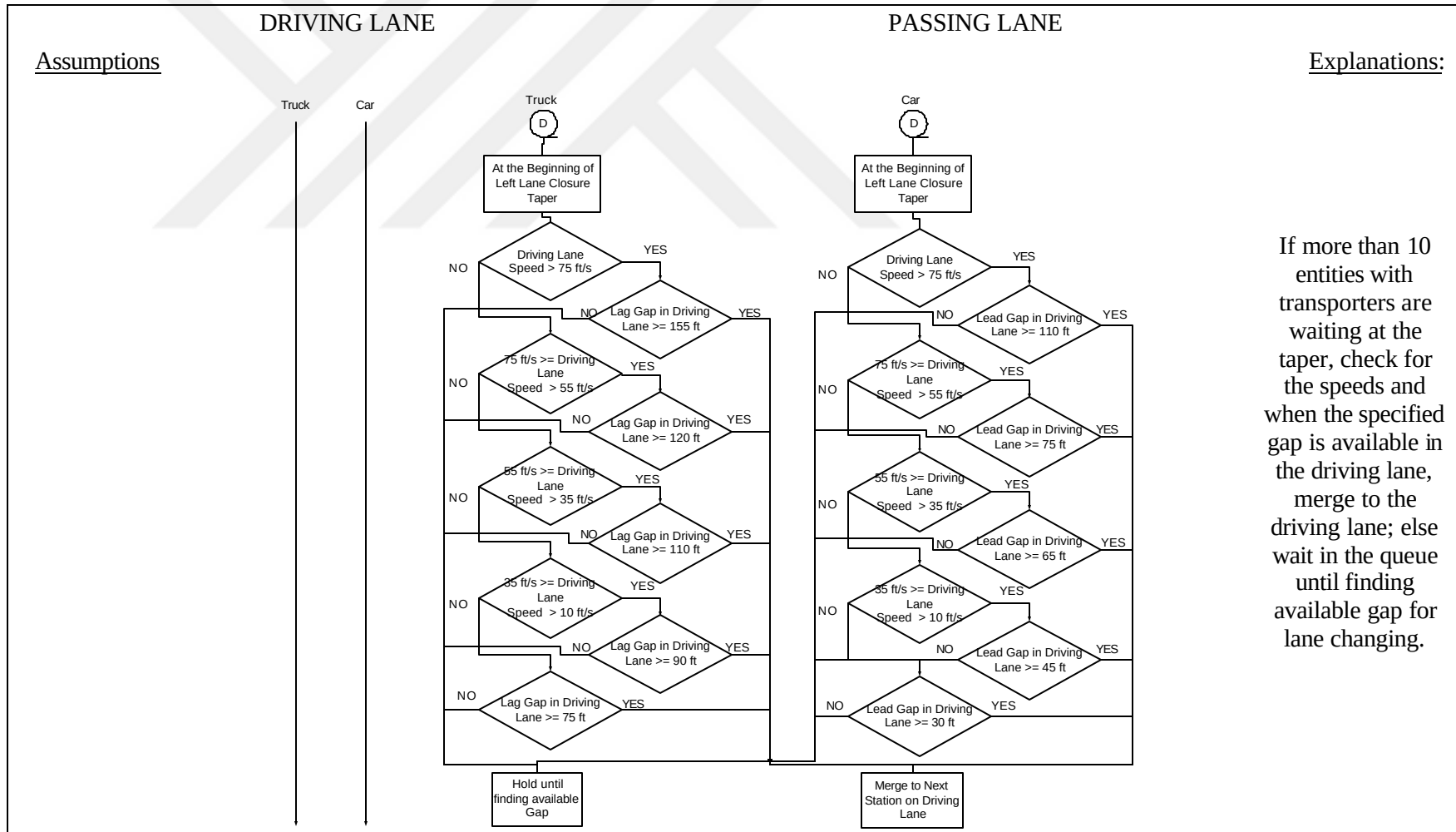


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

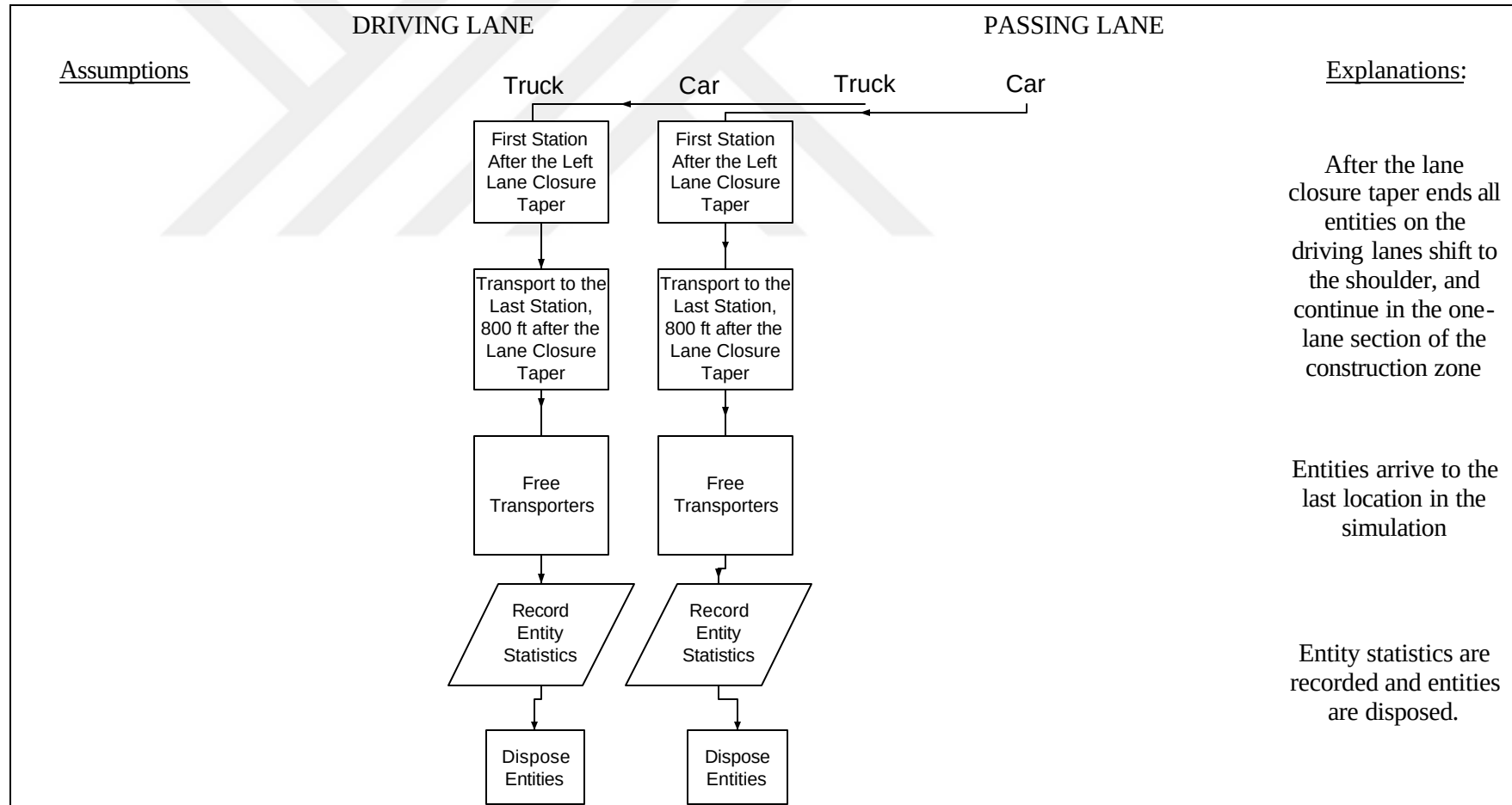


Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)

3.2.2.3 Output Variables

In this study as it was stated in the objectives, delay times and the queue lengths were investigated. This information is the most important outputs of the ARENA simulation model.

The simulation run length was assigned as 24 hours. The model was run for 72 replications in order to get independent and random outputs for the simulation. 900 seconds of warm-up period was also specified in the simulation run parameters.

Statistics for each entity was recorded into a text file by the model. Entity number, entity type (passing lane car, passing lane truck, driving lane car, driving lane car), entity create time, wait time, total times were recorded. In addition, the queue length at the taper and the waiting time at the passing lane closure taper were recorded as the output of the system.

3.2.2.4 Limitations

During the development of the simulation model, some limitations of the ARENA simulation program were observed.

In the ARENA simulation program, some limitations with respect to transporters were observed. After the guided vehicle transporters are freed, released by the entity, they remain at the location where they are freed. When there are many numbers of transporters in the system, the user of the program has to assign greater distances for the locations where the transporters are released. Another limitation observed with respect to transporters was, not being able to specify acceleration and deceleration rates explicitly

for the transporters. Initially variable speed cannot be assigned to the transporters in the program. Velocity of the transporters can be assigned with a probabilistic distribution only after the transporters are requested and active.

In the design of the model, intersections and links were used for simulating road. Intersections were paired with the stations defined in the model. Events occur at the intersections and the vehicle characteristics such as lane changing behavior can only be incorporated at these intersections. In this traffic simulation model in some cases, intersections which are 50 feet apart were created in order to allow the integration of vehicle characteristics in a continuous manner to the system.

Some of the entity output variables were created with the statistics model in order to use the ARENA output analyzer function for the analysis. The output analyzer did not function with the model, resulting in an error that the model size is too big even when it was a research version.

3.3 Description and Design of QuickZone Delay Estimation Program

The QuickZone work zone delay estimation program was developed by The Federal Highway Administration (FHWA) in cooperation with Mitretek Systems [4]. It uses a deterministic queueing model to simulate traffic going through work zones to determine when there would be traffic backups. It is a tool intended for highway engineers to determine how to set up a work zone to minimize traffic disruption [27].

MD-QuickZone Version 1.01 was used for the simulation of the same construction work zone used in the ARENA simulation model. MD-QuickZone Version

1.01 was developed by University of Maryland [29] starting from a regional QuickZone developed by the FHWA.

3.3.1 Inputs of QuickZone Delay Estimation Program

The construction zone described in section 3.2.2.1.1 (page 52) was input to the program. In QuickZone, the layout of the road section to be simulated is entered to the system using nodes and links.

In QuickZone nodes are defined as the beginning and end points of a road section. In the simulation of the I-76 Westbound Work Zone, the road is considered to be composed of 3 nodes. The nodes used in the QuickZone are given in Figure 24. The X and Y values are positions in the coordinate system of each node, given in feet. The X and Y values are derived from the drawing in Figure 13. Traffic flow is in the negative direction, from X=3000 to X=0.

Node Information		
Node Number	X	Y
1	3000.00	1.00
2	2300.00	1.00
3	0.00	1.00

Figure 24: Node Information (X-Horizontal Axis, Y-Vertical Axis)

Links in QuickZone define the road sections. A link joins two nodes. The eight links defined in the I-76 westbound work zone simulation are given in Figure 25. Links also have traffic parameters associated with them.

Link #	A Node	B Node	Lanes	Capacity (VPL)	Length (Miles)	Freeflow Speed (mph)	Jam Density (V/mi/L)	I or O	Type	Position
1	1	2	2	2100	0.1326	65	220	I	M	0
2	2	3	2	2100	0.4356	55	220	I	WZ	0

Figure 25: Link Information

An A node is the beginning point of a link and a B Node is the end point of a link. Thus link 1 connects Node 1 (A Node) to Node 2 (B Node). In the lanes column, number of lanes in the road section is given. Capacity is entered as the number of vehicles per hour per lane (vphpl) and the length of a link road section is given in miles. Jam density, which is the number of standing vehicles that will fit on one lane mile of the road, is also specified. The column headed I or O indicates whether the direction of the link is Inbound or Outbound [28]. All links in the model are inbound. Links are defined as one of six types in QuickZone: Mainline (M), Work zone (WZ), Detour 1 (D1), Detour 2 (D2), Ramp (R) and blank (for links that are none of the other five types). The types of links used in the simulation are entered as the mainline (M) or work zone (WZ). The Position attribute is used for generating a visual representation of the network.

The second input of the QuickZone is the inbound demand pattern. Hourly percentages according to the daily traffic counts are entered into the inbound demand pattern in QuickZone. To calculate these numbers, hourly traffic counts for 08/20/04 Friday were used, the same day as it was used in ARENA simulation model. By dividing each hourly traffic count by the total daily count the percentages for hourly demand patterns were obtained. The traffic counts and calculations of the hourly percentages are

given in Table 24. These percentage values are entered as the hourly demand factors into QuickZone, same demand factors were used for all days of the week Figure 26.

Table 24: Hourly Vehicle Counts and Hourly Demand Factors for 08/20/2004

Time	Number of Vehicles per Hour	Hourly Demand Factors (Number of Vehicles per hour / Number of Vehicles per Day)
0:00	264	1.40%
1:00	207	1.10%
2:00	244	1.29%
3:00	210	1.12%
4:00	276	1.47%
5:00	425	2.26%
6:00	881	4.68%
7:00	1143	6.07%
8:00	1027	5.45%
9:00	956	5.08%
10:00	1101	5.85%
11:00	1117	5.94%
12:00	1153	6.12%
13:00	1092	5.80%
14:00	1262	6.70%
15:00	1249	6.63%
16:00	1233	6.55%
17:00	1121	5.95%
18:00	1008	5.35%
19:00	821	4.36%
20:00	634	3.37%
21:00	602	3.20%
22:00	441	2.34%
23:00	361	1.92%
total	18827	100.0%

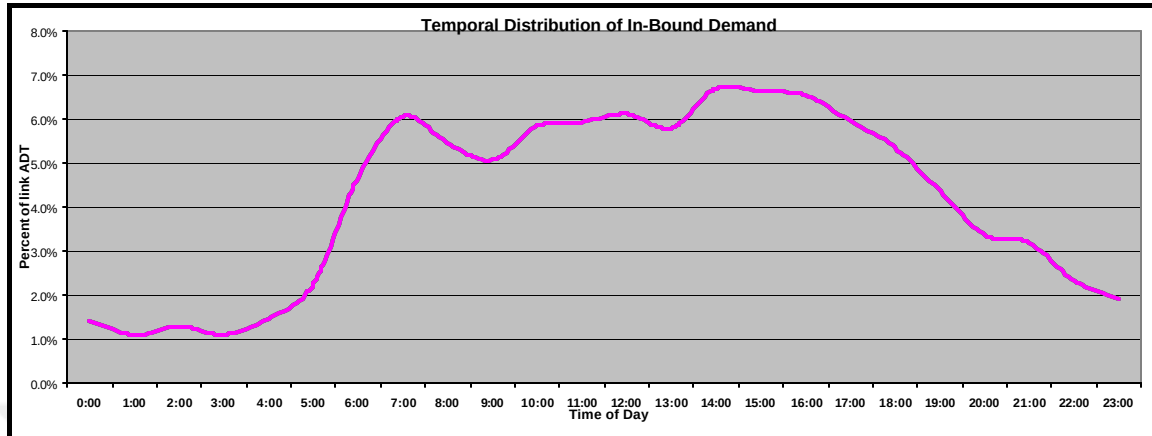


Figure 26: Temporal Distribution of Hourly Inbound Demand on I-76 Westbound based on the Hourly Vehicle Count Data for both Lanes Collected on 08/20/04

QuickZone relies on hourly demand data to conduct its calculations. The demand module is used to generate hourly counts on a link-by-link basis for each day of the week. In the demand module, the daily traffic count for Sunday is entered for all of the links Figure 27. Since there are no entrances or exits within the work zone, all links have the same amount of traffic. In addition the truck percentages data for the day was entered, the daily truck percentage was 27%, and it was assumed to be same for all day. The hourly vehicle counts for all the days of the week and the truck percentages during the day were assumed to be the same in order to be able to compare the results of the simulation with the developed ARENA simulation model.

Link	I or O	AADT
1	I	18827
2	I	18827

Figure 27: Daily Traffic Counts for the Links

Link travel demand volumes may be adjusted by phase based upon the seasonal demand pattern for each month in the year in QuickZone. The seasonal demand pattern is specified as 100% for all of the months, since the simulation project duration was one week during August so values for other months were not used.

In the project information module, the duration and the start and end dates of the project were specified. The project was specified as ending at the completion of the third week in August including 08/20/2004, which was the data used in the simulation.

QuickZone requires at least one phase for a project. The phasing information for the project is entered using the phasing information module. This project is specified as 24-hours a day for one week. In the phasing information module the lane closures and capacity changes are calculated. The Work Zone Link information module under phasing information in QuickZone calculates the capacity decrease using the Highway Capacity Manual 2000 method. When one lane in the work zone is closed, the capacity of the single remaining lane is set to 1600 vehicles per lane per hour. The Economic Analysis and Delay/Cost parameters in the software were not entered for the project because no economic results were needed.

3.3.2 Outputs of QuickZone Delay Estimation Program

QuickZone provides four primary outputs—a delay graph, a travel behavior summary, an amortized delay and construction cost graph, and a summary table. Cost parameters and traveler behavior parameters were not entered. Delay graphs and a summary table were generated as the output of this simulation.

The summary table screen provided data on two key pieces of data relative to the construction project: queue and delay. The table includes the average, total, or maximum value for each construction phase. Three cases can be displayed in summary tables using the QuickZone output options module:

Baseline: Displays the recurring queueing, delay and costs, if any, may occur when there is no work zone.

After: Displays the queueing, delay, and costs associated only with the work zone.

Sum: Displays the combination of the baseline and after queueing, delay, and costs.

The after summary table for the I-76 work zone is given in Table 25.

Table 25: Output Summary Table

QZ I-76 Westbound					
After Data					
Title	Queue-Inbound		Delay-Inbound		
	Weekly	Weekly	Weekly	Weekly	Phase
	Max (mi)	Total (mi)	User Max (min)	Total (VH)	Total (1000 VH)
Phase1	0	0	0	0	0
Phase1-Work1	0	0	0	0	

Queue results include (values depend upon summary table user selection):

- **Weekly Maximum (*Miles*):** Maximum queue experienced within each work zone plan and within the construction phase.
- **Weekly Total (*Miles*):** Sum of the queues for an average week within each work zone plan. Also the weekly average over all the days in the construction phase.

Delay results include (values depend upon summary table user selection):

- **Weekly User Maximum (*Vehicle Hours*):** The maximum delay that occurred during each work zone plan and within each construction phase.
- **Weekly Total (*Vehicle Hours*):** The total weekly delay within each work zone plan. Also, the total among all construction phases over the seven days of the week.
- **Phase Total (*1000 Vehicle Hours*)**—Total delay for the duration of the construction phase.

The project delay summary presents data for each phase with two options. The user has the option of which days to show on the graph. These options include: whole week, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday. The delay graph for whole week is given in Figure 28. As it can be seen no queues or delays were observed for the simulated project.

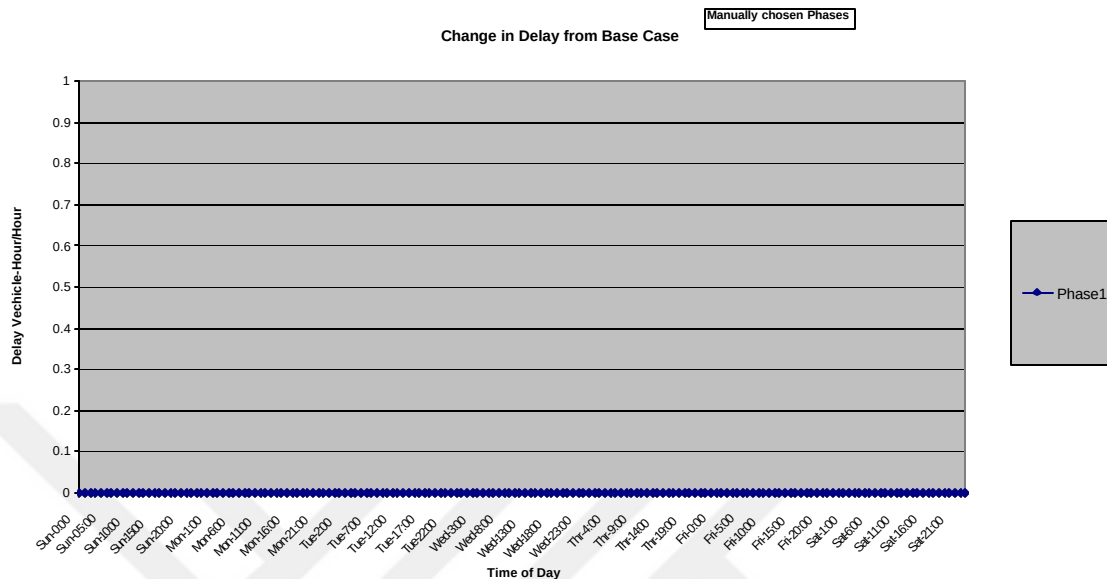


Figure 28: Delay Graph for the Project (Whole Week)

4 RESULTS AND DISCUSSION OF RESULTS

4.1 Analysis and Discussion of ARENA Simulation Results

The outputs of the 72 simulation runs, each for 24 hours were analyzed and compared with the actual traffic data collected in the field. The vehicle count comparisons are given in Table 26.

Table 26: Comparison of Number of Vehicles observed in the Field during Data Collection and the Number of Vehicles Obtained from Simulation Output

	Driving Lane			Passing Lane		
	Actual Data	Simulation Output	Percent Difference (Simulation Output/Actual Data)	Actual Data	Simulation Output	Percent Difference (Simulation Output/Actual Data)
Car	7222	7177	99.377%	6358	5966	93.835%
Truck	4617	4612	99.892%	628	588	93.631%
Total	11840	11789	99.569%	6987	6554	93.803%

The number of vehicles observed showed that the number of vehicles on driving lane is nearly the same with the actual vehicle counts for the average of 72 simulation runs. However there is a 7% difference in the passing lane vehicle counts. The numbers of vehicles observed as the result of the simulation runs for passing lane were less when the averages of 72 replications were compared. In Figure 29 through Figure 35, the difference between the hourly vehicle counts can be observed.

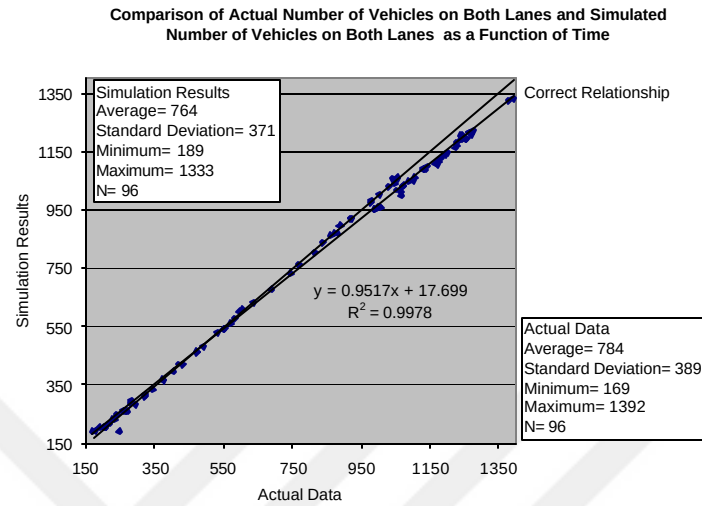


Figure 29: Comparison of Actual Number of Vehicles on Both Lanes and Simulated Number of Vehicles on Both Lanes as a Function of Time

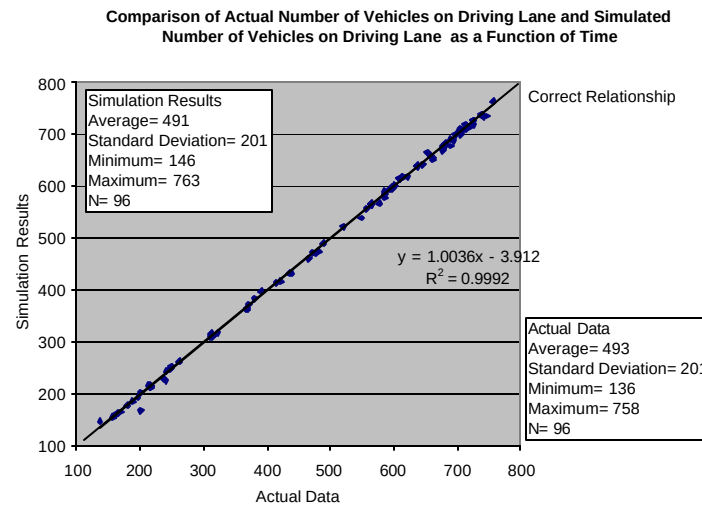


Figure 30: Comparison of Actual Number of Vehicles on Driving Lane and Simulated Number of Vehicles on Driving Lane as a Function of Time

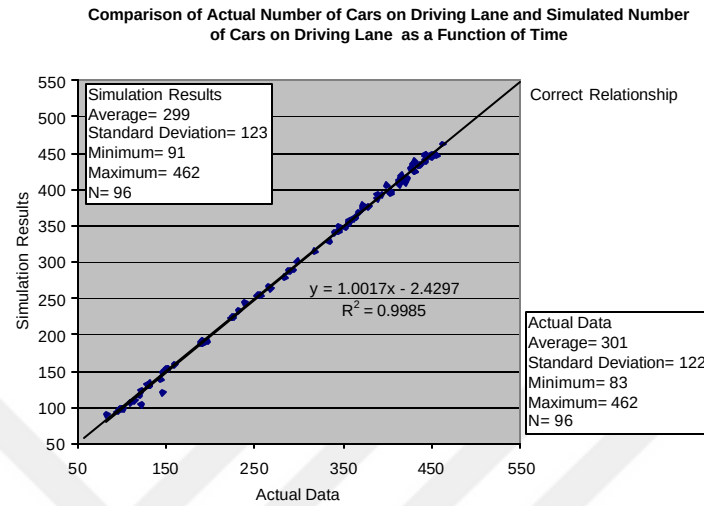


Figure 31: Comparison of Actual Number of Cars on Driving Lane and Simulated Number of Cars on Driving Lane as a Function of Time

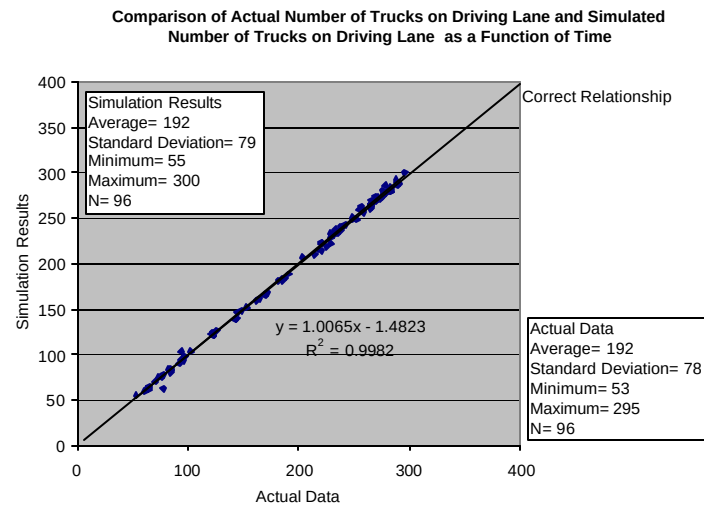


Figure 32: Comparison of Actual Number of Trucks on Driving Lane and Simulated Number of Trucks on Driving Lane as a Function of Time

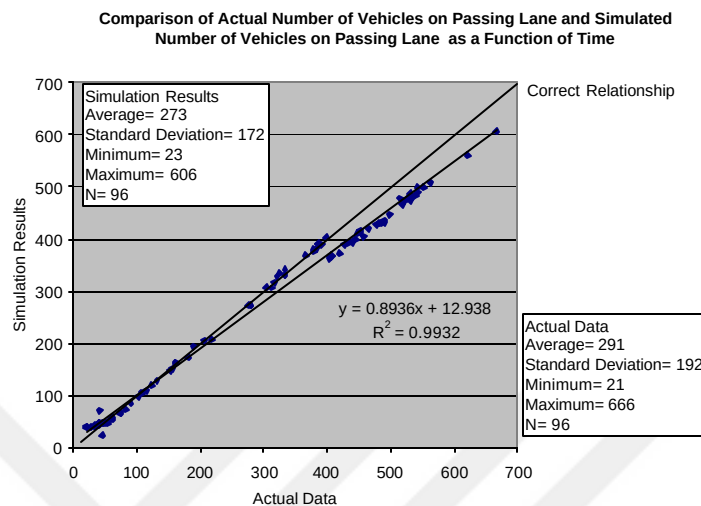


Figure 33: Comparison of Actual Number of Vehicles on Passing Lane and Simulated Number of Vehicles on Passing Lane as a Function of Time

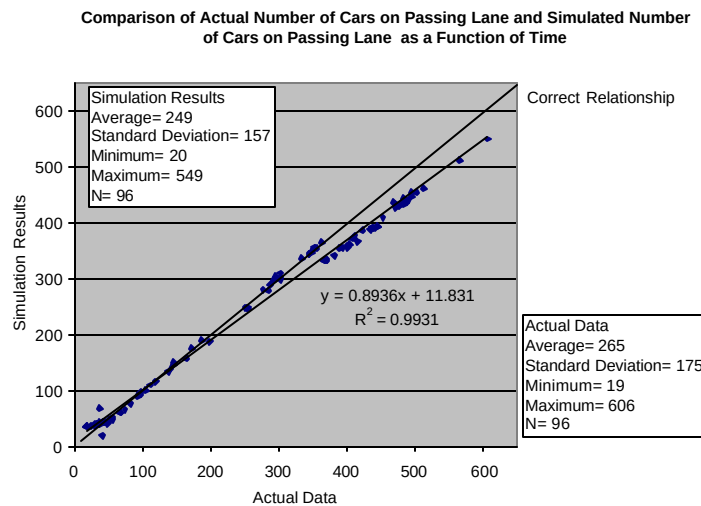


Figure 34: Comparison of Actual Number of Cars on Passing Lane and Simulated Number of Cars on Passing Lane as a Function of Time

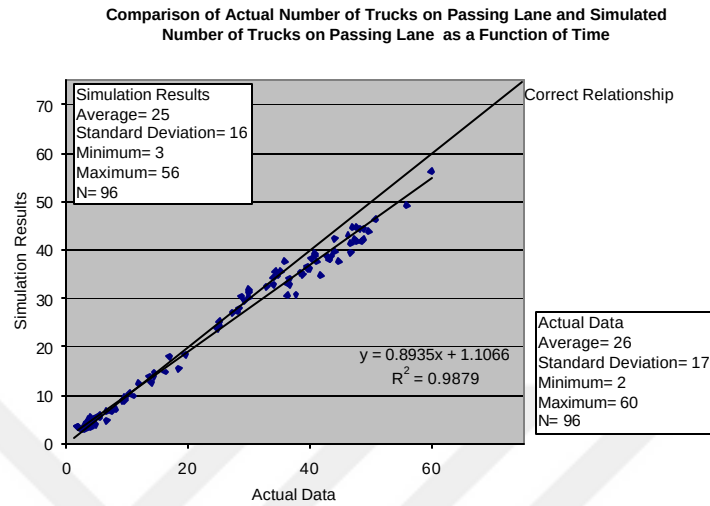


Figure 35: Comparison of Actual Number of Trucks on Passing Lane and Simulated Number of Trucks on Passing Lane as a Function of Time for

The speeds of the vehicles traveling on driving and passing lanes are calculated using the output data generated. The ARENA simulation model gives the transfer times for each lane. The average travel time for all the entities entering the driving lane is calculated as 39.59 seconds, and the average travel time for all the entities entering the passing lane is calculated as 35.26 seconds. Dividing total length simulated which was 3000 feet by these average transfer times gave the overall average speeds for each lane.

$$\text{Average Speed on Driving Lane} = 3000 \text{ ft} / 39.59 \text{ sec} = 75.77 \text{ ft/sec}$$

$$\text{Average Speed on Passing Lane} = 3000 \text{ ft} / 35.26 \text{ sec} = 85.08 \text{ ft/sec}$$

In the model, the speeds were assigned according to a normal distribution using the averages and standard deviations given in Table 15. The speeds assigned for the transporters on the driving lane were 88 feet/sec for 700 ft distance and 75 ft/sec for 2300 ft distance. The speeds assigned for the transporters on the passing lane were 97 ft/sec for 700 ft distance and 84 ft/sec for 2300 ft distance. The overall average speeds for the transporters were calculated as 77.68 ft/sec for driving lane and 86.71 ft/sec for passing lane without the consideration of acceleration and deceleration rates. The average speeds calculated by ARENA are slightly greater than the speeds calculated with the assigned speed averages. This small difference in speed is due to the acceleration and deceleration behavior of the vehicles and the waiting time associated with them.

In Figure 36 the speed flow relationship observed on a freeway is given. In the ARENA simulation results, the average speed for driving lane is calculated as 75.77 ft/sec (83.14 km/h) and the average speed for passing lane is calculated as 85.08 ft/sec (93.36 km/h). It can be seen that the average speed values obtained with ARENA lies in the range specified for the flow rates used in the simulation.

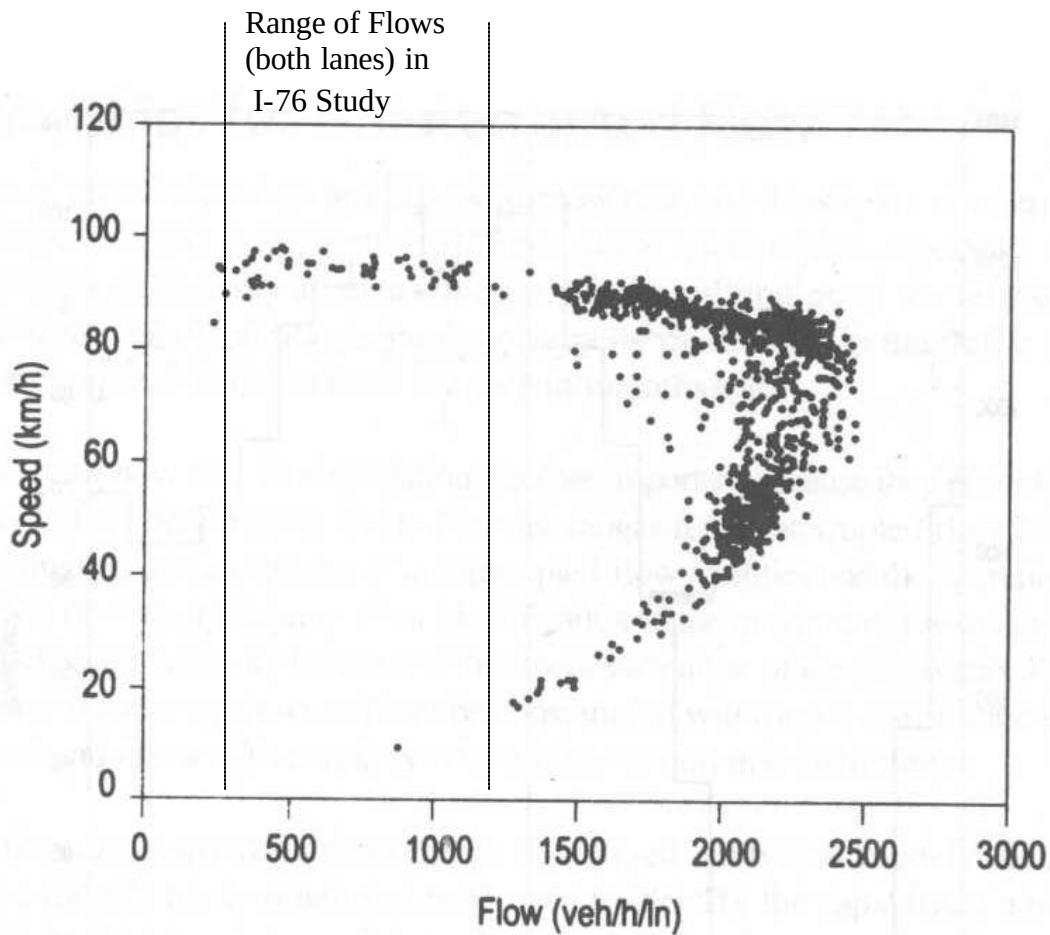


Figure 36: Observed Speed Flow Relationship on a San Diego Freeway [23]

In Figure 37 through Figure 41 the comparison graphs for the driving lane are given. In Figure 37, the average transfer time was plotted with the number of vehicles per hour values for each 15 minutes to observe the effects of traffic volume on average transfer time.

In Figure 38, maximum transfer times for each 15 minute time intervals were plotted with the number of vehicles per hour values for 15 minute intervals. In Figure 39,

minimum transfer times obtained for each 15 minute time interval were plotted with the number of vehicles per hour values for 15 minute intervals.

Waiting time and number of vehicles waiting in queue parameters for the driving lane were observed as zero, as it can be seen in Figure 40 and Figure 41.

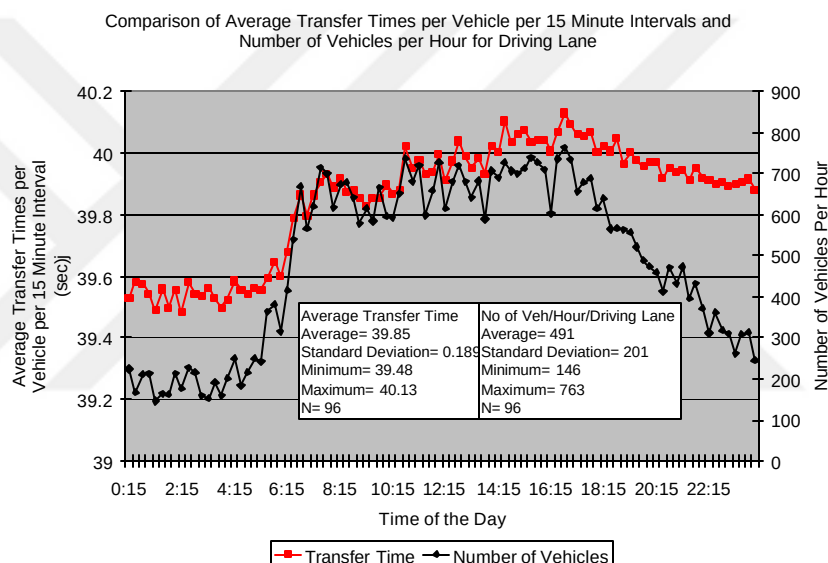


Figure 37: Comparison of Average Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Interval for Driving Lane

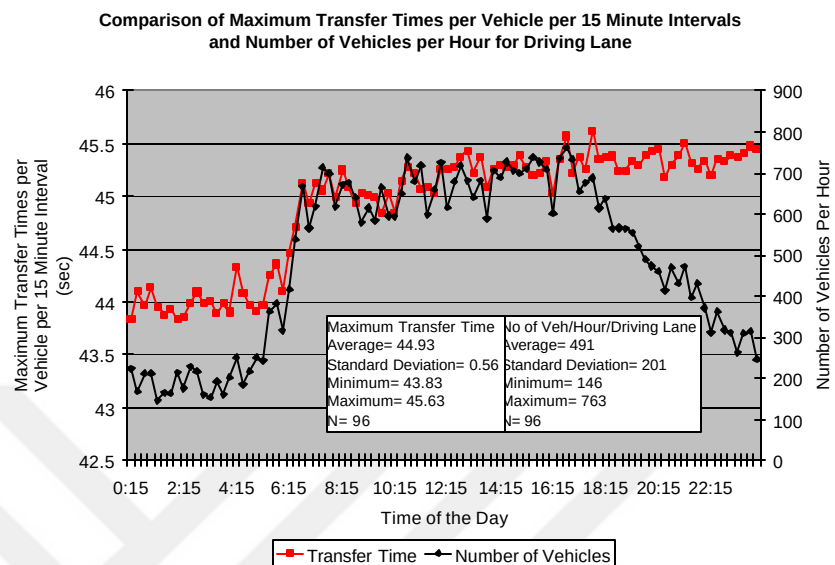


Figure 38: Comparison of Maximum Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Driving Lane

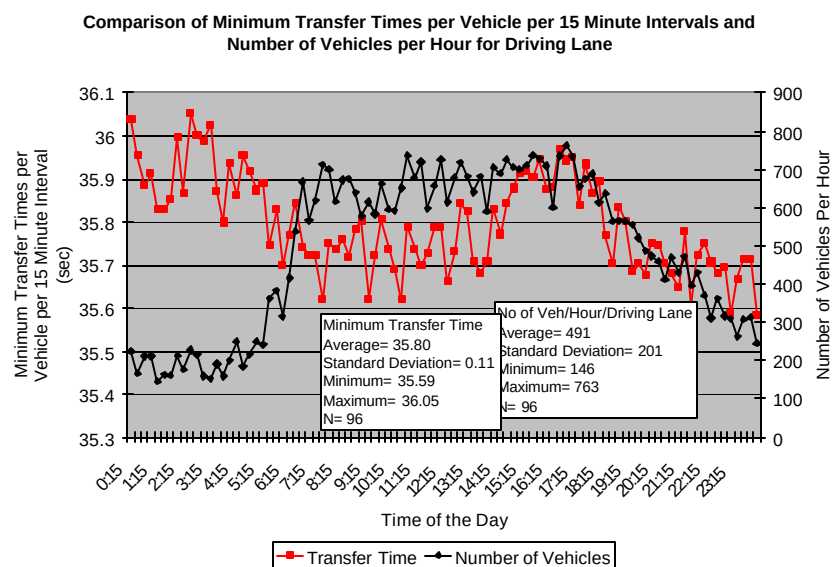


Figure 39: Comparison of Minimum Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Driving Lane

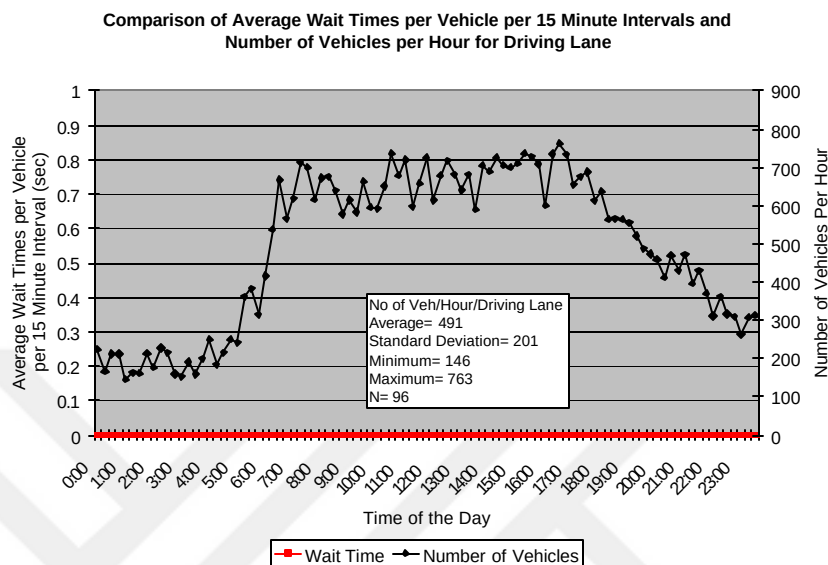


Figure 40: Comparison of Average Wait Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Driving Lane

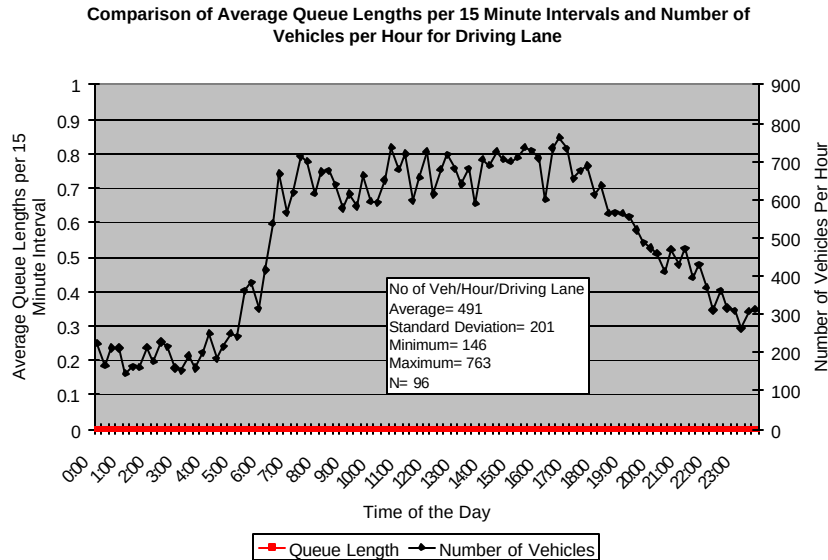


Figure 41: Comparison of Number of Vehicles Waiting in the Queue per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Driving Lane

In Figure 42 through Figure 50 the comparison graphs for the passing lane are given. In Figure 42, the average transfer time was plotted with the number of vehicles per hour values for each 15 minutes to observe the effects of traffic volume on average transfer time. In the graph, slight increments and decrements can be observed depending on the number of vehicles per hour. When the number of vehicles per hour increases, the average travel time also increases slightly.

In Figure 43, maximum transfer times for each 15 minute time interval were plotted with the number of vehicles per hour values for 15 minute intervals. In Figure 44, minimum transfer times obtained for each 15 minute time interval were plotted with the number of vehicles per hour values for 15 minute intervals.

In Figure 45, Figure 46, and Figure 47 average waiting time, maximum waiting time, and minimum waiting time were plotted with the number of vehicles per hour values for each 15 minute intervals.

In Figure 48, Figure 49, and Figure 50 average number of vehicles waiting in queue, maximum number of vehicles waiting in queue, and minimum number of vehicles waiting in queue were plotted with the number of vehicles per hour values for each 15 minute intervals.

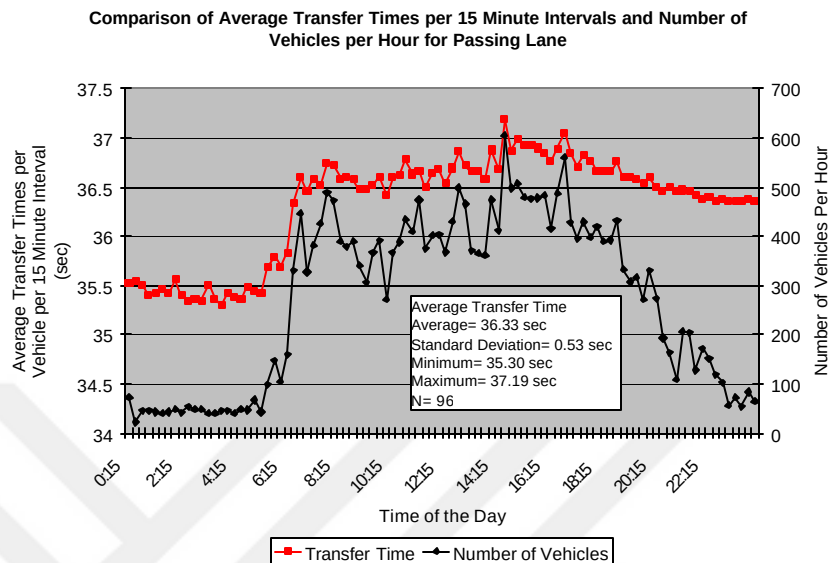


Figure 42: Comparison of Average Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

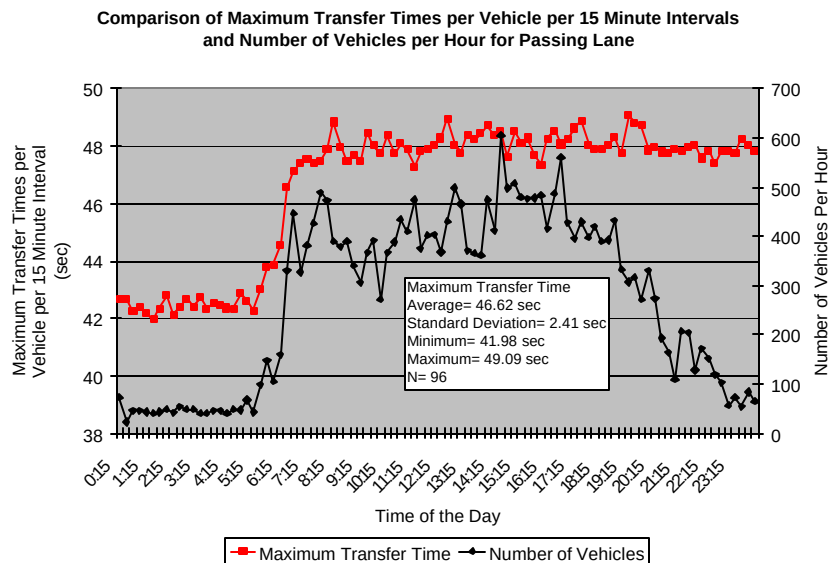


Figure 43: Comparison of Maximum Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

Comparison of Minimum Transfer Times per Vehicle per 15 Minute Intervals and Number of Vehicles per Hour for Passing Lane

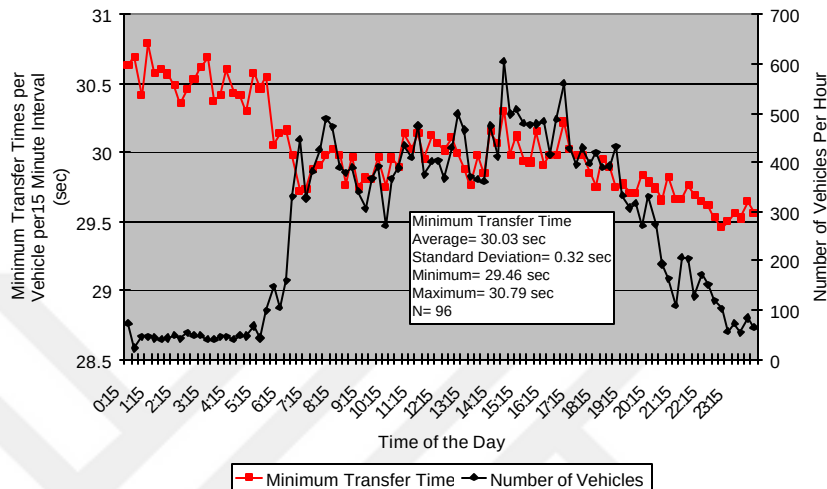


Figure 44: Comparison of Minimum Transfer Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

Comparison of Average Wait Times per Vehicle per 15 Minute Intervals and Number of Vehicles per Hour for Passing Lane

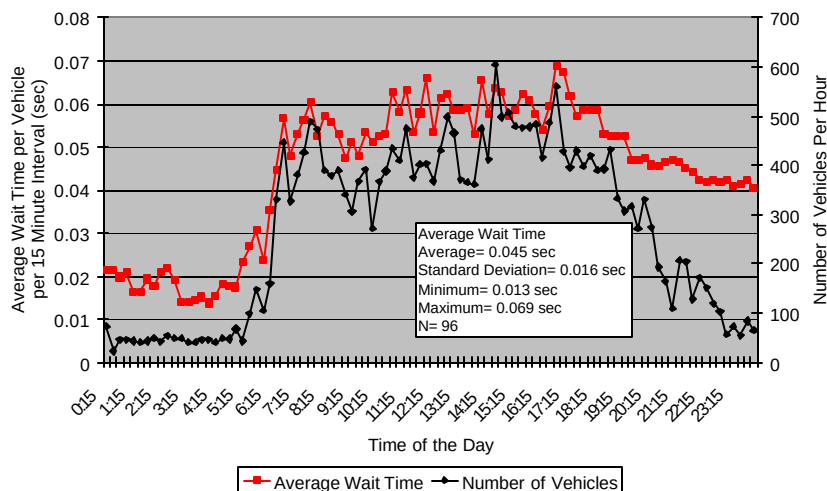


Figure 45: Comparison of Average Waiting Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

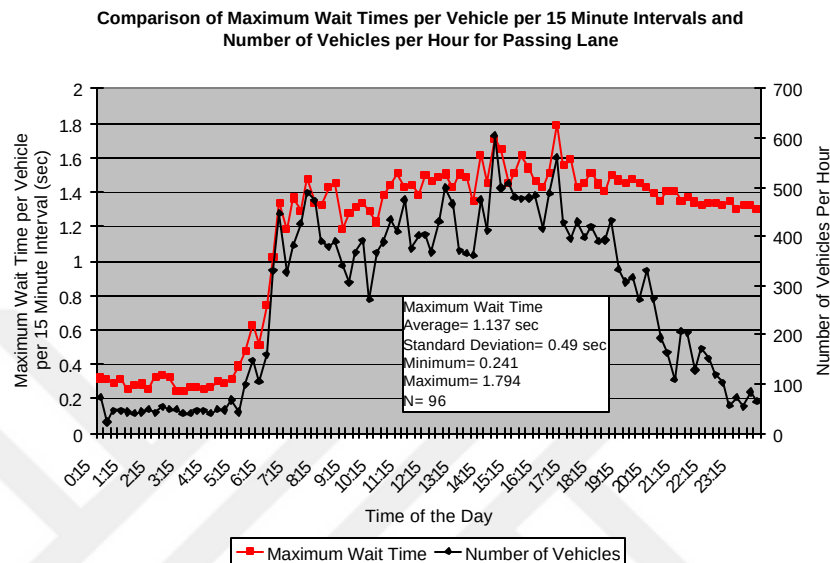


Figure 46: Comparison of Maximum Waiting Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

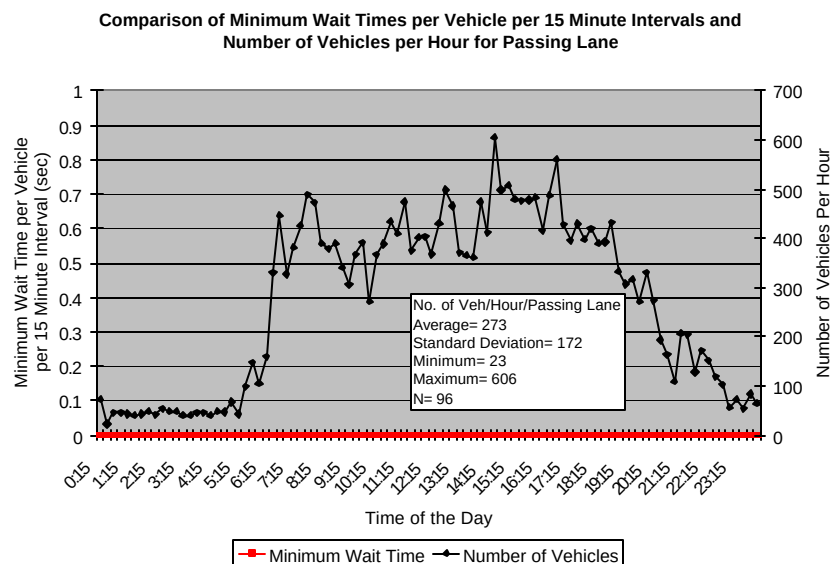


Figure 47: Comparison of Minimum Waiting Times per Vehicle per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

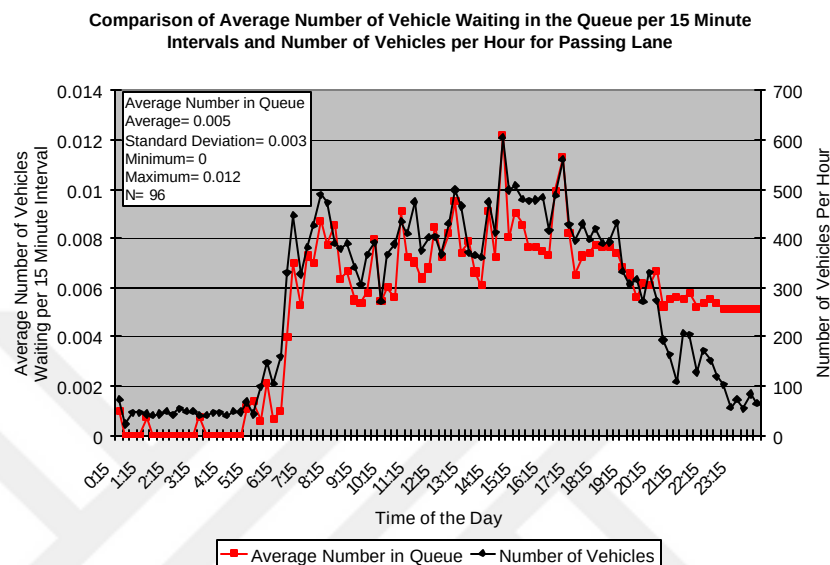


Figure 48: Comparison of Average Number of Vehicle Waiting in Queue per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

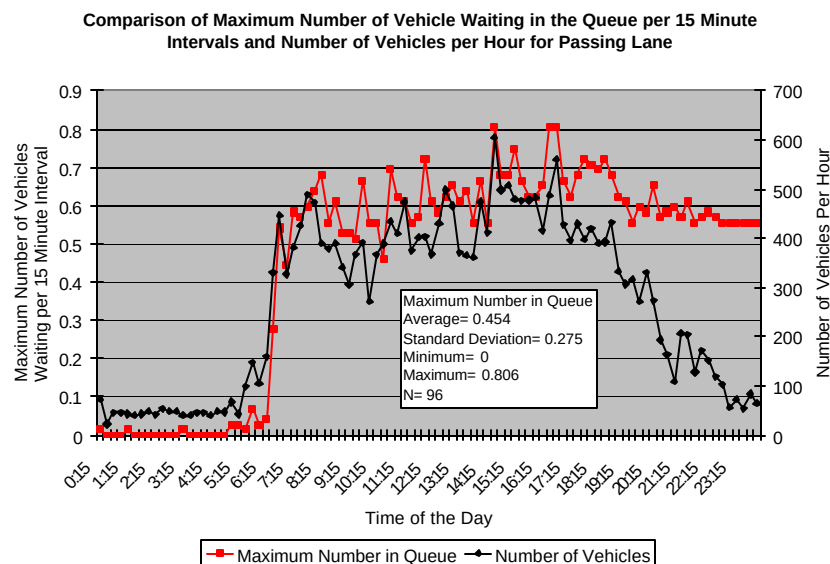


Figure 49: Comparison of Maximum Number of Vehicle Waiting in Queue per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

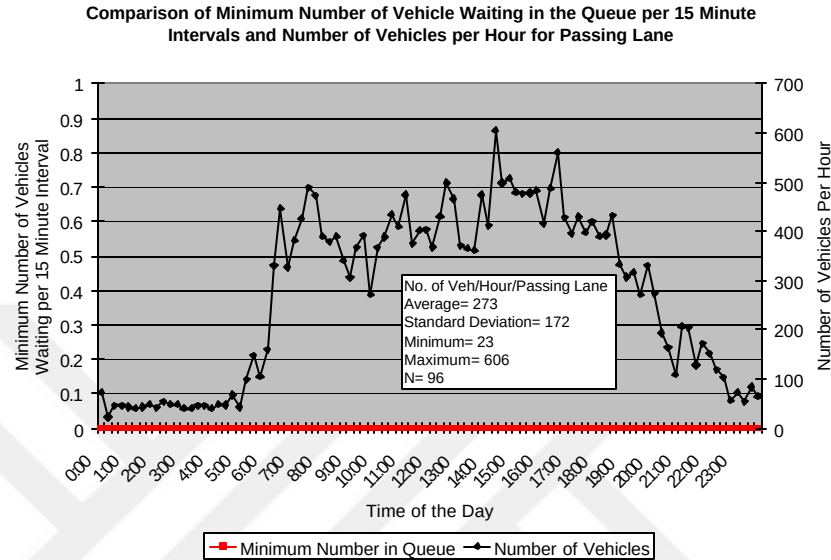


Figure 50: Comparison of Minimum Number of Vehicle Waiting in Queue per 15 Minute Interval and Number of Vehicles per Hour per 15 Minute Intervals for Passing Lane

4.2 Comparison of ARENA with QuickZone

In the ARENA simulation model, probabilistic inter-arrival time distributions were used to model traffic. Driving lane and passing lane had different inter-arrival time distributions at different time intervals. The results for the simulation of the construction work zone traffic were obtained for both lanes separately, which helped us to determine the difference between the output parameters for these two lanes. The analysis of the results of ARENA simulation model showed that there was no actual queue formation at the construction zone simulated. There was at the most a very slight increase in the

average travel times through the construction zone according to the number of vehicles per hour increases both for driving lane and passing lane.

The results of the QuickZone delay estimation program also showed that there were no queues formed and delays observed at the construction work zone simulated.

The results of both simulation models provided similar outputs, showing that there were no queues and delays observed at the I-76 Westbound Construction zone.

4.3 Discussion of Results

The data collected in the construction work zone was used for the modeling of the simulation programs in this study. The use of the actual data provided more realistic results for the simulation outputs, both for developed ARENA model and QuickZone model.

In the ARENA simulation model, probabilistic inter-arrival time distributions were used for different time intervals and the two lanes were simulated separately. Outputs of the ARENA simulation model were obtained both for driving lane and passing lane.

The ARENA simulation model provided the desired outputs, such as queue length, transfer time, and waiting time.

The animation feature of the ARENA simulation program was used. Animation showed that traffic and merging behavior reflect real world conditions observed in the field.

In the outputs of the ARENA simulation program, it was observed that the number of vehicles generated as a function of time of the day is very close to the actual observed data. Therefore the IAT distributions developed appears to be correct.

The developed ARENA simulation model could not be fully evaluated over a wide range of traffic volumes since the actual traffic volumes which were observed and collected at I-76 Westbound were not high enough to cause queues in the lane reduction area. In their study, Maze and Kamyab [2] state that traffic volumes less than 700 vehicles/hour (total for both lanes) do not result in any queues at the taper.

It is expected that the ARENA simulation model developed for the I-76 westbound construction work zone example will be very useful for future construction zone modeling efforts.

5 CONCLUSIONS

In this study, a simulation model for a construction work zone where the number of lanes was reduced was developed using the ARENA simulation program. The cars and the trucks in this model were represented by entity – transporter pairs.

However, the model has a number of limitations and therefore does not represent the lane reduction situation (especially the acceleration and deceleration dynamics) accurately. The vehicles on the driving lane do not decelerate when there is a queue formed at the lane closure taper.

The simulation run time of the model in its present form also takes a considerable time since the time compression factor is about 1:62, in other words 62 real time seconds are simulated in 1 second. It roughly takes 30 minutes on a PC with 2.8 GHz processor to simulate 24 hours of 2 lane car and truck traffic.

Unfortunately, the lane reduction traffic situation for which the data was collected in the real world does not appear to produce queues or delay times, thus the developed model has not been tested when considerable queues or delay times due to lane closures are present.

No queues were observed during the data collection over the three day period, therefore the model output appears to be correct for traffic situations and traffic volumes observed. Similar non-queue observations were stated by Maze and Kamyab [2] for traffic volumes of 700 vehicles/hour (for both lanes) or less.

In order to validate the ARENA simulation model, further analysis where there are 2 lane traffic situations with a lane reduction to one lane and higher traffic volumes resulting in queues at the taper will need to be studied in the future.

In addition, the model needs to be refined to include an algorithm for decreasing the speeds of the vehicles in the driving lane when there are vehicles waiting at the lane closure taper. The continuous modeling option in the ARENA simulation program can also be explored and, if possible, included in the model. The use of the templates option rather than extensive and repetitive groups of if statements could be explored and if possible included in the model.

Further work will be required to model construction work zone situations where there are entrance and exit ramps in the construction work zone. The model will need to be refined to be more adaptable for different construction work zones with different sign locations indicating that there is a lane reduction ahead. The model could be extended to include the percentages of vehicles and speeds of the vehicles varying according to the time of the day. The model could be made more user friendly so that a minimum preparation work is required to run a particular simulation.

6 REFERENCES

- [1] Balci, Osman, Guidelines for Successful Simulation Studies, *Proceedings of the 1990 Winter Simulation Conference*, 1990, pp. 25-32.
- [2] Maze, Tom and Kamyab, Ali, Work Zone Simulation Model: Companion Report for Traffic Measurement Strategies for Merge Areas in Rural Interstate Work Zones, Center for Transportation Research and Education Management Project 97-12, September 1999.
- [3] Gross, Donald and Harris, Carl M., Fundamentals of Queueing Theory, John Wiley & Sons, Inc., Second Edition, 1985, pp. 13-35.
- [4] QuickZone, Turner-Fairbank Highway Research Center, <http://www.tfhrc.gov/its/quickzon.htm>, accessed on 10/12/2004.
- [5] Al-Kaisy, A., Stewart, J.A., and Van Aerde, M., Simulation Approach for Examining Capacity and Operational Performance at Freeway Diverge Areas, *Canadian Journal of Civil Engineering*, Vol.26, No.6, December 1999, pp.760-770.
- [6] Benekohal, Rahim F. and Abu-Lebdeh, Ghassan, Variability Analysis of Traffic Simulation Outputs: Practical Approach for TRAF-NETSIM, *Transportation Research Record*, No.1457, 1994, pp.198-207.
- [7] Bloomberg, Loren and Dale, Jim, Comparison of VISSIM and CORSIM Traffic Simulation Models on a Congested Network, *Transportation Research Record*, No.1727, 2000, pp.52-60.

- [8] Makigami, Yasuji, Nakanishi, Tsunehiko, and Seill, Kim, Simulation Model Applied to Japanese Expressway, *Journal of Transportation Engineering*, Vol.110, No.1, January 1984, pp.94-111.
- [9] Memmott, Jeffrey L. and Dudek, Conrad L., Queue and User Cost Evaluation of Work Zones (QUEWZ), *Transportation Research Record*, No.979, 1984, pp.12-19.
- [10] Morales, Juan M. and Paniati, Jeffrey F., Two-Lane Traffic Simulation- A Field Evaluation of Roadsim, *Public Roads*, Vol.49, No.3, December 1985, pp.95-104.
- [11] Nakanishi, Toshio, Yikai, Kunio, Satoh, Jun-ichi, Miyoshi, Isao, Satoh, Akira, and Takahashi, Michiya, Development of a Road Traffic Simulation System in Broad Areas, *Mathematics and Computers in Simulation*, Vol.39, No.3-4, November 1995, pp.207-212.
- [12] Robles, Juan and Janson, Bruce N., Dynamic Traffic Modeling of the I-25/HOV Corridor Southeast of Denver, *Transportation Research Record*, No.1516, 1995, pp.48-60.
- [13] Schonfeld, Paul, and Chien, Steven I., Optimal Work Zone Lengths for Two-Lane Highways, *Journal of Transportation Engineering*, Vol.125, No.1, January-February 1999, pp.21-29.
- [14] Benz, Robert J., Fenno, David W., and Voigt, Anthony P., Accelerating Major Freeway Reconstruction Process: The Hudson Experience, *Transportation Research Record*, No.1632, September 1998, pp.59-67.

- [15] Rao, Lei and Owen, Larry, Validation of High-Fidelity Traffic Simulation Models, *Transportation Research Record*, No.1710, 2000, pp.69-78.
- [16] Roupail, Nagul M. and Tiwari, Geetam, Flow Characteristics at Freeway Lane Closures, *Transportation Research Record*, No.1035, 1985, pp.50-58.
- [17] Carr, Robert I., Construction Congestion Cost (CO^3) Basic Model, *Journal of Construction Engineering and Management*, March/April 2000, pp.105-113.
- [18] Carter, Mark, Rakha, Hesham, and Van Aerde, Michel, Variability of Traffic-Flow Measures Across Freeway Lanes, *Canadian Journal of Transportation Engineering*, Vol.26, No.3, 1999, pp.270-281.
- [19] Chronopoulos, Anthony Theodore, and Wang Gang, Traffic Flow Simulation Through Parallel Processing, *Transportation Research Record*, No.1566, November 1996, pp.31-38.
- [20] Dudek, Conrad L., Richards, Stephen H., and Buffington, Jesse L., Some Effects of Traffic Control on Four-Lane Divided Highways, *Transportation Research Record*, No.1086, 1986, pp.20-30.
- [21] Lemessi, Marco, An SLX-Based Microsimulation Model for a Two-Lane Road Section, *Proceedings of the 2001 Winter Simulation Conference*, 2001.
- [22] Zwahlen, Helmut T., Russ, Andrew, Oner, Erdinc, and Parthasarathy, Meghna, Evaluation of Microwave Radar Trailers for Nonintrusive Traffic Measurements, prepared for Presentation at the 84th Annual Meeting of the Transportation Research Board, January 9-13, 2005, National Academy of Sciences, Washington, DC.

- [23] Pline, James L., Traffic Engineering Handbook, Fifth Edition, Institute of Transportation Engineers, 1999, pp.62-68.
- [24] May, Adolf D., Traffic Flow Fundamentals, Prentice Hall, Inc., 1990, pp.95-108.
- [25] ARENA Basic Edition User's Guide, Doc ID ARENAB-UM001C-EN-P.
- [26] Pegden, Dennis C., Shannon, Robert E., and Sadowski, Randall P., Introduction to Simulation Using SIMAN, Second Edition, McGraw-Hill, Inc., 1995.
- [27] Zwahlen, Helmut T., Oner, Erdinc, and Russ, Andrew, I76 Westbound Work Zone Simulation Using QuickZone (Maryland version 1.01) Delay Estimation Program, Rootstown and Edinburg Townships, Portage County, report prepared for Ohio Department of Transportation, Ohio University, September 30, 2004.
- [28] QuickZone Delay Estimation Program User Guide, Version 1.01, Mitretek Systems, February 2002.
- [29] Supplementary User's Manual for MD-QuickZone, MD State Highway Administration, Office of Traffic and Safety, August 2002.
- [30] Zwahlen, Helmut T., Improved Work Zone Design Guidelines and Enhanced Model of Travel Delays in Work Zones: Phase I: Portability and Scalability of Inter-arrival and Service Time Probability Distribution Functions for Different Locations in Ohio and the Establishment of Improved Work Zone Design Guidelines, Ohio Research Institute for Transportation and the Environment, Ohio University, Athens, May 2002.
- [31] Thomas Schnell, et al., Evaluation of Traffic Flow Analysis Tools Applied To Work Zones Based on Flow Data Collected In the Field, Operator Performance

Laboratory & Center for Computer Aided Design, The University of Iowa, Iowa City, February 2001.

- [32] Kanaris, A., E. B. Kosmatopoulos, and P. A. Ioannou., Strategies and Spacing Requirements for Lane Changing and Merging in Automated Highway Systems, *IEEE Transactions on Vehicular Technology*, No.50.6, 2001, pp. 1568-581.
- [33] Rakha, Hesham and Crowther, Brent., Comparison of Greenshields, Pipes, and Van Aerde Car-following and Traffic Stream Models, *Transportation Research Record*, No. 1802, 2002, pp. 248-62.

**Appendix A. Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and
Standard Deviations for I-76 Westbound Daytime Driving/Passing Lane**



Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/22/2004	7:00-7:15	46	20.90	19.29	1.41	74.83	66	5.16	56	75	184	184
8/22/2004	6:45-7:00	47	17.81	14.16	1.09	65.72	72	3.60	66	78	188	188
8/22/2004	7:15-7:30	59	15.31	14.99	1.18	77.47	67	3.09	59	74	236	237
8/22/2004	7:45-8:00	65	13.72	12.55	1.05	53.01	69	4.01	61	79	260	261
8/22/2004	7:30-7:45	67	13.45	11.78	1.67	51.37	67	4.47	59	75	268	269
8/22/2004	8:00-8:15	67	13.56	11.05	0.96	54.74	65	2.98	58	73	268	269
8/21/2004	6:45-7:00	72	12.03	8.60	0.65	47.06	59	3.75	52	67	288	289
8/21/2004	7:00-7:15	78	11.78	11.77	0.90	58.43	63	4.38	57	75	312	313
8/22/2004	8:15-8:30	83	10.78	10.07	0.97	52.45	72	5.81	60	81	332	333
8/21/2004	7:15-7:30	86	10.72	10.56	0.92	56.15	67	3.68	56	73	344	345
8/22/2004	8:30-8:45	88	9.99	9.49	0.75	47.34	70	4.63	60	79	352	353
8/21/2004	8:00-8:15	104	8.72	6.95	0.69	31.51	64	3.63	57	72	416	417
8/22/2004	8:45-9:00	107	8.61	8.38	0.72	43.00	72	2.04	67	76	428	429
8/21/2004	7:45-8:00	111	7.99	5.73	0.74	32.08	68	3.86	58	75	444	445
8/20/2004	20:00-20:15	116	7.74	6.89	0.54	36.33	61	4.85	47	68	464	465
8/21/2004	8:15-8:30	116	7.78	6.65	0.58	36.77	66	4.30	56	76	464	465
8/21/2004	7:30-7:45	118	7.63	7.73	0.57	49.19	65	5.70	52	76	472	473
8/20/2004	19:45-20:00	120	7.47	6.25	0.95	28.71	61	4.30	52	70	480	481
8/20/2004	19:30-19:45	122	7.34	5.94	0.60	34.02	62	9.24	44	78	488	489
8/22/2004	9:00-9:15	123	7.37	7.20	0.48	42.23	69	3.92	61	77	492	493
8/21/2004	9:30-9:45	124	7.24	6.56	0.62	29.69	66	5.53	56	78	496	497

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/21/2004	19.15-19.30	127	7.12	5.76	0.58	40.51	70	4.30	61	79	508	509
8/21/2004	20.00-20:15	129	6.97	5.98	0.76	32.29	69	5.57	56	79	516	517
8/22/2004	9.30-9.45	129	6.93	5.24	0.63	21.10	73	3.54	63	79	516	517
8/20/2004	19.15-19.30	130	6.97	6.28	0.54	30.14	67	4.81	54	76	520	521
8/21/2004	19.45-20.00	130	6.87	5.11	0.72	27.18	68	4.12	59	77	520	521
8/21/2004	9.00-9.15	132	6.79	5.84	0.69	27.54	67	5.93	53	79	528	529
8/22/2004	9.15-9.30	132	6.82	6.83	0.71	52.00	70	3.78	61	77	528	529
8/21/2004	19.00-19.15	134	6.94	7.34	0.31	54.43	70	4.33	59	80	536	537
8/21/2004	18.15-18.30	136	6.41	5.63	0.34	33.61	68	4.57	60	78	544	545
8/22/2004	10.15-10.30	136	6.52	5.70	0.70	25.01	69	4.49	57	76	544	545
8/20/2004	19.00-19.15	139	6.43	5.83	0.69	36.44	63	6.66	50	78	556	557
8/21/2004	18.45-19.00	139	6.25	5.12	0.59	24.56	68	5.75	53	78	556	557
8/22/2004	10.00-10.15	139	6.67	5.35	0.57	30.14	69	4.18	56	76	556	557
8/20/2004	6:45-7:00	141	6.41	4.96	0.63	39.27	64	5.14	51	81	564	565
8/20/2004	18.15-18.30	141	6.39	5.57	0.38	29.10	62	4.09	53	73	564	565
8/20/2004	18.45-19.00	141	6.37	4.73	0.49	21.43	67	4.41	58	77	564	565
8/21/2004	17.15-17.30	141	6.40	4.05	0.65	20.48	71	3.90	63	79	564	565
8/21/2004	10.15-10.30	142	6.31	5.56	0.88	27.92	65	5.12	53	75	568	569
8/22/2004	10.30-10.45	142	6.44	5.81	0.49	28.58	68	6.13	53	81	568	569
8/21/2004	19.30-19.45	143	6.29	5.06	0.77	33.25	72	5.92	57	83	572	573
8/20/2004	18.30-18.45	144	6.28	5.69	0.55	45.55	61	5.84	46	73	576	577

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/21/2004	9.45-10:00	144	6.29	5.83	0.69	36.36	65	4.86	53	75	576	577
8/22/2004	10.45-11.00	145	6.19	5.22	0.67	25.90	69	6.13	57	81	580	581
8/20/2004	8.45-9.00	146	6.16	5.29	0.76	25.19	59	5.78	45	70	584	585
8/20/2004	9.15-9.30	146	6.17	5.69	0.49	44.22	62	6.94	47	77	584	585
8/20/2004	13.30-13.45	146	6.20	4.62	0.52	21.14	61	4.91	52	72	584	585
8/21/2004	13.30-13.45	146	6.15	5.17	0.76	28.37	66	5.30	53	74	584	585
8/20/2004	10.00-10.15	148	6.13	5.12	0.58	26.71	58	5.69	43	71	592	593
8/22/2004	19.15-19.30	148	6.02	5.77	0.33	44.18	68	5.73	53	78	592	593
8/20/2004	9.45-10:00	149	6.03	4.25	0.81	23.52	60	4.52	49	70	596	597
8/20/2004	11.15-11.30	149	6.05	5.08	0.60	28.57	58	5.08	45	70	596	597
8/21/2004	13.45-14.00	149	6.09	5.17	0.58	32.83	65	5.41	56	80	596	597
8/20/2004	16.00-16.15	150	5.98	4.59	0.76	22.70	63	4.19	53	72	600	601
8/21/2004	8.45-9.00	151	5.94	6.11	0.49	39.45	66	5.20	54	76	604	605
8/20/2004	9.00-9.15	152	5.94	4.68	0.67	28.90	60	4.14	52	70	608	609
8/20/2004	17.45-18.00	152	5.93	5.06	0.62	28.45	63	6.90	46	74	608	609
8/20/2004	7.45-8.00	153	5.88	4.80	0.53	37.13	61	5.17	52	74	612	613
8/20/2004	12.00-12.15	153	5.87	5.12	0.57	33.31	62	5.38	49	74	612	613
8/21/2004	17.45-18.00	153	5.84	4.37	0.81	23.15	68	3.67	60	80	612	613
8/20/2004	7:00-7:15	155	5.80	4.37	0.66	22.19	63	6.37	45	76	620	621
8/22/2004	14.00-14.15	155	5.82	5.11	0.51	33.72	66	6.37	50	79	620	621
8/22/2004	19.30-19.45	155	5.85	4.57	0.49	26.70	67	5.01	55	75	620	621

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/21/2004	10.45-11.00	156	5.74	5.09	0.89	25.78	65	3.82	57	75	624	625
8/21/2004	15.30-15.45	156	5.79	4.72	0.49	26.02	67	4.11	57	77	624	625
8/21/2004	17.30-17.45	156	5.78	5.15	0.48	34.46	69	3.82	61	76	624	625
8/21/2004	18.00-18.15	156	5.81	4.56	0.98	22.53	68	4.64	59	78	624	625
8/22/2004	11.00-11.15	156	5.74	4.89	0.63	23.79	68	3.76	62	77	624	625
8/22/2004	11.15-11.30	156	5.81	5.81	0.61	35.65	69	2.84	62	75	624	625
8/22/2004	13.45-14.00	156	5.74	4.56	0.68	22.19	71	4.61	61	81	624	625
8/22/2004	16.00-16.15	157	5.71	4.68	0.70	26.24	66	5.40	53	78	628	629
8/21/2004	9.15-9.30	158	5.72	4.46	0.83	23.53	67	4.54	56	75	632	633
8/21/2004	10.00-10.15	158	5.71	5.11	0.56	27.17	66	6.00	55	78	632	633
8/21/2004	15.15-15.30	158	5.68	5.43	0.73	38.41	67	4.57	58	77	632	633
8/20/2004	8.30-8.45	159	5.65	3.99	0.47	21.50	59	6.46	44	72	636	637
8/20/2004	18.00-18.15	159	5.63	4.00	0.47	23.18	67	4.35	56	76	636	637
8/21/2004	13.00-13.15	159	5.66	5.68	0.53	50.56	64	5.54	52	75	636	637
8/21/2004	18.30-18.45	159	5.82	4.72	0.44	32.14	69	4.71	58	80	636	637
8/21/2004	12.45-13.00	160	5.66	4.28	0.63	29.46	62	5.22	51	72	640	641
8/20/2004	13.00-13.15	161	5.55	4.12	0.67	22.39	59	4.84	46	70	644	645
8/21/2004	14.30-14.45	161	5.66	4.39	0.78	28.44	65	5.51	55	80	644	645
8/21/2004	16.45-17.00	161	5.55	4.76	0.67	24.98	67	4.58	55	77	644	645
8/21/2004	17.00-17.15	161	5.60	4.68	0.66	34.05	68	5.09	55	78	644	645
8/20/2004	9.30-9.45	163	5.50	4.40	0.49	21.88	58	7.27	46	78	652	653

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/22/2004	12.00-12.15	163	5.50	4.42	0.66	19.75	71	5.63	56	78	652	653
8/22/2004	12.15-12.30	163	5.55	4.29	0.60	27.17	68	4.35	59	77	652	653
8/20/2004	11.30-11.45	164	5.49	4.01	0.52	21.61	59	3.11	52	67	656	657
8/21/2004	14.15-14.30	164	5.40	4.49	0.51	26.66	66	4.45	53	73	656	657
8/20/2004	10.15-10.30	165	5.44	4.22	0.56	25.61	58	6.29	44	69	660	661
8/20/2004	17.00-17.15	165	5.53	4.60	0.59	21.57	65	4.14	54	77	660	661
8/21/2004	15.45-16.00	165	5.47	5.06	0.75	31.33	68	8.31	50	80	660	661
8/21/2004	16.30-16.45	165	5.45	4.18	0.58	23.41	67	5.55	57	80	660	661
8/21/2004	11.00-11.15	166	5.46	4.91	0.53	31.63	68	6.57	54	79	664	665
8/21/2004	10.30-10.45	167	5.39	4.25	0.46	20.48	64	6.76	52	79	668	669
8/21/2004	16.00-16.15	167	5.27	4.32	0.55	26.24	64	5.37	52	75	668	669
8/20/2004	8.00-8.15	169	5.36	4.67	0.60	28.95	63	6.46	44	79	676	677
8/20/2004	8.15-8.30	169	5.95	4.91	0.83	26.66	64	7.01	45	78	676	677
8/21/2004	12.15-12.30	169	5.39	4.46	0.59	21.36	65	3.56	55	73	676	677
8/21/2004	12.30-12.45	169	5.32	3.61	0.62	15.36	67	4.75	57	76	676	677
8/21/2004	15.00-15.15	169	5.38	4.45	0.60	23.42	66	4.20	58	76	676	677
8/22/2004	12.30-12.45	169	5.30	4.46	0.65	24.25	67	3.85	55	74	676	677
8/20/2004	10.45-11.00	170	5.32	4.40	0.38	26.60	57	4.39	49	67	680	681
8/20/2004	12.15-12.30	170	5.32	3.82	0.75	19.23	57	6.51	43	71	680	681
8/20/2004	12.45-13.00	170	5.29	4.50	0.50	20.52	58	4.49	48	68	680	681
8/20/2004	13.15-13.30	170	5.29	4.53	0.56	30.97	64	5.80	52	75	680	681

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/21/2004	14.00-14.15	170	5.29	4.06	0.63	26.13	68	5.90	53	77	680	681
8/22/2004	13.15-13.30	170	5.25	4.66	0.49	30.23	68	4.66	53	76	680	681
8/22/2004	14.45-15.00	170	5.29	4.22	0.30	24.48	69	3.69	59	77	680	681
8/22/2004	15.45-16.00	170	5.29	4.12	0.79	22.37	66	4.79	55	77	680	681
8/21/2004	11.15-11.30	171	5.25	3.74	0.63	23.57	65	5.12	52	77	684	686
8/21/2004	16.15-16.30	171	5.37	4.43	0.59	29.57	68	7.51	50	79	684	686
8/22/2004	14.30-14.45	171	5.36	4.46	0.50	22.60	69	5.51	54	79	684	686
8/22/2004	20.00-20:15	171	5.23	4.15	0.82	32.61	70	5.34	56	77	684	686
8/20/2004	14.00-14.15	172	5.23	4.44	0.44	30.11	60	5.75	49	74	688	690
8/20/2004	17.15-17.30	172	5.24	3.67	0.58	16.00	63	4.10	55	74	688	690
8/21/2004	12.00-12.15	172	5.19	4.10	0.31	21.19	63	6.45	45	77	688	690
8/22/2004	19.45-20.00	172	5.26	4.11	0.56	22.54	68	4.55	58	79	688	690
8/20/2004	17.30-17.45	173	5.18	3.78	0.65	20.82	64	5.63	48	73	692	694
8/22/2004	15.15-15.30	173	5.16	4.01	0.75	19.93	69	3.21	61	76	692	694
8/22/2004	18.30-18.45	173	5.29	4.21	0.59	27.82	68	6.15	56	78	692	694
8/20/2004	7:30-7:45	174	5.13	4.27	0.40	29.74	66	6.38	49	77	696	698
8/22/2004	16.15-16.30	174	5.19	4.17	0.61	23.64	67	3.65	56	75	696	698
8/22/2004	17.15-17.30	174	5.16	4.49	0.55	28.72	65	7.16	48	76	696	698
8/20/2004	14.30-14.45	175	5.12	4.35	0.62	30.14	62	4.76	53	73	700	702
8/22/2004	13.00-13.15	175	5.14	4.79	0.61	31.27	69	4.57	56	78	700	702
8/22/2004	15.30-15.45	175	5.17	4.17	0.52	22.04	69	3.13	63	78	700	702

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/20/2004	13.45-14.00	176	5.13	4.12	0.72	19.06	58	4.33	51	73	704	706
8/20/2004	14.45-15.00	176	5.11	3.98	0.73	24.23	62	6.47	47	78	704	706
8/20/2004	15.00-15.15	176	5.12	3.75	0.57	21.90	60	5.64	43	69	704	706
8/22/2004	11.45-12.00	176	5.11	4.68	0.58	30.88	70	5.86	52	81	704	706
8/22/2004	18.15-18.30	176	5.04	3.83	0.67	23.21	69	4.64	59	79	704	706
8/22/2004	18.45-19.00	177	5.07	4.09	0.64	24.52	68	4.18	59	77	708	710
8/20/2004	12.30-12.45	178	5.05	4.08	0.49	24.19	57	7.19	41	72	712	714
8/20/2004	15.45-16.00	178	5.06	3.83	0.37	23.62	61	5.00	48	71	712	714
8/21/2004	11.30-11.45	178	5.07	4.27	0.54	25.36	62	7.80	39	74	712	714
8/21/2004	13.15-13.30	178	5.01	3.52	0.53	17.56	66	4.72	55	76	712	714
8/20/2004	7:15-7:30	179	5.03	4.30	0.44	26.27	64	6.66	52	77	716	718
8/22/2004	11.30-11.45	180	4.99	4.01	0.66	27.08	71	3.92	61	81	720	722
8/22/2004	14.15-14.30	180	4.90	4.34	0.38	29.24	71	2.95	66	78	720	722
8/20/2004	11.00-11.15	181	4.98	3.51	0.80	17.02	58	6.45	42	74	724	726
8/20/2004	11.45-12.00	181	4.97	4.94	0.60	33.89	60	6.18	43	74	724	726
8/20/2004	14.15-14.30	181	4.97	3.54	0.39	17.29	60	4.82	46	69	724	726
8/20/2004	15.30-15.45	181	4.97	4.14	0.91	26.49	61	4.46	51	70	724	726
8/22/2004	18.00-18.15	181	4.95	4.81	0.82	44.09	67	6.38	53	79	724	726
8/22/2004	13.30-13.45	182	4.99	3.66	0.64	17.33	68	5.39	57	82	728	730
8/22/2004	17.30-17.45	182	4.95	3.77	0.62	22.89	66	4.37	56	75	728	730
8/21/2004	14.45-15.00	183	4.90	3.72	0.69	19.01	68	4.44	55	79	732	734

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/20/2004	10.30-10.45	184	4.86	3.90	0.73	22.56	56	5.09	45	66	736	738
8/20/2004	15.15-15.30	184	4.91	3.49	0.60	22.50	64	5.31	48	74	736	738
8/22/2004	16.30-16.45	184	4.93	3.66	0.43	20.24	68	6.31	55	82	736	738
8/20/2004	16.45-17.00	185	4.80	4.29	0.39	26.57	62	6.98	42	72	740	742
8/21/2004	11.45-12.00	185	4.82	3.75	0.81	19.80	65	4.05	57	76	740	742
8/20/2004	16.15-16.30	186	4.79	3.41	0.64	20.42	57	5.85	42	69	744	746
8/22/2004	16.45-17.00	186	4.83	3.46	0.59	20.90	66	4.06	56	76	744	746
8/22/2004	19.00-19.15	186	4.85	3.89	0.69	23.91	69	4.51	59	80	744	746
8/20/2004	16.30-16.45	189	4.82	4.16	0.76	25.79	63	4.66	52	72	756	758
8/22/2004	17.45-18.00	190	4.75	3.68	0.52	19.20	65	3.97	55	73	760	762
8/22/2004	17.00-17.15	191	4.71	3.49	0.26	21.94	67	4.52	55	75	764	766
8/22/2004	15.00-15.15	195	4.61	3.95	0.38	26.87	68	5.43	55	78	780	782
N=		159	159	159			159	159			159	159
Average=		153	6.28	5.26			65	5.04			614	615
Standard Deviation=		29.97	2.31	2.23			3.84	1.16			120	120
Minimum=		46	4.61	3.41			56	2.04			184	184
Maximum=		195	20.90	19.29			73	9.24			780	782
*3 additional time intervals (randomly selected from a total of 162 intervals) were selected for a later validation of the model.												
*8/22/2004	9.45-10:00	117	7.51	6.70	0.66	43.67	66	5.13	56	77	468	469
*8/21/2004	8.30-8.45	145	6.24	4.90	0.62	23.28	68	4.50	58	76	580	581
*8/22/2004	12.45-13.00	174	5.20	3.88	0.48	22.98	67	4.57	59	77	696	698

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/22/2004	7.15-7.30	16	70.29	86.33	2.73	240.03	72	1.54	69	75	64	66
8/22/2004	7.45-8.00	16	56.44	48.93	0.81	157.66	76	2.10	73	81	64	66
8/21/2004	6:45-7:00	20	36.67	41.87	0.81	176.74	73	2.56	68	77	80	82
8/22/2004	7.30-7.45	21	41.47	50.31	0.71	172.08	72	1.19	71	74	84	86
8/22/2004	8.15-8.30	23	39.12	41.36	0.69	131.26	76	2.75	70	79	92	95
8/22/2004	8.45-9.00	26	37.01	34.79	0.63	121.23	70	2.33	65	73	104	107
8/21/2004	7:00-7.15	30	33.55	38.95	2.22	155.96	73	1.50	70	76	120	123
8/21/2004	7.15-7.30	30	30.55	37.71	0.48	149.27	72	1.66	69	76	120	123
8/21/2004	7.45-8.00	31	29.17	35.79	0.57	137.21	70	3.98	64	78	124	127
8/22/2004	9.15-9.30	31	29.45	33.96	0.65	149.79	73	2.82	63	77	124	127
8/22/2004	8.30-8.45	32	25.64	37.34	0.88	165.68	71	2.41	67	76	128	132
8/22/2004	9.30-9.45	40	22.44	28.74	0.64	148.22	71	2.76	67	76	160	164
8/21/2004	20.00-20.15	46	20.59	27.56	0.85	145.36	71	2.50	66	78	184	189
8/21/2004	7.30-7.45	48	18.66	19.53	0.57	69.03	72	2.24	68	76	192	197
8/21/2004	8.00-8.15	49	17.29	20.73	0.69	81.57	71	3.44	64	77	196	201
8/21/2004	19.30-19.45	49	15.44	13.34	0.49	47.75	76	2.41	71	80	196	201
8/22/2004	10.30-10.45	49	18.29	30.01	0.58	181.63	72	3.05	68	77	196	201
8/22/2004	9.45-10.00	52	17.41	20.98	0.58	80.30	74	2.97	69	79	208	214
8/21/2004	17.15-17.30	54	16.90	19.29	0.28	91.31	72	1.77	68	76	216	222
8/22/2004	11.00-11.15	55	15.27	16.91	1.05	78.39	72	1.82	68	75	220	226
8/22/2004	10.00-10.15	56	16.02	21.47	0.69	114.03	73	3.23	67	78	224	230

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/21/2004	19.15-19.30	58	15.58	19.79	0.66	86.13	71	3.92	64	79	232	238
8/21/2004	17.45-18.00	59	15.14	17.42	0.78	99.36	75	3.17	68	79	236	243
8/21/2004	19.45-20.00	63	15.66	22.58	0.68	151.05	71	5.31	63	80	252	259
8/22/2004	10.15-10.30	63	14.33	21.54	0.53	121.28	71	1.63	67	74	252	259
8/21/2004	8.15-8.30	64	14.41	17.36	0.49	80.40	69	3.62	63	76	256	263
8/21/2004	18.45-19.00	64	13.78	17.89	0.65	74.43	75	4.09	68	82	256	263
8/20/2004	9.45-10.00	67	13.02	14.55	0.61	72.88	69	2.22	64	74	268	276
8/20/2004	19.30-19.45	68	13.19	20.22	0.72	87.23	67	4.97	58	75	272	280
8/20/2004	20.00-20.15	68	13.11	15.86	0.58	67.84	68	2.58	64	75	272	280
8/21/2004	9.15-9.30	68	14.20	18.25	0.61	84.55	69	3.42	62	76	272	280
8/21/2004	19.00-19.15	68	13.46	21.37	0.70	141.96	70	4.27	63	80	272	280
8/21/2004	9.00-9.15	69	12.12	14.59	0.49	56.42	69	2.69	64	74	276	284
8/22/2004	10.45-11.00	69	12.88	16.33	0.55	73.03	71	2.97	67	79	276	284
8/21/2004	15.45-16.00	70	12.84	14.59	0.53	68.45	72	2.22	68	77	280	288
8/21/2004	18.15-18.30	71	12.67	20.18	0.56	92.68	72	2.67	67	77	284	292
8/20/2004	19.00-19.15	74	12.24	14.53	0.81	63.66	73	2.60	67	78	296	304
8/21/2004	8.45-9.00	74	12.08	15.70	0.48	103.44	72	1.80	68	75	296	304
8/22/2004	11.15-11.30	74	13.14	21.34	0.53	121.05	69	4.59	58	75	296	304
8/21/2004	8.30-8.45	75	12.51	17.75	0.48	68.04	71	2.06	67	75	300	308
8/20/2004	9.00-9.15	76	11.71	15.35	0.40	78.09	67	3.30	60	74	304	313
8/20/2004	19.15-19.30	77	11.14	14.19	0.62	82.89	72	3.73	65	78	308	317

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/20/2004	6:45-7:00	78	11.31	11.72	0.60	65.84	71	4.12	64	79	312	321
8/20/2004	18.45-19.00	79	11.32	15.46	0.54	74.36	72	3.77	63	79	316	325
8/22/2004	13.00-13.15	79	10.75	15.07	0.51	81.51	68	3.25	61	75	316	325
8/21/2004	10.15-10.30	80	11.15	16.67	0.50	84.57	69	2.41	65	74	320	329
8/22/2004	11.45-12.00	80	10.94	13.86	0.69	67.44	72	3.87	65	79	320	329
8/20/2004	8.45-9.00	81	11.09	12.73	0.74	50.93	67	2.67	59	72	324	333
8/20/2004	19.45-20.00	81	11.57	14.56	0.41	65.69	71	3.62	63	77	324	333
8/21/2004	10.00-10.15	81	11.12	13.81	0.62	66.63	71	3.36	65	78	324	333
8/21/2004	18.30-18.45	81	11.43	12.44	0.53	53.31	74	1.75	70	77	324	333
8/21/2004	9.30-9.45	82	10.47	15.67	0.33	78.55	71	4.00	62	80	328	337
8/21/2004	16.45-17.00	82	10.94	11.30	0.50	49.31	67	3.33	61	73	328	337
8/21/2004	17.00-17.15	82	10.75	13.31	0.35	61.42	72	4.06	63	80	328	337
8/21/2004	13.30-13.45	83	10.83	15.89	0.60	76.68	70	2.27	64	74	332	341
8/21/2004	17.30-17.45	83	11.03	14.50	0.44	90.81	72	2.06	68	76	332	341
8/22/2004	12.00-12.15	83	11.06	10.53	0.49	45.79	72	4.64	64	79	332	341
8/22/2004	13.45-14.00	83	10.74	12.30	0.51	50.54	71	2.02	67	75	332	341
8/22/2004	14.45-15.00	83	10.84	10.94	0.46	51.48	73	3.01	68	80	332	341
8/21/2004	11.00-11.15	84	10.85	12.40	0.60	66.21	71	2.27	67	76	336	345
8/21/2004	13.00-13.15	86	10.88	13.56	0.59	64.65	67	3.04	62	74	344	354
8/21/2004	16.30-16.45	86	10.36	11.56	0.47	63.05	71	4.04	62	78	344	354
8/21/2004	18.00-18.15	86	10.27	11.84	0.54	63.26	73	2.53	67	80	344	354

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/21/2004	16.00-16.15	88	10.04	14.51	0.53	68.43	73	2.44	67	77	352	362
8/21/2004	16.15-16.30	88	10.43	12.03	0.51	57.15	70	3.91	63	78	352	362
8/22/2004	11.30-11.45	88	10.17	10.99	0.57	43.39	73	3.22	66	79	352	362
8/20/2004	12.00-12.15	89	10.15	12.71	0.55	49.34	67	3.20	59	74	356	366
8/21/2004	12.30-12.45	89	10.17	12.08	0.56	62.65	71	2.75	64	76	356	366
8/22/2004	13.15-13.30	89	10.59	13.75	0.56	68.60	72	2.18	66	76	356	366
8/20/2004	8.15-8.30	92	9.81	12.78	0.52	78.58	68	4.85	57	75	368	378
8/20/2004	11.15-11.30	92	9.72	13.22	0.67	65.29	66	3.29	58	75	368	378
8/21/2004	11.15-11.30	92	9.84	12.21	0.58	60.89	70	2.97	64	76	368	378
8/21/2004	15.30-15.45	92	9.86	11.43	0.68	70.78	71	3.60	64	79	368	378
8/22/2004	19.15-19.30	92	9.61	11.44	0.52	57.25	69	3.82	61	76	368	378
8/20/2004	7:00-7.15	93	9.63	13.01	0.59	70.70	70	4.63	59	78	372	382
8/20/2004	9.30-9.45	94	9.73	14.17	0.40	72.17	67	3.69	59	73	376	387
8/20/2004	10.15-10.30	94	9.68	11.40	0.69	50.06	67	3.14	60	72	376	387
8/21/2004	12.00-12.15	94	9.34	11.27	0.49	59.64	70	3.64	62	78	376	387
8/22/2004	12.45-13.00	94	9.71	11.43	0.32	59.22	70	2.44	65	75	376	387
8/20/2004	8.30-8.45	95	9.49	8.46	0.51	40.78	70	2.60	65	76	380	391
8/22/2004	14.30-14.45	95	9.60	12.39	0.36	65.23	71	4.45	62	79	380	391
8/21/2004	11.45-12.00	96	9.30	12.73	0.60	77.35	68	3.86	60	75	384	395
8/22/2004	19.30-19.45	96	9.30	13.28	0.36	75.20	72	3.40	65	78	384	395
8/20/2004	11.30-11.45	97	9.32	10.32	0.34	54.16	66	4.85	58	77	388	399

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/20/2004	18.15-18.30	107	8.29	12.39	0.47	78.41	71	3.46	59	77	428	440
8/20/2004	17.30-17.45	108	8.54	10.30	0.51	54.99	72	3.05	66	77	432	444
8/20/2004	10.45-11.00	109	8.44	12.41	0.50	92.56	65	4.11	57	73	436	448
8/21/2004	13.15-13.30	109	8.20	8.91	0.41	35.90	72	3.69	65	79	436	448
8/22/2004	18.45-19.00	109	8.45	11.68	0.42	55.51	71	3.00	63	76	436	448
8/20/2004	14.00-14.15	110	8.07	11.09	0.58	73.68	69	3.32	62	76	440	452
8/20/2004	16.00-16.15	110	8.19	9.19	0.60	40.65	71	3.60	64	79	440	452
8/21/2004	12.45-13.00	110	7.85	9.39	0.69	44.12	69	3.36	62	77	440	452
8/22/2004	12.30-12.45	110	8.06	8.22	0.50	39.97	73	2.83	64	78	440	452
8/20/2004	11.45-12.00	111	8.03	11.36	0.49	64.75	66	2.97	59	72	444	456
8/21/2004	14.00-14.15	112	8.22	11.44	0.40	63.89	70	2.77	64	75	448	461
8/22/2004	15.15-15.30	112	7.94	10.49	0.38	52.48	70	2.17	64	75	448	461
8/20/2004	17.45-18.00	113	7.95	9.80	0.49	46.51	69	3.99	59	76	452	465
8/21/2004	11.30-11.45	115	7.82	10.45	0.43	55.96	69	3.53	62	77	460	473
8/20/2004	7.15-7.30	116	7.83	10.51	0.28	62.54	67	4.45	59	76	464	477
8/21/2004	14.15-14.30	116	7.56	9.67	0.37	45.91	69	3.67	64	78	464	477
8/20/2004	16.45-17.00	117	7.90	11.50	0.42	88.21	69	4.37	61	76	468	481
8/20/2004	17.15-17.30	117	7.49	8.89	0.43	47.02	71	2.06	67	77	468	481
8/20/2004	12.15-12.30	118	7.54	11.45	0.48	89.97	67	3.89	59	74	472	485
8/22/2004	18.15-18.30	118	7.59	8.75	0.55	35.47	70	2.32	64	74	472	485
8/20/2004	10.30-10.45	119	7.40	9.36	0.48	44.01	68	3.36	61	74	476	489

Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/20/2004	18.30-18.45	119	7.69	10.14	0.42	55.10	72	3.06	65	78	476	489
8/21/2004	14.45-15.00	119	7.47	10.40	0.47	73.44	70	3.06	62	75	476	489
8/22/2004	13.30-13.45	119	7.62	10.12	0.51	48.00	71	2.69	64	76	476	489
8/22/2004	16.30-16.45	119	7.61	8.86	0.38	45.15	73	2.74	66	83	476	489
8/22/2004	15.30-15.45	120	7.63	9.85	0.51	59.38	72	2.92	66	77	480	493
8/22/2004	18.30-18.45	121	7.29	8.77	0.44	40.40	69	4.30	61	77	484	498
8/20/2004	15.30-15.45	125	7.24	10.08	0.38	65.50	68	4.85	56	79	500	514
8/22/2004	19.45-20.00	125	7.39	8.95	0.40	57.82	71	3.02	66	78	500	514
8/20/2004	7.45-8.00	126	7.18	9.52	0.54	50.87	70	2.97	61	77	504	518
8/20/2004	13.45-14.00	127	7.08	9.24	0.29	63.52	71	3.52	62	78	508	522
8/22/2004	16.15-16.30	127	7.13	9.23	0.56	44.40	71	3.43	64	78	508	522
8/20/2004	15.15-15.30	128	7.01	9.24	0.50	60.20	70	4.50	63	80	512	526
8/20/2004	11.00-11.15	129	7.01	9.79	0.42	59.77	67	5.08	54	74	516	530
8/20/2004	16.15-16.30	129	6.82	9.05	0.42	49.23	63	5.91	50	75	516	530
8/20/2004	15.00-15.15	130	6.95	8.30	0.36	45.13	70	2.98	63	78	520	535
8/22/2004	17.15-17.30	130	7.01	8.73	0.33	51.96	72	2.67	68	81	520	535
8/20/2004	15.45-16.00	131	6.86	8.94	0.63	56.10	70	3.92	63	80	524	539
8/20/2004	7.30-7.45	132	6.80	9.25	0.53	68.40	70	4.94	57	79	528	543
8/20/2004	14.30-14.45	132	6.81	8.74	0.51	46.47	69	3.80	61	78	528	543
8/22/2004	16.00-16.15	132	6.88	8.98	0.29	50.66	71	2.56	65	76	528	543
8/20/2004	12.30-12.45	134	6.79	8.98	0.44	49.13	67	4.80	54	77	536	551

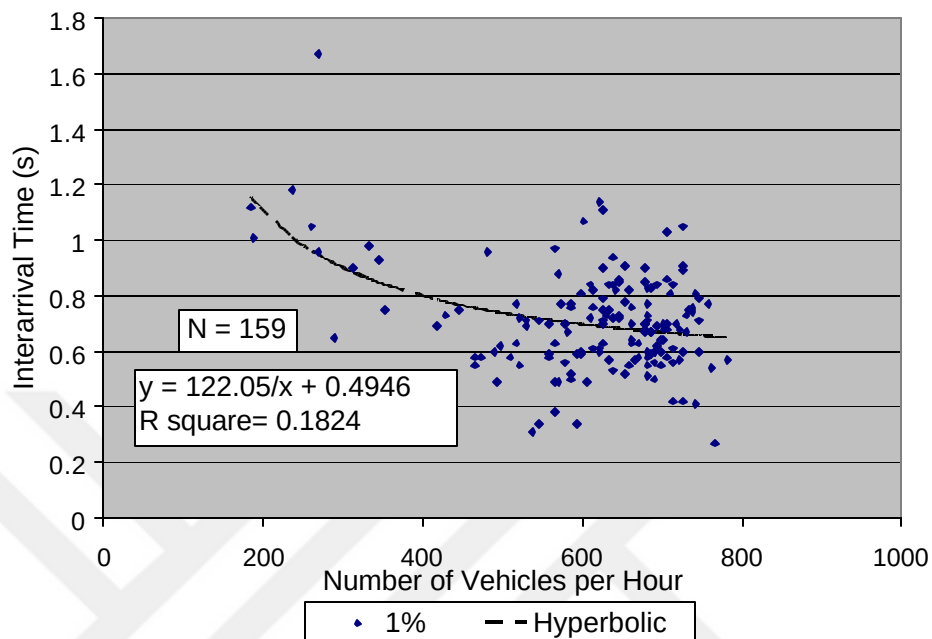
Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

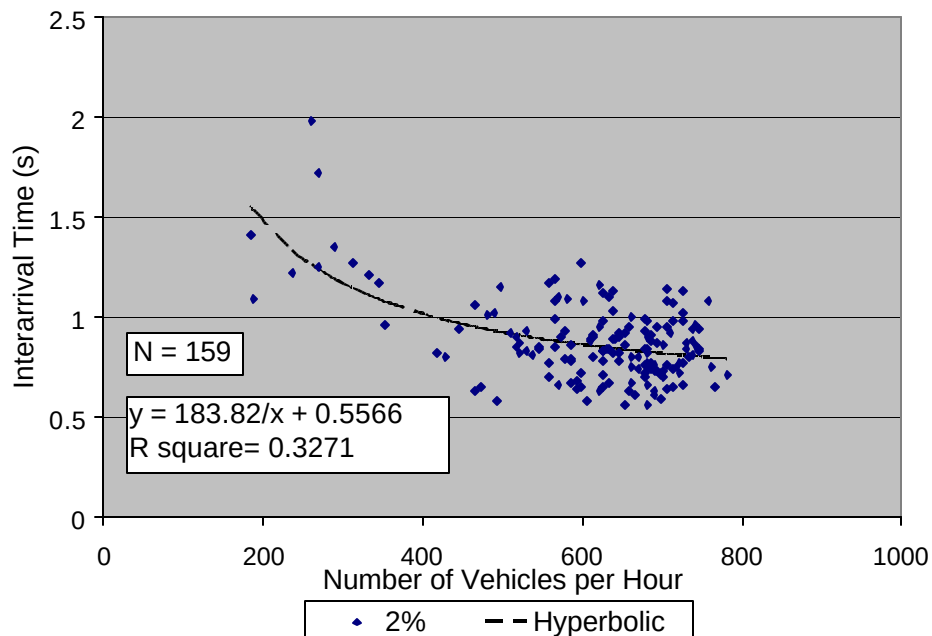
Date	Time	Number of Vehicles for 15 Minutes Intervals	Inter-arrival Time (second)				Speed (mph)				Number of Vehicles per Hour	Adjusted Number of Vehicles per Hour
			Average	Standard Deviation	Min	Max	Average	Standard Deviation	Min	Max		
8/22/2004	19.00-19.15	134	6.72	10.16	0.51	58.59	69	3.55	61	75	536	551
8/20/2004	14.45-15.00	137	6.56	7.97	0.28	43.76	70	4.31	58	79	548	563
8/22/2004	17.30-17.45	144	6.19	7.89	0.30	48.79	70	2.20	66	76	576	592
8/22/2004	17.00-17.15	146	6.18	8.21	0.36	42.74	73	3.43	64	79	584	600
8/22/2004	17.45-18.00	150	5.91	8.17	0.44	45.37	71	3.71	63	78	600	617
8/20/2004	16.30-16.45	151	5.91	7.47	0.54	51.02	69	3.56	59	76	604	621
8/22/2004	18.00-18.15	151	6.12	9.07	0.42	53.72	72	2.36	65	76	604	621
8/22/2004	16.45-17.00	157	5.64	7.53	0.54	53.81	72	3.74	66	80	628	646
8/20/2004	14.15-14.30	162	5.62	7.66	0.50	45.05	68	5.16	53	77	648	666
N=		156	156	156			156	156			156	156
Average=		92	12.07	15.02			70	3.30			368	378
Standard Deviation=		30.83	8.70	9.80			2.14	0.91			123	127
Minimum=		16	5.62	7.47			63	1.19			64	66
Maximum=		162	70.29	86.33			76	5.91			648	666
*3 additional time intervals (randomly selected from a total of 162 intervals) were selected for a later validation of the model.												
**3 time interval was not used to determine the model parameters since they produced outlier values												
*8/22/2004	9.00-9.15	37	24.46	27.66	0.62	91.45	66	3.01	62	72	148	152
*8/20/2004	13.00-13.15	102	8.76	12.58	0.33	59.92	65	5.13	54	76	408	419
*8/20/2004	12.45-13.00	126	7.20	9.40	0.45	46.04	68	3.62	57	75	504	518
**8/22/2004	6:45-7:00	5	152.50	111.55	56.79	324.31	71	1.34	69	72	20	21
**8/22/2004	7:00-7.15	6	129.10	142.13	0.90	382.97	73	1.47	71	75	24	25
**8/22/2004	8.00-8.15	15	61.82	41.23	12.81	139.87	68	2.32	63	71	60	62

Appendix B. IATs as a Function of Volume (vphpl) on Driving Lane and Passing Lane during Daytime for Cumulative Percentage Values 1%, 2%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 95%, 98%, 99%, and 100% (Maximum)

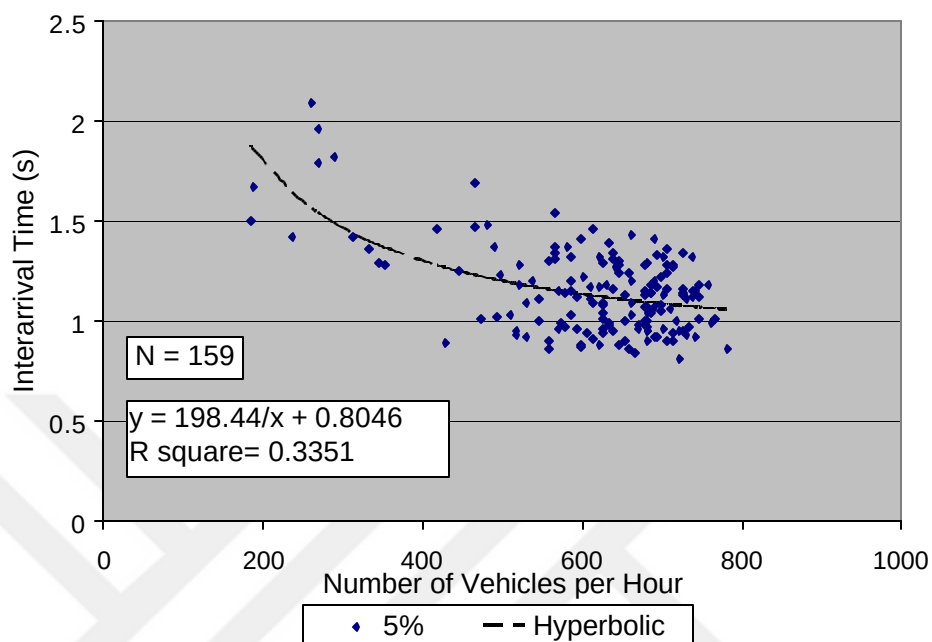




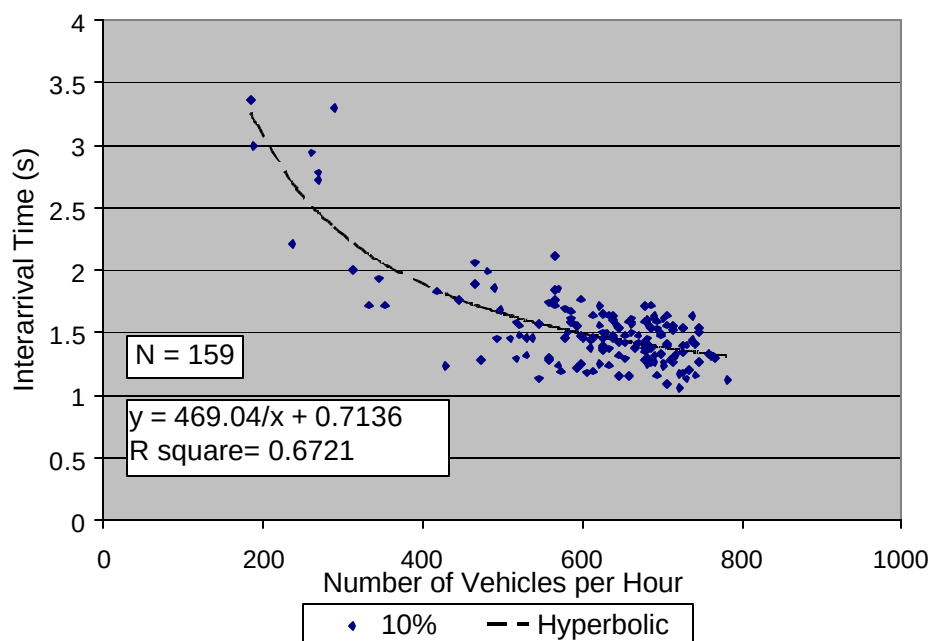
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 1%



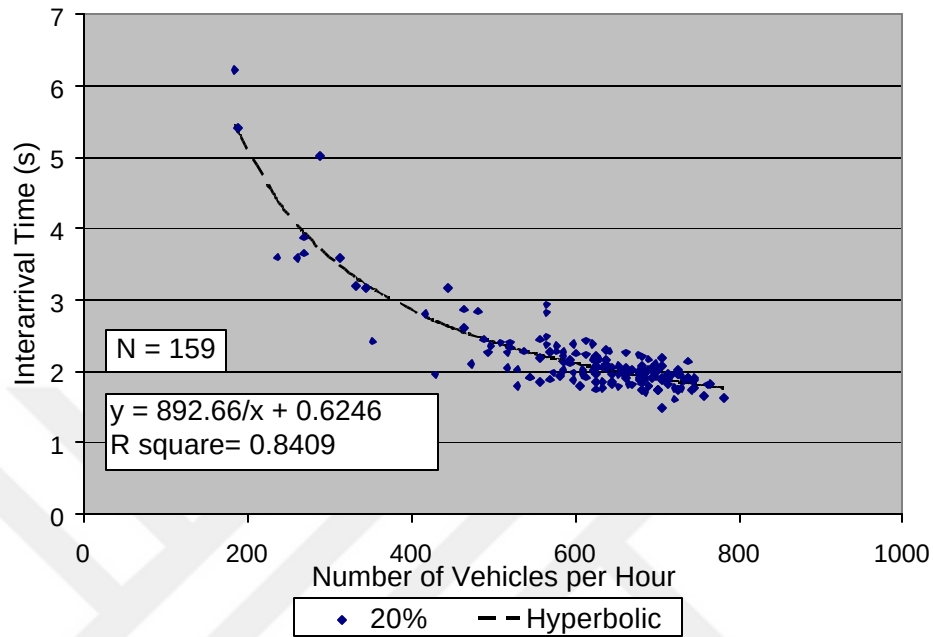
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 2%



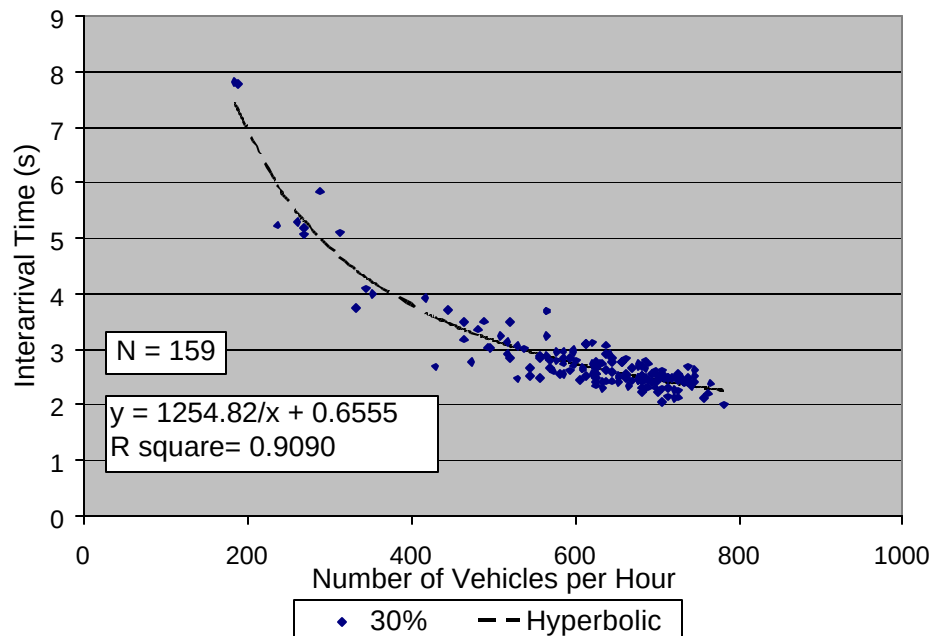
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 5%



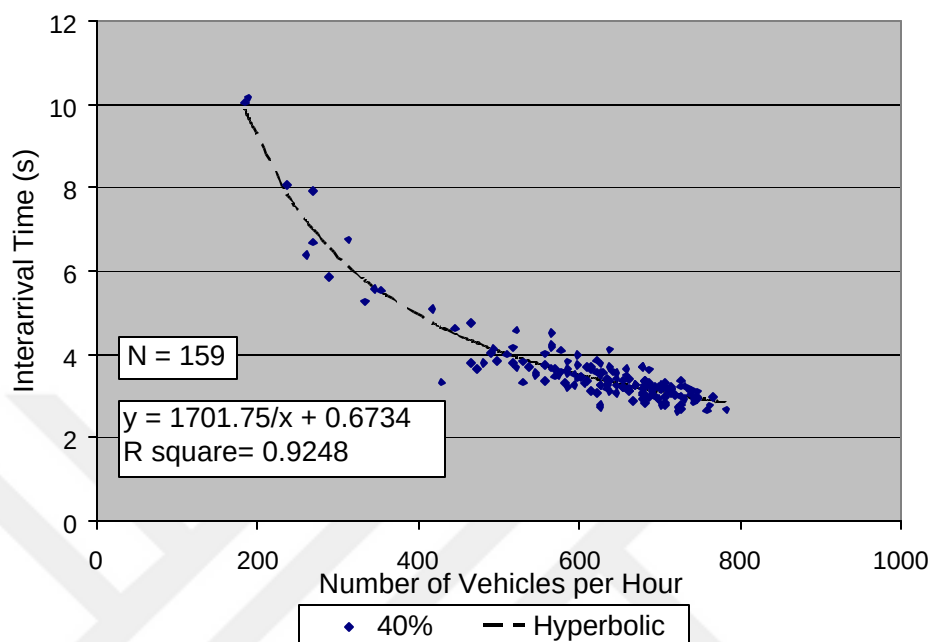
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 10%



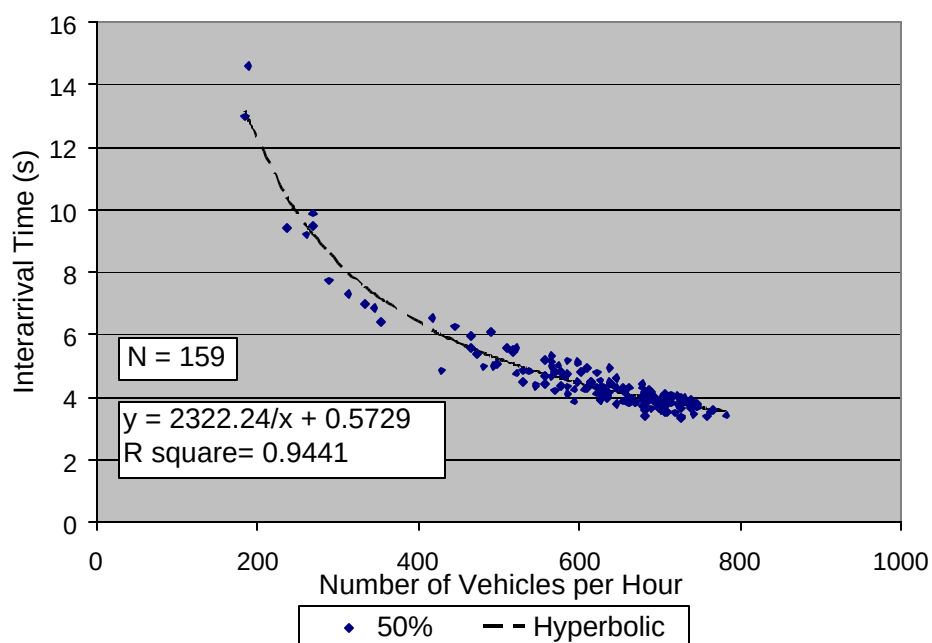
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 20%



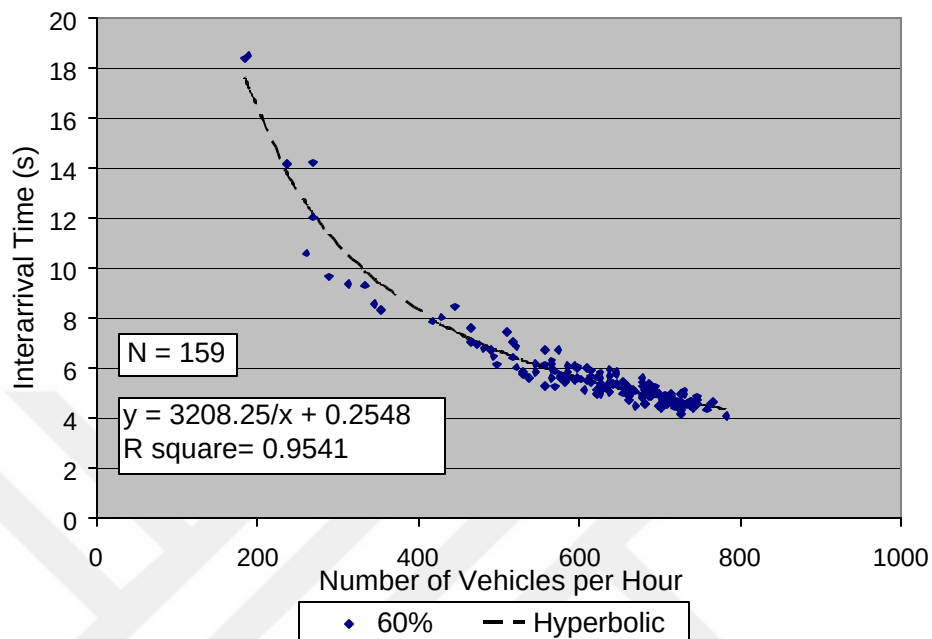
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 30%



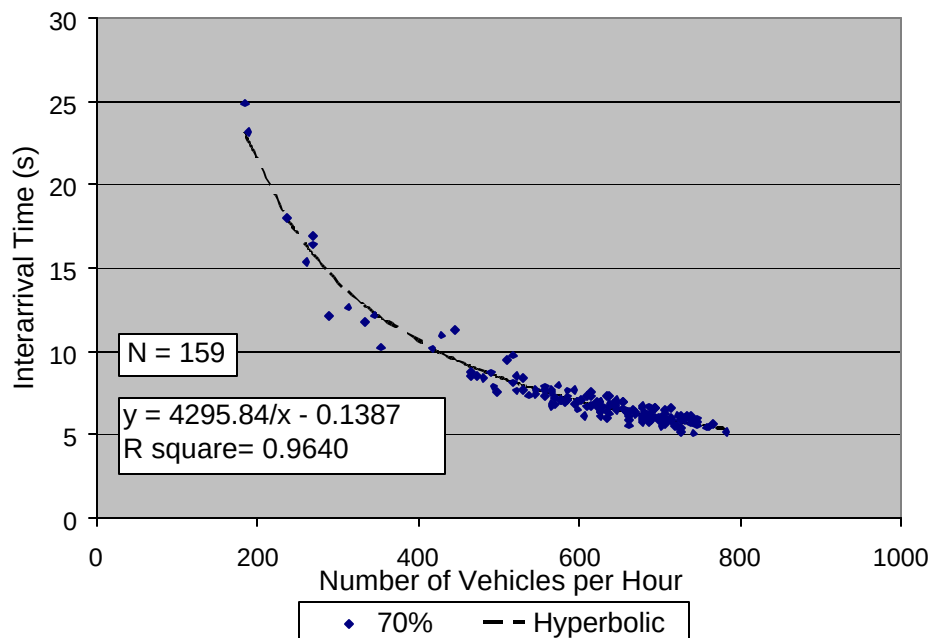
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 40%



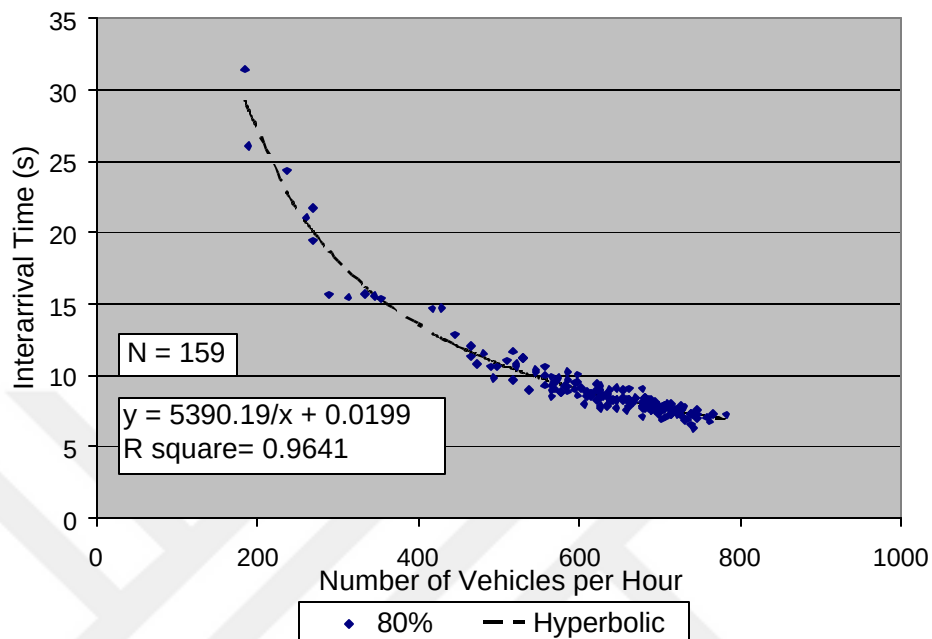
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 50%



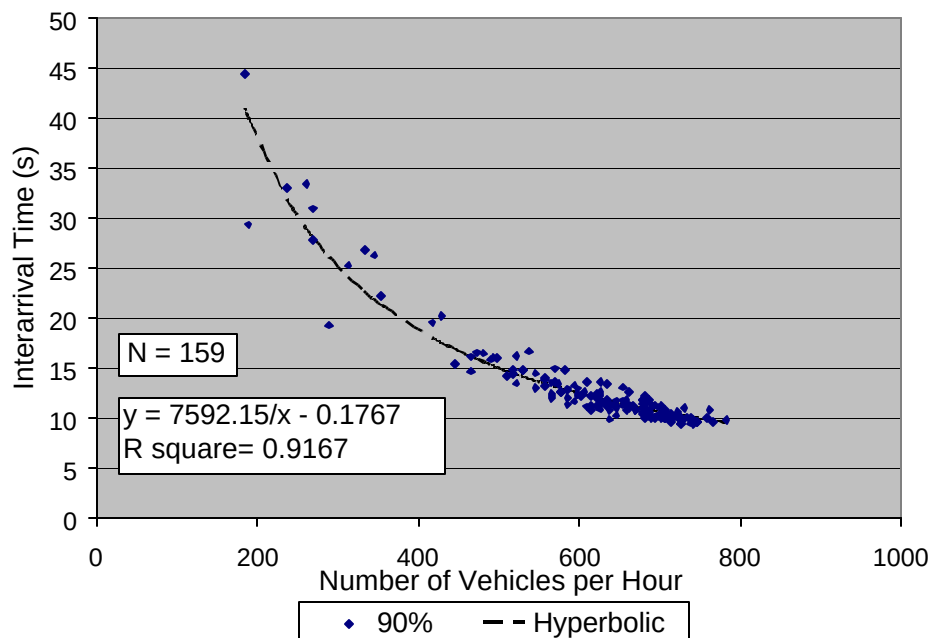
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 60%



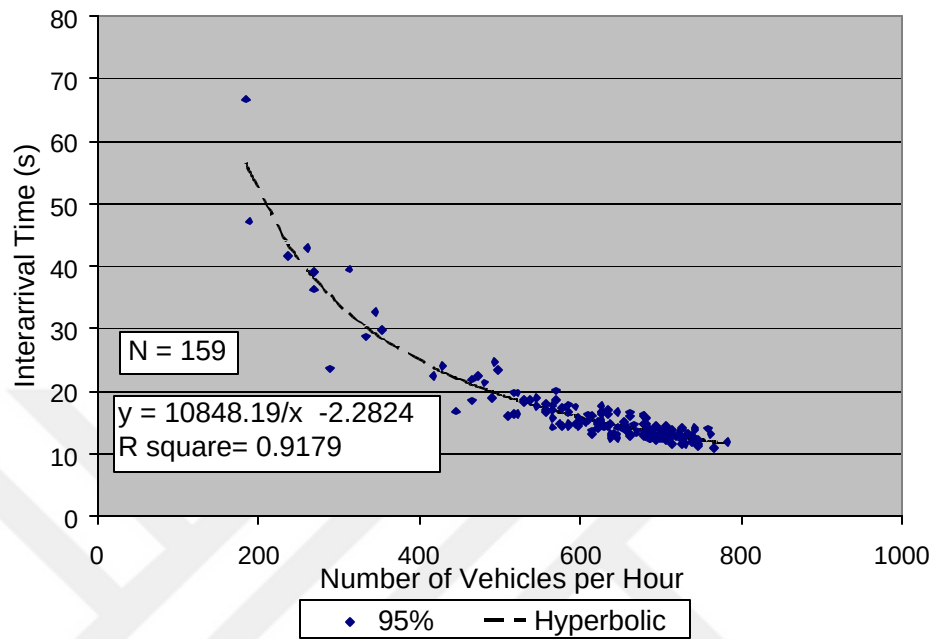
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 70%



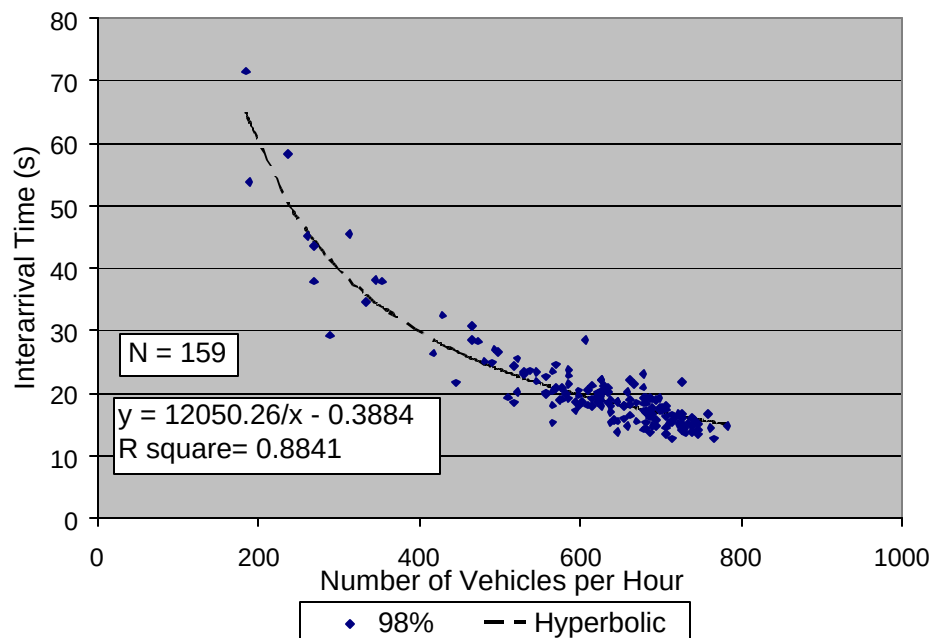
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 80%



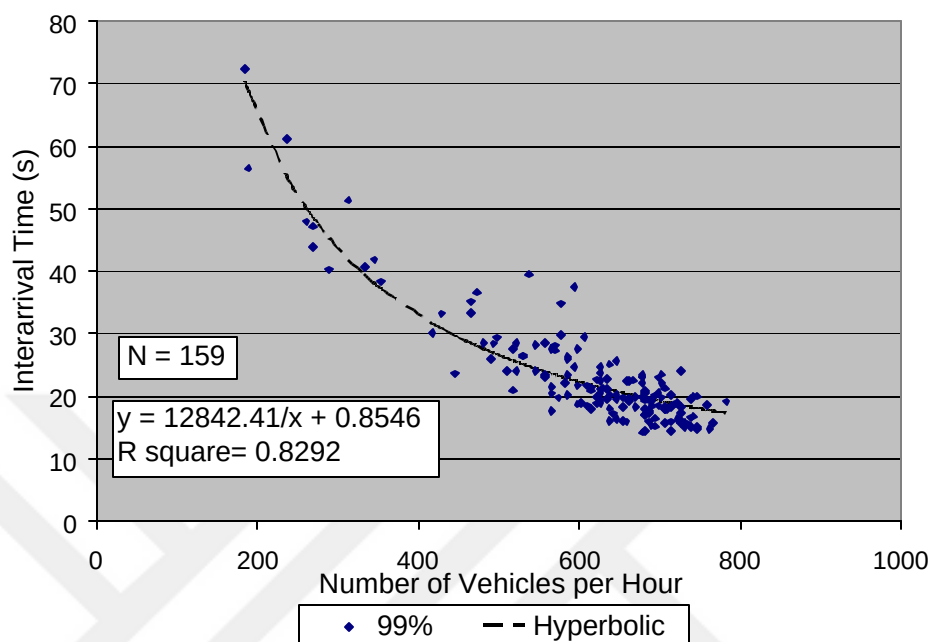
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 90%



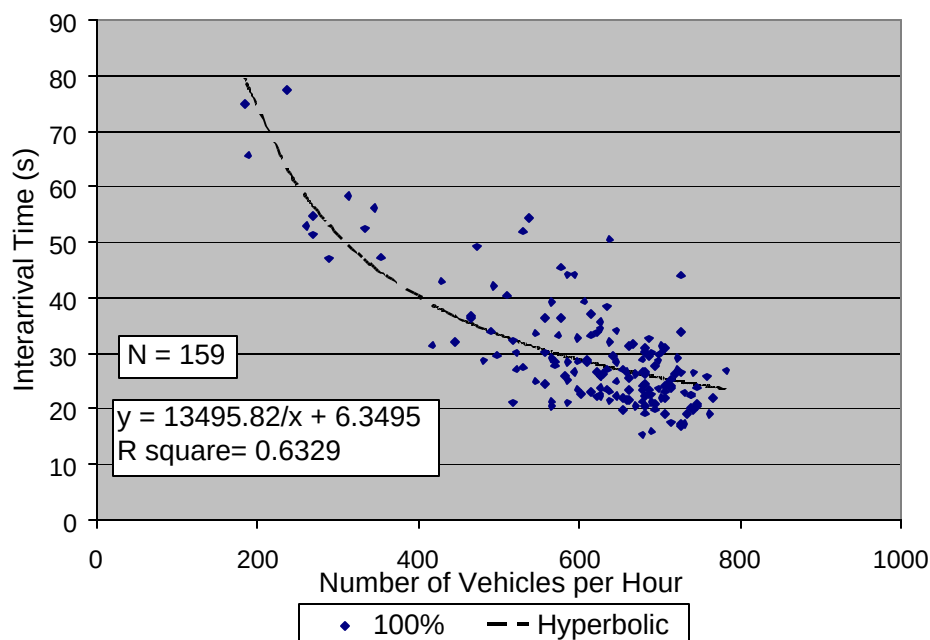
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 95%



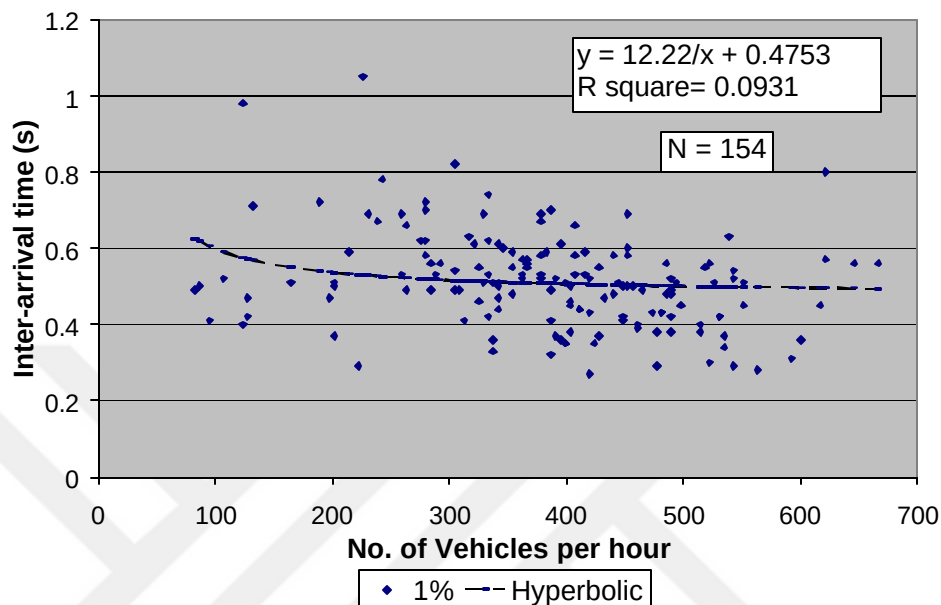
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 98%



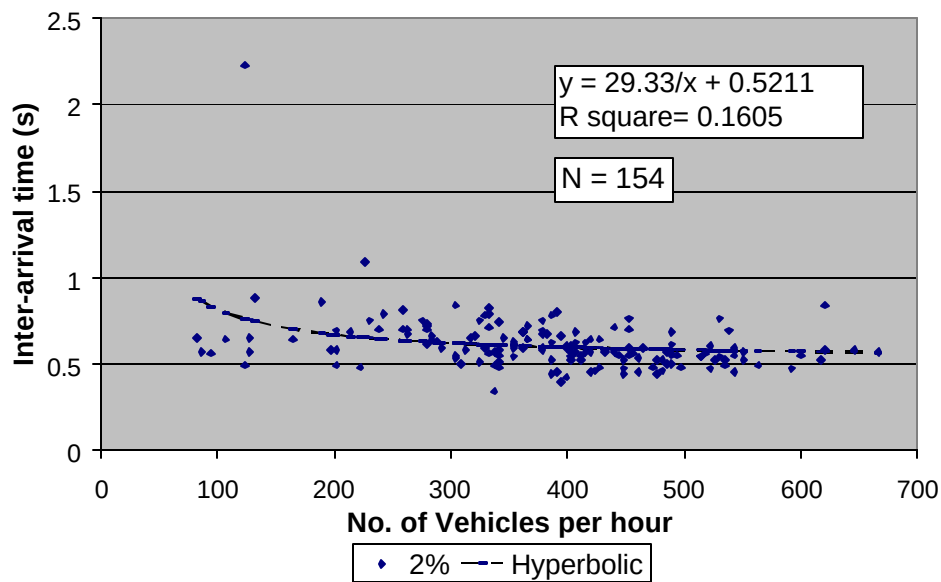
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 99%



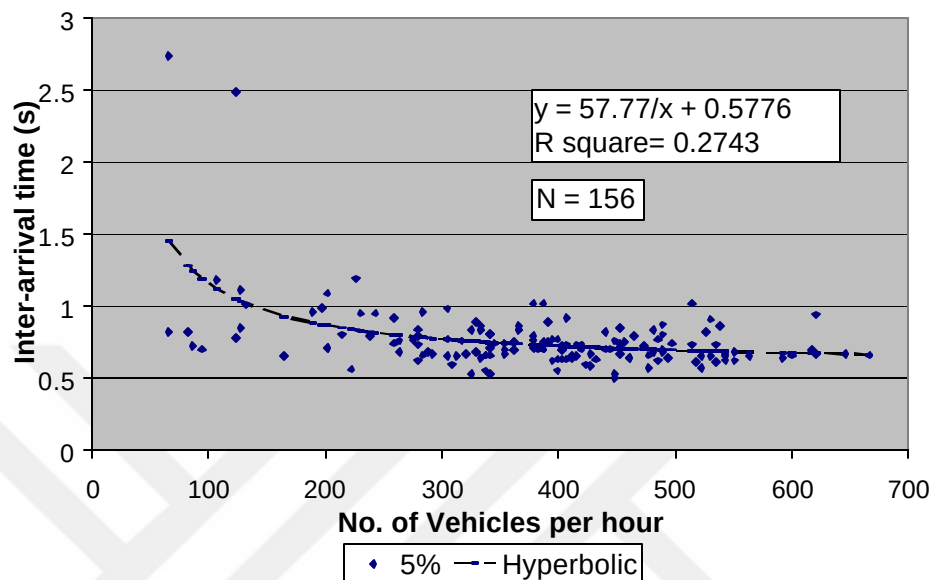
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 100% (Maximum)



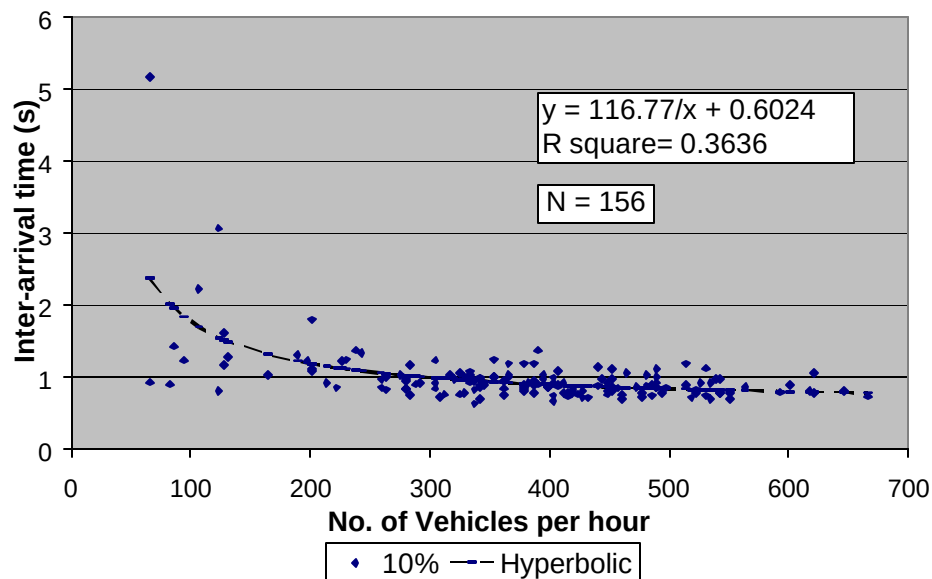
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 1%



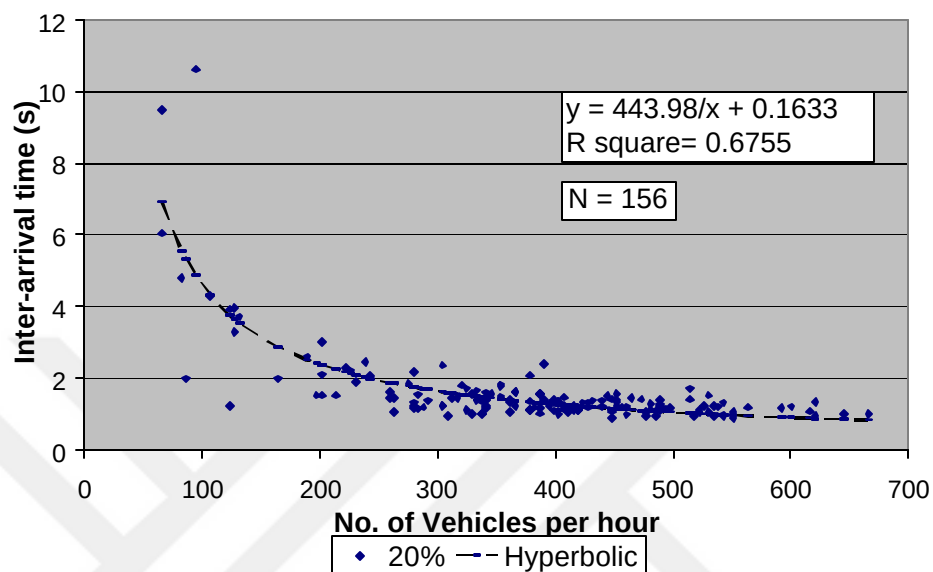
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 2%



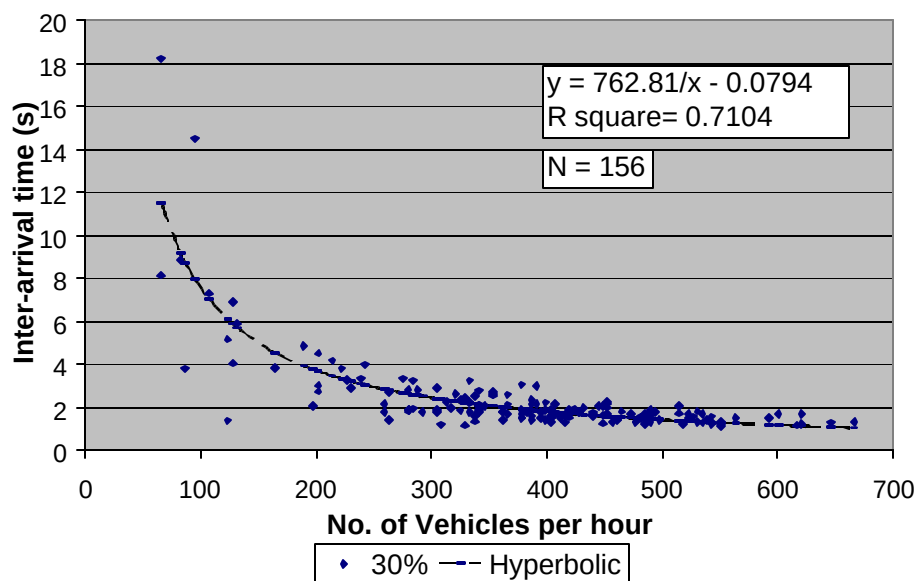
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 5%



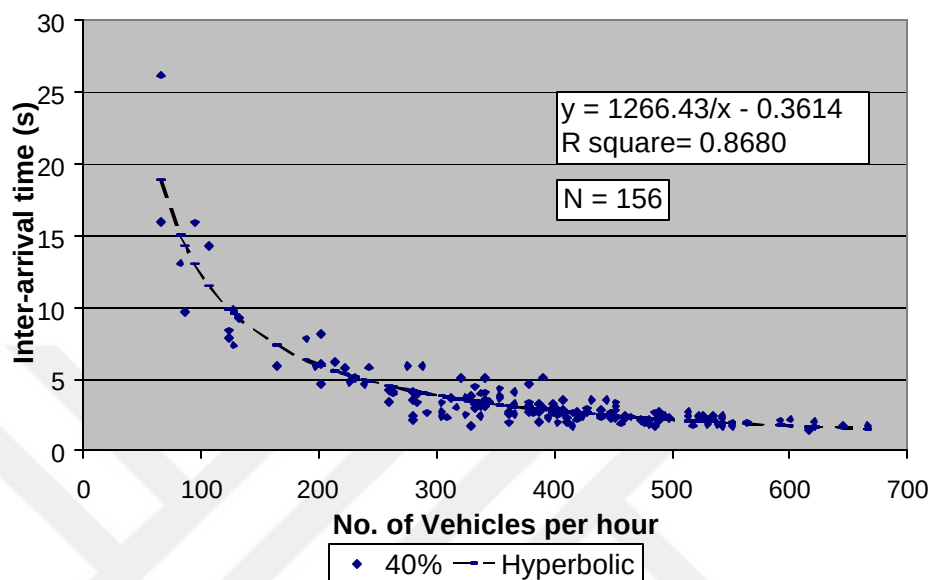
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 10%



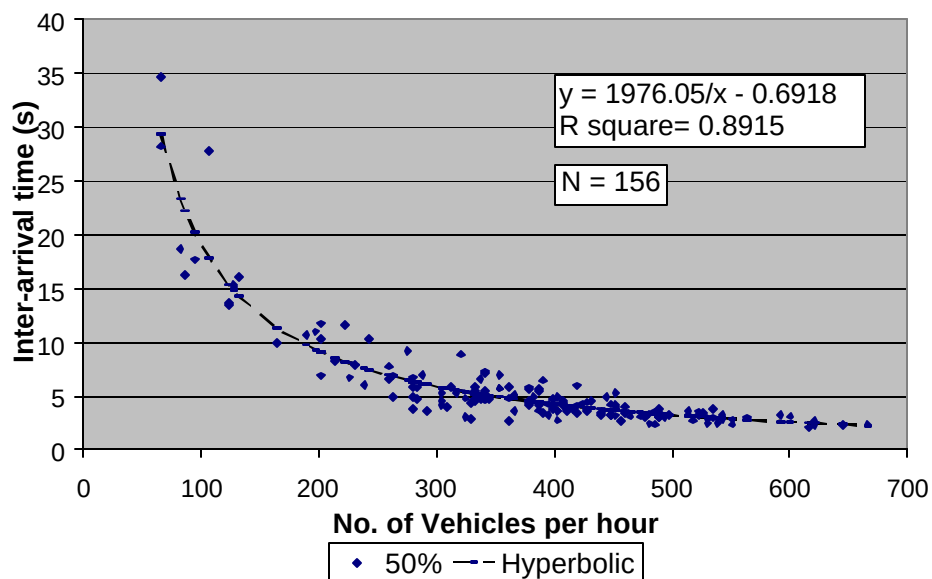
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 20%



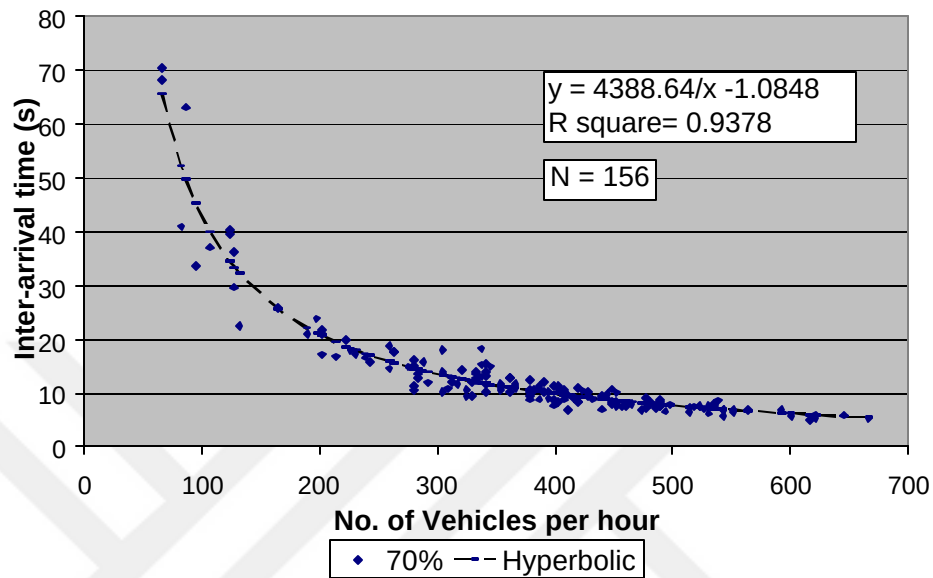
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 30%



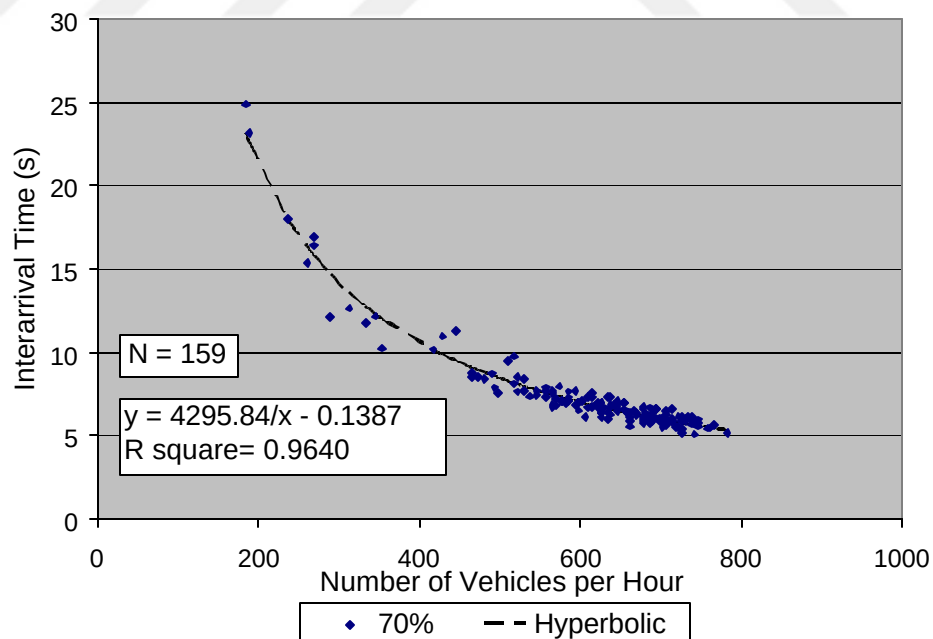
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 40%



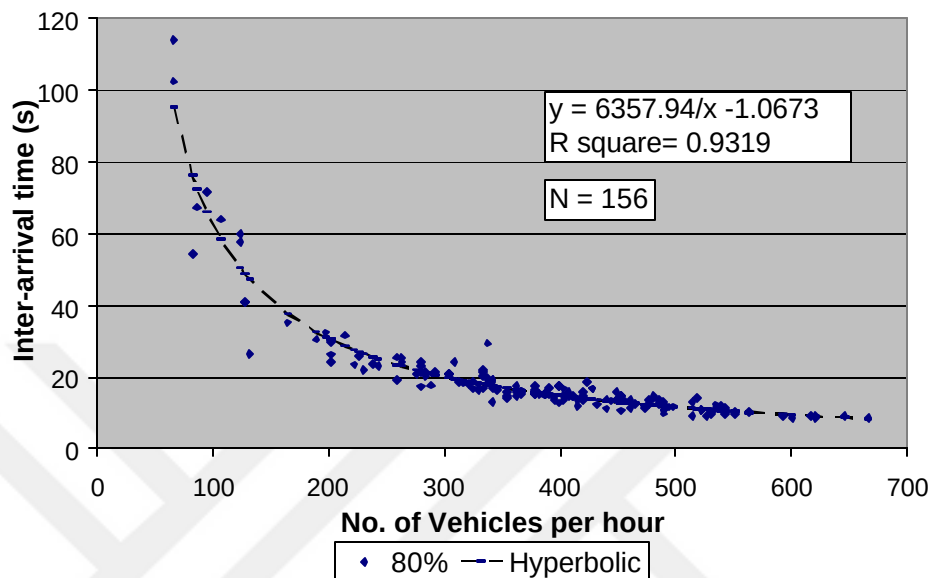
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 50%



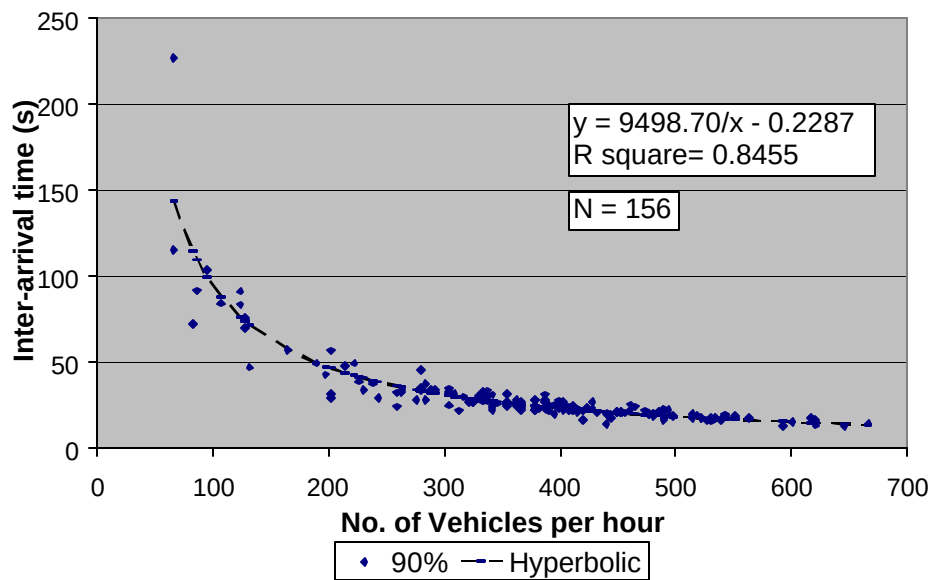
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 60%



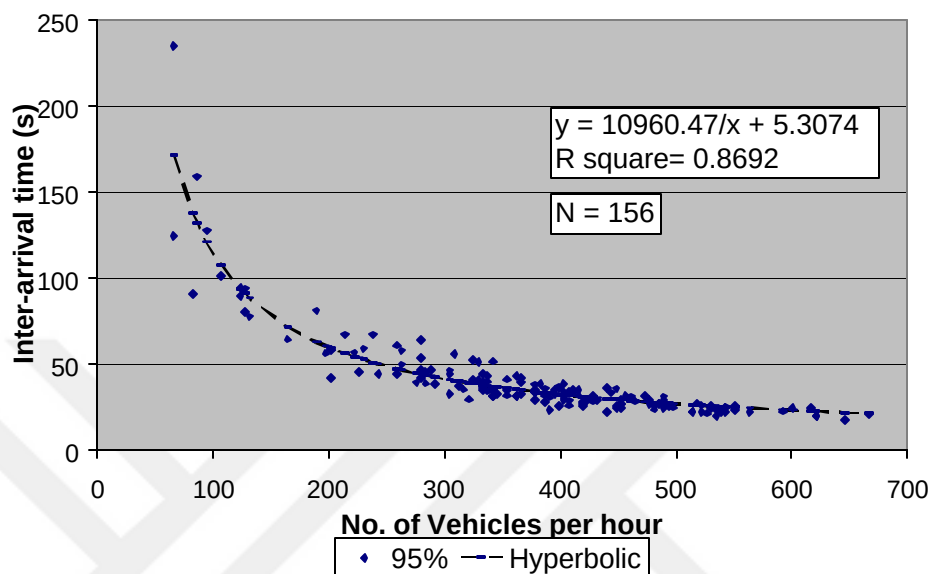
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 70%



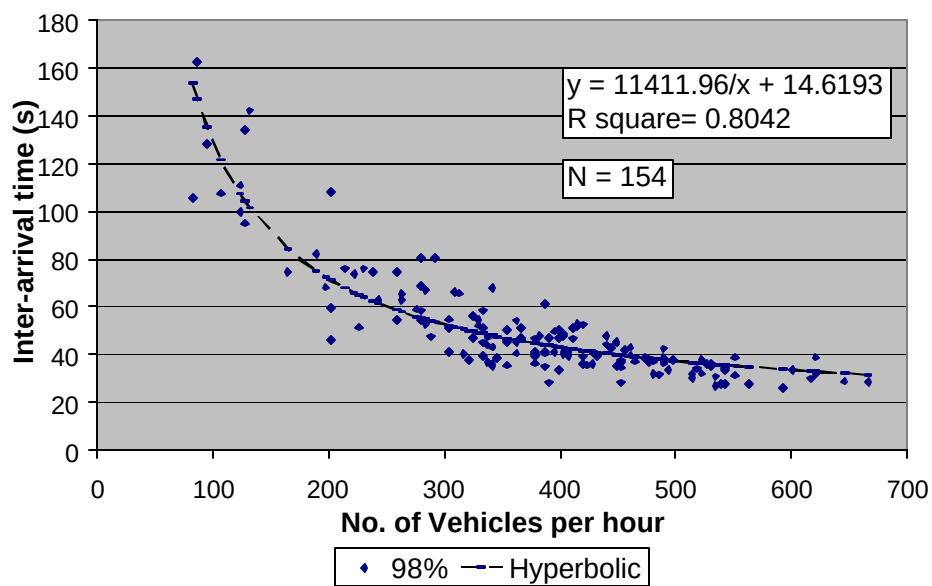
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 80%



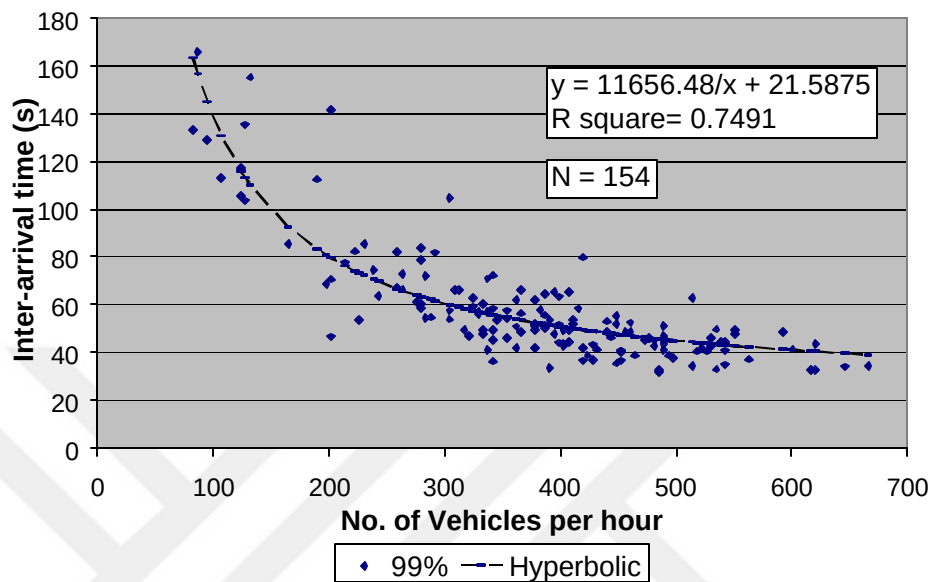
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 90%



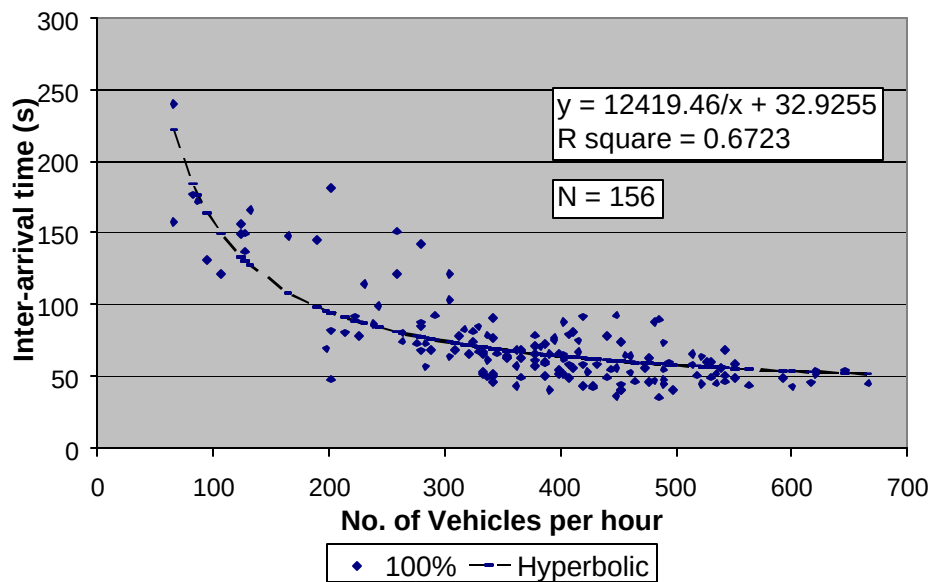
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 95%



IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 98%



IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 99%



IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 100% (Maximum)

**Appendix C. Calculated Inter-arrival Times for 15 Minute Intervals for 24-hour
Time Period**



**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane for
ARENA Simulation Model**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100% (Max)	
0:00-0:15	0.10	1.04	1.38	1.70	2.78	4.52	6.13	8.09	10.68	14.19	18.51	23.42	32.78	44.73	51.92	56.65	65.19	241
0:15-0:30	0.10	1.15	1.54	1.87	3.19	5.30	7.22	9.57	12.70	16.98	22.24	28.11	39.38	54.15	62.39	67.81	76.95	200
0:30-0:45	0.10	1.11	1.47	1.80	3.03	5.00	6.80	8.99	11.91	15.90	20.78	26.28	36.80	50.48	58.30	63.46	72.36	216
0:45-1:00	0.10	1.11	1.47	1.80	3.03	5.00	6.80	8.99	11.91	15.90	20.78	26.28	36.80	50.48	58.30	63.46	72.36	216
1:00-1:15	0.10	1.43	1.96	2.33	4.25	7.33	10.06	13.42	17.95	24.24	31.96	40.30	56.55	78.68	89.64	96.87	107.51	136
1:15-1:30	0.10	1.29	1.75	2.10	3.72	6.31	8.64	11.50	15.33	20.61	27.10	34.21	47.96	66.41	76.02	82.34	92.23	168
1:30-1:45	0.10	1.31	1.77	2.13	3.79	6.44	8.82	11.74	15.65	21.07	27.71	34.97	49.04	67.95	77.72	84.16	94.14	164
1:45-2:00	0.10	1.11	1.47	1.80	3.03	5.00	6.80	8.99	11.91	15.90	20.78	26.28	36.80	50.48	58.30	63.46	72.36	216
2:00-2:15	0.10	1.24	1.67	2.01	3.52	5.93	8.11	10.77	14.34	19.25	25.28	31.92	44.75	61.82	70.91	76.89	86.50	180
2:15-2:30	0.10	1.05	1.39	1.71	2.82	4.60	6.24	8.23	10.88	14.47	18.87	23.88	33.42	45.65	52.94	57.74	66.34	237
2:30-2:45	0.10	1.12	1.49	1.82	3.07	5.08	6.90	9.14	12.11	16.17	21.15	26.74	37.45	51.40	59.33	64.55	73.51	212
2:45-3:00	0.10	1.31	1.77	2.13	3.79	6.44	8.82	11.74	15.65	21.07	27.71	34.97	49.04	67.95	77.72	84.16	94.14	164
3:00-3:15	0.10	1.34	1.83	2.18	3.92	6.69	9.18	12.22	16.31	21.97	28.92	36.49	51.18	71.01	81.13	87.79	97.96	156
3:15-3:30	0.10	1.17	1.56	1.90	3.25	5.43	7.40	9.81	13.03	17.44	22.85	28.87	40.45	55.69	64.09	69.63	78.86	196
3:30-3:45	0.10	1.33	1.80	2.16	3.85	6.57	9.00	11.98	15.98	21.52	28.31	35.73	50.11	69.48	79.42	85.97	96.05	160
3:45-4:00	0.10	1.15	1.54	1.87	3.19	5.30	7.22	9.57	12.70	16.98	22.24	28.11	39.38	54.15	62.39	67.81	76.95	200
4:00-4:15	0.10	1.02	1.34	1.66	2.70	4.37	5.92	7.80	10.28	13.65	17.78	22.51	31.49	42.89	49.87	54.47	62.90	249
4:15-4:30	0.10	1.20	1.62	1.96	3.39	5.68	7.76	10.29	13.68	18.35	24.06	30.40	42.60	58.75	67.50	73.26	82.68	188
4:30-4:45	0.10	1.11	1.47	1.80	3.03	5.00	6.80	8.99	11.91	15.90	20.78	26.28	36.80	50.48	58.30	63.46	72.36	216
4:45-5:00	0.10	1.02	1.34	1.66	2.70	4.37	5.92	7.80	10.28	13.65	17.78	22.51	31.49	42.89	49.87	54.47	62.90	249

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane for
ARENA Simulation Model (continued)**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%	
																	(Max)	
5:00-5:15	0.10	1.04	1.38	1.70	2.78	4.52	6.13	8.09	10.68	14.19	18.51	23.42	32.78	44.73	51.92	56.65	65.19	241
5:15-5:30	0.10	0.85	1.09	1.38	2.04	3.14	4.18	5.45	7.08	9.23	11.87	15.09	21.05	27.99	33.30	36.79	44.26	369
5:30-5:45	0.10	0.84	1.07	1.36	2.00	3.06	4.07	5.29	6.87	8.94	11.48	14.60	20.36	27.01	32.20	35.62	43.03	381
5:45-6:00	0.10	0.92	1.18	1.49	2.29	3.60	4.83	6.33	8.29	10.89	14.09	17.88	24.97	33.59	39.53	43.43	51.27	313
6:00-6:15	0.10	0.80	1.02	1.31	1.87	2.82	3.73	4.84	6.25	8.08	10.33	13.16	18.33	24.11	28.98	32.18	39.40	421
6:15-6:30	0.10	0.73	0.90	1.18	1.59	2.28	2.98	3.82	4.86	6.17	7.78	9.96	13.82	17.69	21.82	24.54	31.31	549
6:30-6:45	0.10	0.68	0.83	1.10	1.41	1.95	2.51	3.19	4.01	5.01	6.22	8.00	11.07	13.78	17.46	19.88	26.35	677
6:45-7:00	0.10	0.72	0.89	1.17	1.56	2.23	2.91	3.73	4.74	6.01	7.56	9.68	13.42	17.13	21.20	23.87	30.60	565
7:00-7:15	0.10	0.70	0.86	1.13	1.48	2.08	2.70	3.44	4.34	5.46	6.83	8.77	12.14	15.30	19.16	21.70	28.28	621
7:15-7:30	0.10	0.66	0.81	1.08	1.36	1.87	2.40	3.04	3.80	4.72	5.84	7.52	10.39	12.82	16.38	18.72	25.11	718
7:30-7:45	0.10	0.67	0.82	1.09	1.39	1.90	2.45	3.11	3.90	4.85	6.01	7.74	10.70	13.26	16.87	19.25	25.68	698
7:45-8:00	0.10	0.70	0.86	1.14	1.49	2.10	2.72	3.48	4.39	5.53	6.92	8.88	12.30	15.54	19.42	21.97	28.58	613
8:00-8:15	0.10	0.68	0.83	1.10	1.41	1.95	2.51	3.19	4.01	5.01	6.22	8.00	11.07	13.78	17.46	19.88	26.35	677
8:15-8:30	0.10	0.68	0.83	1.10	1.41	1.95	2.51	3.19	4.01	5.01	6.22	8.00	11.07	13.78	17.46	19.88	26.35	677
8:30-8:45	0.10	0.69	0.85	1.12	1.46	2.04	2.64	3.36	4.24	5.32	6.64	8.53	11.81	14.84	18.64	21.14	27.69	637
8:45-9:00	0.10	0.71	0.88	1.15	1.53	2.17	2.83	3.62	4.59	5.80	7.28	9.33	12.94	16.44	20.42	23.05	29.72	585
9:00-9:15	0.10	0.70	0.86	1.14	1.49	2.11	2.74	3.49	4.42	5.57	6.97	8.94	12.39	15.66	19.55	22.11	28.73	609
9:15-9:30	0.10	0.71	0.88	1.15	1.53	2.17	2.83	3.62	4.59	5.80	7.28	9.33	12.94	16.44	20.42	23.05	29.72	585
9:30-9:45	0.10	0.68	0.84	1.11	1.44	2.00	2.59	3.29	4.14	5.19	6.46	8.30	11.49	14.38	18.13	20.59	27.11	653
9:45-10:00	0.10	0.70	0.87	1.15	1.51	2.14	2.78	3.55	4.50	5.68	7.12	9.12	12.64	16.02	19.96	22.55	29.19	597
10:00-10:15	0.10	0.71	0.87	1.15	1.52	2.15	2.80	3.58	4.53	5.72	7.17	9.19	12.74	16.16	20.12	22.72	29.37	593

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane
for ARENA Simulation Model (continued)**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%	
																	(Max)	
10.15-10.30	0.10	0.68	0.84	1.11	1.43	1.98	2.56	3.26	4.10	5.13	6.38	8.20	11.35	14.18	17.91	20.35	26.86	661
10.30-10.45	0.10	0.66	0.80	1.07	1.34	1.83	2.35	2.97	3.71	4.59	5.67	7.30	10.08	12.38	15.89	18.20	24.56	738
10.45-11.00	0.10	0.67	0.83	1.10	1.40	1.94	2.50	3.18	3.99	4.98	6.18	7.95	11.00	13.68	17.35	19.76	26.22	681
11.00-11.15	0.10	0.66	0.81	1.08	1.36	1.85	2.38	3.01	3.77	4.67	5.77	7.43	10.27	12.64	16.19	18.51	24.89	726
11.15-11.30	0.10	0.70	0.87	1.15	1.51	2.14	2.78	3.55	4.50	5.68	7.12	9.12	12.64	16.02	19.96	22.55	29.19	597
11.30-11.45	0.10	0.68	0.84	1.11	1.43	1.99	2.57	3.27	4.12	5.16	6.42	8.25	11.42	14.28	18.02	20.47	26.98	657
11.45-12.00	0.10	0.66	0.81	1.08	1.36	1.85	2.38	3.01	3.77	4.67	5.77	7.43	10.27	12.64	16.19	18.51	24.89	726
12.00-12.15	0.10	0.70	0.86	1.14	1.49	2.10	2.72	3.48	4.39	5.53	6.92	8.88	12.30	15.54	19.42	21.97	28.58	613
12.15-12.30	0.10	0.67	0.83	1.10	1.40	1.94	2.50	3.18	3.99	4.98	6.18	7.95	11.00	13.68	17.35	19.76	26.22	681
12.30-12.45	0.10	0.66	0.81	1.08	1.37	1.87	2.41	3.05	3.82	4.74	5.87	7.56	10.45	12.90	16.48	18.83	25.22	714
12.45-13.00	0.10	0.67	0.83	1.10	1.40	1.94	2.50	3.18	3.99	4.98	6.18	7.95	11.00	13.68	17.35	19.76	26.22	681
13.00-13.15	0.10	0.69	0.85	1.12	1.45	2.02	2.61	3.33	4.19	5.25	6.55	8.41	11.65	14.60	18.38	20.86	27.39	645
13.15-13.30	0.10	0.67	0.83	1.10	1.40	1.94	2.50	3.18	3.99	4.98	6.18	7.95	11.00	13.68	17.35	19.76	26.22	681
13.30-13.45	0.10	0.71	0.88	1.15	1.53	2.17	2.83	3.62	4.59	5.80	7.28	9.33	12.94	16.44	20.42	23.05	29.72	585
13.45-14.00	0.10	0.67	0.82	1.08	1.38	1.89	2.43	3.08	3.86	4.80	5.94	7.65	10.57	13.08	16.67	19.03	25.45	706
14.00-14.15	0.10	0.67	0.82	1.09	1.39	1.92	2.48	3.14	3.94	4.91	6.09	7.84	10.84	13.46	17.09	19.49	25.93	690
14.15-14.30	0.10	0.66	0.81	1.08	1.36	1.85	2.38	3.01	3.77	4.67	5.77	7.43	10.27	12.64	16.19	18.51	24.89	726
14.30-14.45	0.10	0.67	0.82	1.09	1.38	1.89	2.44	3.10	3.88	4.82	5.98	7.69	10.63	13.16	16.77	19.14	25.56	702
14.45-15.00	0.10	0.67	0.82	1.08	1.38	1.89	2.43	3.08	3.86	4.80	5.94	7.65	10.57	13.08	16.67	19.03	25.45	706
15.00-15.15	0.10	0.67	0.82	1.08	1.38	1.89	2.43	3.08	3.86	4.80	5.94	7.65	10.57	13.08	16.67	19.03	25.45	706

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane
for ARENA Simulation Model (continued)**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%	
																	(Max)	
15.15-15.30	0.10	0.66	0.80	1.07	1.34	1.83	2.35	2.97	3.71	4.59	5.67	7.30	10.08	12.38	15.89	18.20	24.56	738
15.30-15.45	0.10	0.66	0.81	1.08	1.36	1.85	2.38	3.01	3.77	4.67	5.77	7.43	10.27	12.64	16.19	18.51	24.89	726
15.45-16.00	0.10	0.66	0.81	1.08	1.37	1.87	2.41	3.05	3.82	4.74	5.87	7.56	10.45	12.90	16.48	18.83	25.22	714
16.00-16.15	0.10	0.70	0.87	1.14	1.51	2.13	2.76	3.53	4.47	5.64	7.06	9.06	12.55	15.89	19.81	22.39	29.03	601
16.15-16.30	0.10	0.66	0.80	1.07	1.34	1.81	2.33	2.94	3.67	4.54	5.60	7.22	9.96	12.21	15.70	18.00	24.34	746
16.30-16.45	0.10	0.65	0.79	1.06	1.33	1.79	2.30	2.90	3.62	4.47	5.50	7.10	9.79	11.97	15.43	17.71	24.03	758
16.45-17.00	0.10	0.66	0.80	1.07	1.34	1.82	2.34	2.96	3.69	4.56	5.63	7.26	10.02	12.30	15.80	18.10	24.45	742
17.00-17.15	0.10	0.68	0.84	1.11	1.43	1.98	2.56	3.26	4.10	5.13	6.38	8.20	11.35	14.18	17.91	20.35	26.86	661
17.15-17.30	0.10	0.67	0.82	1.09	1.39	1.92	2.48	3.14	3.94	4.91	6.09	7.84	10.84	13.46	17.09	19.49	25.93	690
17.30-17.45	0.10	0.67	0.82	1.09	1.39	1.91	2.46	3.13	3.92	4.88	6.05	7.79	10.77	13.36	16.98	19.37	25.80	694
17.45-18.00	0.10	0.70	0.86	1.14	1.49	2.11	2.74	3.49	4.42	5.57	6.97	8.94	12.39	15.66	19.55	22.11	28.73	609
18.00-18.15	0.10	0.69	0.85	1.12	1.46	2.04	2.64	3.36	4.24	5.32	6.64	8.53	11.81	14.84	18.64	21.14	27.69	637
18.15-18.30	0.10	0.72	0.89	1.17	1.56	2.23	2.91	3.73	4.74	6.01	7.56	9.68	13.42	17.13	21.20	23.87	30.60	565
18.30-18.45	0.10	0.71	0.88	1.16	1.54	2.20	2.86	3.66	4.65	5.88	7.39	9.47	13.13	16.71	20.73	23.38	30.07	577
18.45-19.00	0.10	0.72	0.89	1.17	1.56	2.23	2.91	3.73	4.74	6.01	7.56	9.68	13.42	17.13	21.20	23.87	30.60	565
19.00-19.15	0.10	0.72	0.90	1.17	1.57	2.25	2.94	3.77	4.80	6.09	7.67	9.82	13.62	17.41	21.51	24.20	30.95	557
19.15-19.30	0.10	0.74	0.92	1.20	1.64	2.38	3.11	4.00	5.11	6.52	8.24	10.54	14.64	18.85	23.12	25.93	32.77	521
19.30-19.45	0.10	0.76	0.95	1.23	1.70	2.50	3.28	4.23	5.42	6.94	8.81	11.25	15.64	20.29	24.72	27.63	34.58	489
19.45-20.00	0.10	0.76	0.96	1.24	1.72	2.53	3.33	4.30	5.51	7.07	8.97	11.46	15.93	20.69	25.17	28.12	35.09	481
20.00-20:15	0.10	0.77	0.97	1.26	1.76	2.60	3.42	4.42	5.69	7.31	9.30	11.86	16.50	21.51	26.08	29.08	36.12	465

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane
for ARENA Simulation Model (continued)**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%	
																	(Max)	
20:15-20:30	0.10	0.81	1.02	1.31	1.89	2.85	3.77	4.90	6.33	8.20	10.48	13.35	18.60	24.50	29.41	32.64	39.88	415
20:30-20:45	0.10	0.77	0.96	1.24	1.73	2.55	3.36	4.34	5.56	7.14	9.07	11.58	16.11	20.95	25.45	28.42	35.41	476
20:45-21:00	0.10	0.79	1.00	1.29	1.83	2.74	3.62	4.69	6.04	7.80	9.95	12.69	17.66	23.16	27.92	31.05	38.20	436
21:00-21:15	0.10	0.77	0.96	1.25	1.74	2.57	3.38	4.37	5.61	7.20	9.15	11.68	16.25	21.15	25.68	28.66	35.67	472
21:15-21:30	0.10	0.83	1.05	1.35	1.97	2.99	3.97	5.17	6.69	8.70	11.15	14.19	19.78	26.19	31.29	34.65	42.00	391
21:30-21:45	0.10	0.79	1.00	1.28	1.82	2.71	3.59	4.65	5.99	7.72	9.85	12.56	17.48	22.90	27.64	30.75	37.88	440
21:45-22:00	0.10	0.85	1.08	1.38	2.04	3.12	4.16	5.42	7.05	9.18	11.80	15.01	20.93	27.82	33.11	36.59	44.05	371
22:00-22:15	0.10	0.91	1.18	1.48	2.28	3.58	4.81	6.30	8.24	10.83	14.00	17.77	24.82	33.37	39.28	43.17	51.00	315
22:15-22:30	0.10	0.85	1.09	1.38	2.05	3.15	4.20	5.48	7.12	9.28	11.93	15.17	21.16	28.15	33.48	36.98	44.46	367
22:30-22:45	0.10	0.90	1.16	1.47	2.24	3.51	4.71	6.16	8.05	10.57	13.66	17.34	24.21	32.49	38.31	42.14	49.90	323
22:45-23:00	0.10	0.92	1.19	1.49	2.30	3.62	4.86	6.37	8.33	10.96	14.18	17.99	25.13	33.81	39.77	43.69	51.54	311
23:00-23:15	0.10	1.00	1.31	1.62	2.61	4.20	5.68	7.47	9.84	13.04	16.96	21.48	30.04	40.83	47.58	52.02	60.32	262
23:15-23:30	0.10	0.91	1.18	1.48	2.28	3.58	4.81	6.30	8.24	10.83	14.00	17.77	24.82	33.37	39.28	43.17	51.00	315
23:30-23:45	0.10	0.91	1.18	1.48	2.28	3.58	4.81	6.30	8.24	10.83	14.00	17.77	24.82	33.37	39.28	43.17	51.00	315
23:45-0:00	0.10	1.03	1.36	1.68	2.73	4.43	6.00	7.91	10.43	13.85	18.05	22.85	31.98	43.58	50.64	55.29	63.76	246

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane
for ARENA Simulation Model**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100% (Max)	
0:00-0:15	0.11	0.78	1.22	1.93	3.31	10.33	17.38	28.61	44.50	71.06	99.31	144.40	217.17	257.30	276.56	289.48	318.85	41
0:15-0:30	0.11	0.77	1.19	1.88	3.21	9.96	16.73	27.54	42.83	68.38	95.60	139.02	209.13	247.65	266.90	279.61	308.33	45
0:30-0:45	0.10	0.75	1.14	1.78	3.01	9.21	15.44	25.39	39.49	63.03	88.18	128.27	193.07	228.34	247.58	259.87	287.29	53
0:45-1:00	0.11	0.78	1.22	1.93	3.31	10.33	17.38	28.61	44.50	71.06	99.31	144.40	217.17	257.30	276.56	289.48	318.85	41
1:00-1:15	0.11	0.80	1.27	2.03	3.50	11.08	18.67	30.75	47.84	76.41	106.73	155.15	233.23	276.61	295.87	309.21	339.89	33
1:15-1:30	0.11	0.83	1.34	2.18	3.80	12.21	20.60	33.96	52.85	84.44	117.86	171.28	257.33	305.57	324.85	338.82	371.46	21
1:30-1:45	0.11	0.78	1.22	1.93	3.31	10.33	17.38	28.61	44.50	71.06	99.31	144.40	217.17	257.30	276.56	289.48	318.85	41
1:45-2:00	0.10	0.76	1.17	1.83	3.11	9.58	16.09	26.47	41.16	65.71	91.89	133.65	201.10	237.99	257.24	269.74	297.81	49
2:00-2:15	0.11	0.79	1.24	1.98	3.40	10.71	18.02	29.68	46.17	73.73	103.02	149.78	225.20	266.95	286.22	299.35	329.37	37
2:15-2:30	0.10	0.73	1.11	1.72	2.88	8.74	14.63	24.06	37.40	59.68	83.54	121.55	183.03	216.27	235.51	247.53	274.13	58
2:30-2:45	0.11	0.77	1.19	1.88	3.21	9.96	16.73	27.54	42.83	68.38	95.60	139.02	209.13	247.65	266.90	279.61	308.33	45
2:45-3:00	0.11	0.78	1.22	1.93	3.31	10.33	17.38	28.61	44.50	71.06	99.31	144.40	217.17	257.30	276.56	289.48	318.85	41
3:00-3:15	0.11	0.83	1.34	2.18	3.80	12.21	20.60	33.96	52.85	84.44	117.86	171.28	257.33	305.57	324.85	338.82	371.46	21
3:15-3:30	0.11	0.81	1.29	2.08	3.60	11.46	19.31	31.82	49.51	79.09	110.44	160.53	241.26	286.26	305.53	319.08	350.42	29
3:30-3:45	0.11	0.78	1.22	1.93	3.31	10.33	17.38	28.61	44.50	71.06	99.31	144.40	217.17	257.30	276.56	289.48	318.85	41
3:45-4:00	0.11	0.79	1.24	1.98	3.40	10.71	18.02	29.68	46.17	73.73	103.02	149.78	225.20	266.95	286.22	299.35	329.37	37
4:00-4:15	0.11	0.80	1.27	2.03	3.50	11.08	18.67	30.75	47.84	76.41	106.73	155.15	233.23	276.61	295.87	309.21	339.89	33
4:15-4:30	0.11	0.77	1.19	1.88	3.21	9.96	16.73	27.54	42.83	68.38	95.60	139.02	209.13	247.65	266.90	279.61	308.33	45
4:30-4:45	0.10	0.76	1.17	1.83	3.11	9.58	16.09	26.47	41.16	65.71	91.89	133.65	201.10	237.99	257.24	269.74	297.81	49
4:45-5:00	0.10	0.69	1.01	1.52	2.49	7.24	12.05	19.77	30.72	48.98	68.70	100.05	150.90	177.66	196.87	208.05	232.05	74
5:00-5:15	0.11	0.79	1.24	1.98	3.40	10.71	18.02	29.68	46.17	73.73	103.02	149.78	225.20	266.95	286.22	299.35	329.37	37

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane
for ARENA Simulation Model (continued)**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100% (Max)	
5:15-5:30	0.10	0.62	0.84	1.19	1.82	4.70	7.70	12.55	19.44	30.91	43.65	63.77	96.69	112.88	131.67	141.42	161.01	103
5:30-5:45	0.10	0.57	0.74	0.99	1.42	3.20	5.12	8.27	12.77	20.22	28.82	42.28	64.57	80.27	92.98	101.85	118.77	152
5:45-6:00	0.10	0.62	0.83	1.17	1.79	4.58	7.49	12.19	18.89	30.02	42.42	61.98	94.01	110.17	128.45	138.13	157.49	107
6:00-6:15	0.10	0.57	0.73	0.97	1.39	3.07	4.91	7.91	12.21	19.33	27.59	40.49	61.90	77.18	89.74	98.54	115.24	160
6:15-6:30	0.10	0.52	0.61	0.76	0.96	1.51	2.24	3.49	5.31	8.28	12.26	18.26	28.66	38.68	49.44	57.20	70.94	333
6:30-6:45	0.10	0.50	0.58	0.70	0.86	1.14	1.61	2.44	3.68	5.67	8.63	13.00	20.78	29.50	39.72	47.17	60.10	498
6:45-7:00	0.10	0.52	0.62	0.77	0.98	1.57	2.33	3.64	5.55	8.66	12.78	19.03	29.81	40.01	50.84	58.64	72.49	321
7:00-7:15	0.10	0.51	0.60	0.73	0.91	1.33	1.93	2.97	4.50	6.99	10.46	15.65	24.75	34.14	44.64	52.25	65.60	382
7:15-7:30	0.10	0.50	0.59	0.71	0.87	1.19	1.68	2.57	3.88	5.98	9.06	13.62	21.71	30.59	40.89	48.38	61.41	477
7:30-7:45	0.10	0.49	0.57	0.68	0.83	1.05	1.44	2.17	3.26	4.99	7.69	11.63	18.72	27.09	37.16	44.51	57.20	543
7:45-8:00	0.10	0.50	0.58	0.68	0.84	1.09	1.52	2.29	3.45	5.29	8.10	12.24	19.63	28.16	38.30	45.70	58.50	518
8:00-8:15	0.10	0.51	0.60	0.73	0.91	1.33	1.92	2.96	4.49	6.96	10.43	15.61	24.69	34.06	44.57	52.18	65.53	428
8:15-8:30	0.10	0.51	0.60	0.73	0.91	1.34	1.95	3.01	4.56	7.08	10.59	15.84	25.04	34.47	44.99	52.61	65.99	378
8:30-8:45	0.10	0.51	0.60	0.73	0.90	1.30	1.88	2.88	4.37	6.77	10.16	15.22	24.11	33.39	43.85	51.43	64.72	391
8:45-9:00	0.10	0.52	0.61	0.76	0.96	1.51	2.24	3.49	5.31	8.28	12.26	18.26	28.66	38.68	49.44	57.20	70.94	333
9:00-9:15	0.10	0.52	0.62	0.77	0.99	1.60	2.39	3.74	5.71	8.91	13.13	19.54	30.57	40.90	51.77	59.60	73.53	313
9:15-9:30	0.10	0.51	0.60	0.74	0.93	1.39	2.03	3.14	4.78	7.42	11.06	16.54	26.08	35.69	46.29	53.96	67.46	403
9:30-9:45	0.10	0.51	0.60	0.73	0.90	1.31	1.90	2.92	4.43	6.87	10.29	15.42	24.40	33.72	44.20	51.80	65.11	387
9:45-10:00	0.10	0.53	0.64	0.80	1.05	1.81	2.75	4.33	6.63	10.39	15.18	22.51	35.02	46.06	57.19	65.18	79.52	276
10:00-10:15	0.10	0.51	0.60	0.74	0.93	1.38	2.02	3.11	4.73	7.35	10.96	16.39	25.85	35.43	46.01	53.68	67.15	407
10:15-10:30	0.10	0.51	0.60	0.73	0.90	1.31	1.90	2.92	4.43	6.87	10.29	15.42	24.40	33.72	44.20	51.80	65.11	387

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane
for ARENA Simulation Model (continued)**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%	
																	(Max)	
10.30-10.45	0.10	0.50	0.58	0.70	0.86	1.16	1.64	2.50	3.77	5.80	8.81	13.27	21.18	29.96	40.22	47.69	60.66	489
10.45-11.00	0.10	0.51	0.59	0.72	0.89	1.28	1.83	2.81	4.26	6.60	9.92	14.87	23.58	32.77	43.19	50.76	63.99	448
11.00-11.15	0.10	0.49	0.57	0.68	0.83	1.07	1.48	2.23	3.36	5.15	7.90	11.95	19.19	27.64	37.76	45.13	57.88	530
11.15-11.30	0.10	0.51	0.60	0.73	0.91	1.34	1.95	3.01	4.56	7.08	10.59	15.84	25.04	34.47	44.99	52.61	65.99	378
11.30-11.45	0.10	0.50	0.59	0.72	0.89	1.27	1.83	2.81	4.25	6.58	9.90	14.84	23.54	32.72	43.14	50.71	63.93	399
11.45-12.00	0.10	0.50	0.59	0.71	0.88	1.23	1.76	2.69	4.07	6.29	9.49	14.24	22.64	31.68	42.05	49.59	62.72	456
12.00-12.15	0.10	0.51	0.60	0.74	0.93	1.38	2.02	3.12	4.74	7.36	10.98	16.42	25.90	35.47	46.05	53.70	67.17	366
12.15-12.30	0.10	0.50	0.58	0.70	0.86	1.17	1.65	2.52	3.80	5.86	8.89	13.39	21.35	30.17	40.44	47.92	60.91	485
12.30-12.45	0.10	0.49	0.57	0.71	0.82	1.02	1.40	2.09	3.14	4.80	7.42	11.24	18.13	26.41	36.44	43.77	56.40	551
12.45-13.00	0.10	0.50	0.58	0.68	0.84	1.09	1.52	2.29	3.45	5.29	8.10	12.24	19.63	28.16	38.30	45.70	58.50	518
13.00-13.15	0.10	0.51	0.60	0.73	0.92	1.35	1.96	3.03	4.59	7.13	10.66	15.94	25.19	34.65	45.19	52.82	66.23	419
13.15-13.30	0.10	0.51	0.60	0.74	0.93	1.38	2.02	3.11	4.73	7.35	10.96	16.39	25.85	35.43	46.01	53.68	67.15	407
13.30-13.45	0.10	0.51	0.60	0.74	0.93	1.39	2.03	3.14	4.78	7.42	11.06	16.54	26.08	35.69	46.29	53.96	67.46	403
13.45-14.00	0.10	0.49	0.58	0.68	0.84	1.08	1.50	2.27	3.42	5.25	8.04	12.14	19.48	27.99	38.12	45.51	58.29	522
14.00-14.15	0.10	0.50	0.59	0.72	0.88	1.24	1.77	2.71	4.10	6.34	9.57	14.36	22.82	31.89	42.27	49.81	62.97	452
14.15-14.30	0.10	0.48	0.55	0.65	0.77	0.85	1.11	1.63	2.42	3.64	5.80	8.89	14.61	22.25	31.98	39.11	51.28	666
14.30-14.45	0.10	0.49	0.57	0.68	0.83	1.05	1.44	2.17	3.26	4.99	7.69	11.63	18.72	27.09	37.16	44.51	57.20	543
14.45-15.00	0.10	0.49	0.57	0.70	0.81	1.00	1.37	2.04	3.06	4.68	7.25	11.00	17.77	25.98	35.98	43.29	55.87	563
15.00-15.15	0.10	0.49	0.57	0.68	0.83	1.06	1.47	2.21	3.32	5.09	7.82	11.82	19.01	27.43	37.53	44.89	57.62	535
15.15-15.30	0.10	0.49	0.57	0.68	0.84	1.08	1.49	2.25	3.39	5.20	7.97	12.04	19.34	27.82	37.94	45.32	58.09	526
15.30-15.45	0.10	0.50	0.58	0.68	0.84	1.10	1.53	2.31	3.48	5.34	8.17	12.33	19.77	28.33	38.49	45.89	58.71	514

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane
for ARENA Simulation Model (continued)**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100%	
																	(Max)	
15.45-16.00	0.10	0.49	0.57	0.68	0.83	1.05	1.46	2.19	3.29	5.04	7.75	11.73	18.87	27.26	37.35	44.70	57.41	539
16.00-16.15	0.10	0.50	0.59	0.72	0.88	1.24	1.77	2.71	4.10	6.34	9.57	14.36	22.82	31.89	42.27	49.81	62.97	452
16.15-16.30	0.10	0.49	0.57	0.68	0.83	1.07	1.48	2.23	3.36	5.15	7.90	11.95	19.19	27.64	37.76	45.13	57.88	530
16.30-16.45	0.10	0.48	0.56	0.66	0.78	0.91	1.22	1.80	2.68	4.06	6.38	9.74	15.88	23.76	33.60	40.80	53.14	621
16.45-17.00	0.10	0.50	0.58	0.71	0.87	1.18	1.67	2.54	3.84	5.92	8.98	13.50	21.53	30.38	40.67	48.15	61.16	481
17.00-17.15	0.10	0.51	0.60	0.72	0.90	1.30	1.87	2.87	4.35	6.74	10.12	15.17	24.02	33.28	43.74	51.33	64.60	440
17.15-17.30	0.10	0.50	0.58	0.71	0.87	1.18	1.67	2.54	3.84	5.92	8.98	13.50	21.53	30.38	40.67	48.15	61.16	481
17.30-17.45	0.10	0.51	0.59	0.72	0.90	1.29	1.85	2.84	4.31	6.67	10.02	15.02	23.80	33.03	43.47	51.04	64.30	444
17.45-18.00	0.10	0.50	0.59	0.71	0.88	1.21	1.73	2.64	3.99	6.15	9.30	13.98	22.24	31.21	41.55	49.07	62.16	465
18.00-18.15	0.10	0.51	0.60	0.73	0.91	1.32	1.90	2.93	4.45	6.89	10.32	15.46	24.47	33.80	44.29	51.90	65.22	432
18.15-18.30	0.10	0.51	0.60	0.72	0.90	1.30	1.87	2.87	4.35	6.74	10.12	15.17	24.02	33.28	43.74	51.33	64.60	440
18.30-18.45	0.10	0.50	0.58	0.70	0.86	1.16	1.64	2.50	3.77	5.80	8.81	13.27	21.18	29.96	40.22	47.69	60.66	489
18.45-19.00	0.10	0.52	0.62	0.76	0.97	1.55	2.30	3.59	5.47	8.53	12.61	18.77	29.43	39.57	50.37	58.16	71.98	325
19.00-19.15	0.10	0.52	0.62	0.78	1.00	1.64	2.46	3.86	5.89	9.20	13.53	20.11	31.43	41.90	52.82	60.69	74.70	304
19.15-19.30	0.10	0.52	0.62	0.77	0.98	1.59	2.36	3.69	5.63	8.79	12.96	19.28	30.19	40.46	51.31	59.12	73.01	317
19.30-19.45	0.10	0.53	0.64	0.80	1.04	1.79	2.71	4.26	6.52	10.21	14.94	22.15	34.48	45.44	56.54	64.51	78.80	280
19.45-20.00	0.10	0.52	0.61	0.76	0.96	1.51	2.24	3.49	5.31	8.28	12.26	18.26	28.66	38.68	49.44	57.20	70.94	333
20.00-20:15	0.10	0.53	0.64	0.80	1.04	1.79	2.71	4.26	6.52	10.21	14.94	22.15	34.48	45.44	56.54	64.51	78.80	280
20:15-20:30	0.10	0.56	0.70	0.91	1.26	2.62	4.13	6.62	10.20	16.10	23.11	34.00	52.20	65.96	78.03	86.54	102.41	189
20:30-20:45	0.10	0.57	0.73	0.97	1.39	3.07	4.91	7.91	12.21	19.33	27.59	40.49	61.90	77.18	89.74	98.54	115.24	160
20:45-21:00	0.10	0.61	0.82	1.14	1.72	4.32	7.06	11.48	17.78	28.24	39.95	58.40	88.66	104.75	122.00	131.53	150.46	115

**Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane
for ARENA Simulation Model (continued)**

Time	Cumulative percentage																	No of Veh/hr
	0%	1%	2%	5%	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	98%	99%	100% (Max)	
21:00-21:15	0.10	0.54	0.67	0.87	1.17	2.28	3.54	5.65	8.68	13.67	19.75	29.12	44.91	57.53	69.21	77.50	92.74	218
21:15-21:30	0.10	0.55	0.68	0.88	1.20	2.39	3.74	5.97	9.18	14.47	20.86	30.73	47.32	60.31	72.12	80.50	95.94	206
21:30-21:45	0.10	0.59	0.78	1.07	1.58	3.79	6.14	9.96	15.41	24.45	34.69	50.78	77.29	93.24	108.30	117.52	135.50	132
21:45-22:00	0.10	0.56	0.71	0.93	1.30	2.74	4.34	6.98	10.75	16.99	24.35	35.79	54.87	69.05	81.26	89.85	105.95	181
22:00-22:15	0.10	0.57	0.73	0.98	1.40	3.14	5.02	8.09	12.49	19.77	28.20	41.38	63.24	78.72	91.36	100.19	117.01	156
22:15-22:30	0.10	0.60	0.80	1.11	1.65	4.07	6.63	10.76	16.66	26.46	37.47	54.81	83.31	99.33	115.55	124.94	143.42	123
22:30-22:45	0.10	0.62	0.83	1.17	1.79	4.58	7.49	12.19	18.89	30.02	42.42	61.98	94.01	110.17	128.45	138.13	157.49	107
22:45-23:00	0.10	0.72	1.09	1.67	2.79	8.36	13.99	22.99	35.73	57.01	79.83	116.18	175.00	206.62	225.85	237.66	263.61	62
23:00-23:15	0.10	0.67	0.96	1.42	2.29	6.48	10.76	17.63	27.38	43.62	61.28	89.30	134.84	158.35	177.56	188.31	211.00	82
23:15-23:30	0.10	0.72	1.09	1.67	2.79	8.36	13.99	22.99	35.73	57.01	79.83	116.18	175.00	206.62	225.85	237.66	263.61	62
23:30-23:45	0.10	0.65	0.91	1.33	2.09	5.73	9.48	15.49	24.04	38.27	53.86	78.55	118.78	139.05	158.24	168.57	189.96	90
23:45-0:00	0.10	0.69	1.01	1.52	2.49	7.24	12.05	19.77	30.72	48.98	68.70	100.05	150.90	177.66	196.87	208.05	232.05	74

Appendix D. ARENA Simulation Model SIMAN Code

SIMAN Codes .mod file

```
RequestLCar REQUEST, 1:LAGV(LCar#),1000000,LStation5750;
TransportLCarfrom0 TRANSPORT: LAGV,LStation6450;
```

```
RequestRCar REQUEST, 1:RAGV(RCar#),1000000,RStation5750;
TransportRCarfrom0 TRANSPORT: RAGV,RStation6450;
```

```
;
;
; Model statements for module: Station 124
;
```

```
0$ STATION, LStation5750;
1645$ DELAY: 0.0,,VA:NEXT(4$);
```

```
;
;
; Model statements for module: Decide 1683
;
```

```
4$ BRANCH, 1:
    If,Entity.Type==LCar,1646$,Yes:
    Else,1647$,Yes;
1646$ ASSIGN: DecideVehicleTypeLeft0.NumberOut
True=DecideVehicleTypeLeft0.NumberOut True + 1:NEXT(16$);
```

```
1647$ ASSIGN: DecideVehicleTypeLeft0.NumberOut
False=DecideVehicleTypeLeft0.NumberOut False + 1:NEXT(17$);
```

```
;
;
; Model statements for module: Record 3
;
16$ COUNT: Number of Cars Entered to Left Lane,1:NEXT(12$);
```

```
;
;
; Model statements for module: Assign 242
;
12$ ASSIGN: LCar#=DecideVehicleTypeLeft0.NumberOut True:NEXT(14$);
```

```
;
;
; Model statements for module: Assign 244
;
14$ ASSIGN: VTU(LAGV,LCar#)=NORM(101,1):NEXT(RequestLCar);
```

```

;
;
; Model statements for module: Record 4
;
17$      COUNT:      Number of Trucks Entered to Left Lane,1:NEXT(13$);

;
;
; Model statements for module: Assign 243
;
13$      ASSIGN:      LTruck#=DecideVehicleTypeLeft0.NumberOut False:NEXT(15$);

;
;
; Model statements for module: Assign 245
;
15$      ASSIGN:      VTU(LAGVT,LTruck#)=NORM(101,1):NEXT(RequestLTruck);

RequestLTruck REQUEST,      1:LAGVT(LTruck#),1000000,LStation5750;
TransportLTruckfrom0 TRANSPORT: LAGVT,LStation6450;

;
;
; Model statements for module: Station 125
;

1$      STATION,      RStation5750;
1650$    DELAY:      0.0,,VA:NEXT(5$);

;
;
; Model statements for module: Decide 1684
;
5$      BRANCH,      1:
                If,Entity.Type==RCar,1651$,Yes:
                Else,1652$,Yes;
1651$    ASSIGN:      DecideVehicleTypeRight0.NumberOut
True=DecideVehicleTypeRight0.NumberOut True + 1:NEXT(18$);

1652$    ASSIGN:      DecideVehicleTypeRight0.NumberOut
False=DecideVehicleTypeRight0.NumberOut False + 1:NEXT(19$);

;
;
; Model statements for module: Record 5
;

```

```

18$      COUNT:      Number of Cars Entered to Right Lane,1:NEXT(8$);

;
;
; Model statements for module: Assign 238
;
8$      ASSIGN:      RCar#=DecideVehicleTypeRight0.NumberOut True:NEXT(10$);

;
;
; Model statements for module: Assign 240
;
10$     ASSIGN:      VTU(RAGV,RCar#)=NORM(90,2):NEXT(6$);

;
;
; Model statements for module: Assign 236
;
6$      ASSIGN:      SpeedatRStation5750=VT(RAGV):NEXT(RequestRCar);

;
;
; Model statements for module: Record 6
;
19$     COUNT:      Number of Trucks Entered to Right Lane,1:NEXT(9$);

;
;
; Model statements for module: Assign 239
;
9$      ASSIGN:      RTruck#=DecideVehicleTypeRight0.NumberOut False:NEXT(11$);

;
;
; Model statements for module: Assign 241
;
11$     ASSIGN:      VTU(RAGVT,RTruck#)=NORM(90,2):NEXT(7$);

;
;
; Model statements for module: Assign 237
;
7$      ASSIGN:      SpeedatRStation5750=VT(RAGVT):NEXT(RequestRTruck);

RequestRTruck REQUEST, 1:RAGVT(RTruck#),1000000,RStation5750;
TransportRTruckfrom0 TRANSPORT: RAGVT,RStation6450;

```



```

;
;
; Model statements for module: Assign 206
;
2$    ASSIGN:    Vehicle Index=DISC( 0.91,1 , 1.0, 2):
                Entity.Type=Entity Types ( Vehicle Index ):
                Entity.Picture=Part Pictures( Vehicle Index ):NEXT(0$);

;
;
; Model statements for module: Assign 207
;
3$    ASSIGN:    Vehicle Index=DISC( 0.61,3 , 1.0, 4):
                Entity.Type=Entity Types ( Vehicle Index ):
                Entity.Picture=Part Pictures( Vehicle Index ):NEXT(1$);

;
;
; Model statements for module: Create 336
;
1653$    CREATE,    1,SecondstoBaseTime(100),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.61,0.05,0.76,0.1,0.97,0.2,1.55,0.3,2.30,0.4,3.59,0.5,
5.47,0.6,8.53,0.7,12.60,0.8,18.76,0.9,29.39,0.95,39.48,0.98,50.20,0.99,57.93,1,71.65)),
                80:NEXT(1654$);

1654$    ASSIGN:    L.Vehicle Entry645_700.NumberOut=L.Vehicle Entry645_700.NumberOut +
1:NEXT(20$);

;
;
; Model statements for module: Decide 1688
;
20$    BRANCH,    1:
                If,Entity.CreateTime<1000,1657$,Yes:
                Else,1658$,Yes;
1657$    ASSIGN:    Decide 1688.NumberOut True=Decide 1688.NumberOut True + 1:NEXT(2$);
1658$    ASSIGN:    Decide 1688.NumberOut False=Decide 1688.NumberOut False + 1:NEXT(21$);

;
;
; Model statements for module: Dispose 9
;
21$    ASSIGN:    Dispose 9.NumberOut=Dispose 9.NumberOut + 1;
1659$    DISPOSE:    No;

```

```

;
;
;   Model statements for module: Create 337
;

1660$   CREATE,      1,SecondstoBaseTime(1000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.60,0.05,0.73,0.1,0.91,0.2,1.32,0.3,1.92,0.4,2.95,0.5,
4.48,0.6,6.94,0.7,10.39,0.8,15.56,0.9,24.61,0.95,33.97,0.98,44.46,0.99,52.07,1,65.40)),
          96:NEXT(1661$);

1661$   ASSIGN:      L.Vehicle Entry700_715.NumberOut=L.Vehicle Entry700_715.NumberOut +
1:NEXT(22$);

;
;
;   Model statements for module: Decide 1689
;
22$     BRANCH,      1:
          If,Entity.CreateTime<1900,1664$,Yes:
          Else,1665$,Yes;
1664$   ASSIGN:      Decide 1689.NumberOut True=Decide 1689.NumberOut True + 1:NEXT(2$);
1665$   ASSIGN:      Decide 1689.NumberOut False=Decide 1689.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 338
;

1666$   CREATE,      1,SecondstoBaseTime(1900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.70,0.1,0.85,0.2,1.09,0.3,1.52,0.4,2.29,0.5,
3.45,0.6,5.30,0.7,8.12,0.8,12.26,0.9,19.69,0.95,28.29,0.98,38.54,0.99,46.02,1,58.96)),
          119:NEXT(1667$);

1667$   ASSIGN:      L.Vehicle Entry715_73.NumberOut=L.Vehicle Entry715_73.NumberOut +
1:NEXT(23$);

;
;
;   Model statements for module: Decide 1690
;
23$     BRANCH,      1:
          If,Entity.CreateTime<2800,1670$,Yes:
          Else,1671$,Yes;
1670$   ASSIGN:      Decide 1690.NumberOut True=Decide 1690.NumberOut True + 1:NEXT(2$);

```

```

1671$    ASSIGN:    Decide 1690.NumberOut False=Decide 1690.NumberOut False + 1:NEXT(21$);

;
;
;    Model statements for module: Create 339
;

1672$    CREATE,    1,SecondstoBaseTime(2800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.68,0.1,0.82,0.2,0.98,0.3,1.33,0.4,1.97,0.5,
2.95,0.6,4.49,0.7,7.00,0.8,10.65,0.9,17.27,0.95,25.50,0.98,35.64,0.99,43.06,1,55.81)),
136:NEXT(1673$);

1673$    ASSIGN:    L.Vehicle Entry730_745.NumberOut=L.Vehicle Entry730_745.NumberOut +
1:NEXT(24$);

;
;
;    Model statements for module: Decide 1691
;
24$      BRANCH,    1:
                If,Entity.CreateTime<3700,1676$,Yes:
                Else,1677$,Yes;
1676$    ASSIGN:    Decide 1691.NumberOut True=Decide 1691.NumberOut True + 1:NEXT(2$);
1677$    ASSIGN:    Decide 1691.NumberOut False=Decide 1691.NumberOut False + 1:NEXT(21$);

;
;
;    Model statements for module: Create 340
;

1678$    CREATE,    1,SecondstoBaseTime(3700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.69,0.1,0.83,0.2,1.02,0.3,1.39,0.4,2.08,0.5,
3.12,0.6,4.77,0.7,7.39,0.8,11.20,0.9,18.10,0.95,26.46,0.98,36.45,0.99,44.09,1,56.90)),
130:NEXT(1679$);

1679$    ASSIGN:    L.Vehicle Entry745_800.NumberOut=L.Vehicle Entry745_800.NumberOut +
1:NEXT(25$);

;
;
;    Model statements for module: Decide 1692
;
25$      BRANCH,    1:
                If,Entity.CreateTime<4600,1682$,Yes:
                Else,1683$,Yes;
1682$    ASSIGN:    Decide 1692.NumberOut True=Decide 1692.NumberOut True + 1:NEXT(2$);

```

1683\$ ASSIGN: Decide 1692.NumberOut False=Decide 1692.NumberOut False + 1:NEXT(21\$);

;
;
;
; Model statements for module: Create 341
;

1684\$ CREATE, 1,SecondstoBaseTime(4600),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.71,0.1,0.88,0.2,1.20,0.3,1.70,0.4,2.60,0.5,
3.93,0.6,6.06,0.7,9.18,0.8,13.80,0.9,21.98,0.95,30.94,0.98,41.30,0.99,48.84,1,61.97)),
107:NEXT(1685\$);

1685\$ ASSIGN: L.Vehicle Entry800_815.NumberOut=L.Vehicle Entry800_815.NumberOut +
1:NEXT(26\$);

;
;
;
; Model statements for module: Decide 1693
;

26\$ BRANCH, 1:
 If,Entity.CreateTime<5500,1688\$,Yes:
 Else,1689\$,Yes;

1688\$ ASSIGN: Decide 1693.NumberOut True=Decide 1693.NumberOut True + 1:NEXT(2\$);

1689\$ ASSIGN: Decide 1693.NumberOut False=Decide 1693.NumberOut False + 1:NEXT(21\$);

;
;
;
; Model statements for module: Create 342
;

1690\$ CREATE, 1,SecondstoBaseTime(5500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.60,0.05,0.73,0.1,0.91,0.2,1.34,0.3,1.94,0.4,2.99,0.5,
4.53,0.6,7.03,0.7,10.52,0.8,15.74,0.9,24.88,0.95,34.28,0.98,44.79,0.99,52.40,1,65.75)),
95:NEXT(1691\$);

1691\$ ASSIGN: L.Vehicle Entry815_830.NumberOut=L.Vehicle Entry815_830.NumberOut +
1:NEXT(27\$);

;
;
;
; Model statements for module: Decide 1694
;

27\$ BRANCH, 1:
 If,Entity.CreateTime<6400,1694\$,Yes:
 Else,1695\$,Yes;

```

1694$    ASSIGN:    Decide 1694.NumberOut True=Decide 1694.NumberOut True + 1:NEXT(2$);

1695$    ASSIGN:    Decide 1694.NumberOut False=Decide 1694.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 343
;

1696$    CREATE,    1,SecondstoBaseTime(6400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.60,0.05,0.73,0.1,0.90,0.2,1.30,0.3,1.87,0.4,2.88,0.5,
4.37,0.6,6.76,0.7,10.15,0.8,15.21,0.9,24.09,0.95,33.37,0.98,43.83,0.99,51.43,1,64.72)),
          98:NEXT(1697$);

1697$    ASSIGN:    L.Vehicle Entry830_845.NumberOut=L.Vehicle Entry830_845.NumberOut +
1:NEXT(28$);

;
;
;   Model statements for module: Decide 1695
;
28$      BRANCH,    1:
                If,Entity.CreateTime<7300,1700$,Yes:
                Else,1701$,Yes;

1700$    ASSIGN:    Decide 1695.NumberOut True=Decide 1695.NumberOut True + 1:NEXT(2$);

1701$    ASSIGN:    Decide 1695.NumberOut False=Decide 1695.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 344
;

1702$    CREATE,    1,SecondstoBaseTime(7300),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.61,0.05,0.75,0.1,0.95,0.2,1.50,0.3,2.21,0.4,3.44,0.5,
5.24,0.6,8.16,0.7,12.09,0.8,18.02,0.9,28.28,0.95,38.21,0.98,48.88,0.99,56.58,1,70.21)),
          83:NEXT(1703$);

1703$    ASSIGN:    L.Vehicle Entry845_900.NumberOut=L.Vehicle Entry845_900.NumberOut +
1:NEXT(29$);

;
;
;   Model statements for module: Decide 1696
;
29$      BRANCH,    1:
                If,Entity.CreateTime<8200,1706$,Yes:

```

```

        Else,1707$,Yes;
1706$    ASSIGN:    Decide 1696.NumberOut True=Decide 1696.NumberOut True + 1:NEXT(2$);

1707$    ASSIGN:    Decide 1696.NumberOut False=Decide 1696.NumberOut False + 1:NEXT(21$);

;
;
;    Model statements for module: Create 345
;

1708$    CREATE,    1,SecondstoBaseTime(8200),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.61,0.05,0.76,0.1,0.98,0.2,1.58,0.3,2.36,0.4,3.69,0.5,
5.63,0.6,8.79,0.7,12.96,0.8,19.28,0.9,30.17,0.95,40.38,0.98,51.14,0.99,58.89,1,72.67)),
        78:NEXT(1709$);

1709$    ASSIGN:    L.Vehicle Entry900_915.NumberOut=L.Vehicle Entry900_915.NumberOut +
1:NEXT(30$);

;
;
;    Model statements for module: Decide 1697
;

30$      BRANCH,    1:
                If,Entity.CreateTime<9100,1712$,Yes:
                Else,1713$,Yes;
1712$    ASSIGN:    Decide 1697.NumberOut True=Decide 1697.NumberOut True + 1:NEXT(2$);

1713$    ASSIGN:    Decide 1697.NumberOut False=Decide 1697.NumberOut False + 1:NEXT(21$);

;
;
;    Model statements for module: Create 346
;

1714$    CREATE,    1,SecondstoBaseTime(9100),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.59,0.05,0.72,0.1,0.89,0.2,1.27,0.3,1.81,0.4,2.78,0.5,
4.21,0.6,6.52,0.7,9.81,0.8,14.71,0.9,23.34,0.95,32.51,0.98,42.94,0.99,50.51,1,63.74)),
        101:NEXT(1715$);

1715$    ASSIGN:    L.Vehicle Entry915_930.NumberOut=L.Vehicle Entry915_930.NumberOut +
1:NEXT(31$);

;
;
;    Model statements for module: Decide 1698
;

31$      BRANCH,    1:

```

```

                If,Entity.CreateTime<10000,1718$,Yes:
                Else,1719$,Yes;
1718$    ASSIGN:    Decide 1698.NumberOut True=Decide 1698.NumberOut True + 1:NEXT(2$);

1719$    ASSIGN:    Decide 1698.NumberOut False=Decide 1698.NumberOut False + 1:NEXT(21$);

;
;
;    Model statements for module: Create 347
;

1720$    CREATE,    1,SecondstoBaseTime(10000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.60,0.05,0.73,0.1,0.90,0.2,1.31,0.3,1.89,0.4,2.92,0.5,
4.42,0.6,6.85,0.7,10.27,0.8,15.37,0.9,24.35,0.95,33.66,0.98,44.14,0.99,51.74,1,65.06)),
97:NEXT(1721$);

1721$    ASSIGN:    L.Vehicle Entry930_945.NumberOut=L.Vehicle Entry930_945.NumberOut +
1:NEXT(32$);

;
;
;    Model statements for module: Decide 1699
;

32$    BRANCH,    1:
                If,Entity.CreateTime<10900,1724$,Yes:
                Else,1725$,Yes;
1724$    ASSIGN:    Decide 1699.NumberOut True=Decide 1699.NumberOut True + 1:NEXT(2$);

1725$    ASSIGN:    Decide 1699.NumberOut False=Decide 1699.NumberOut False + 1:NEXT(21$);

;
;
;    Model statements for module: Create 348
;

1726$    CREATE,    1,SecondstoBaseTime(10900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.52,0.02,0.63,0.05,0.79,0.1,1.03,0.2,1.77,0.3,2.69,0.4,4.24,0.5,
6.48,0.6,10.15,0.7,14.84,0.8,22.01,0.9,34.25,0.95,45.09,0.98,56.04,0.99,63.90,1,78.00)),
69:NEXT(1727$);

1727$    ASSIGN:    L.Vehicle Entry945_1000.NumberOut=L.Vehicle Entry945_1000.NumberOut +
1:NEXT(33$);

;
;
;    Model statements for module: Decide 1700
;

```

```

33$      BRANCH,      1:
          If,Entity.CreateTime<11800,1730$,Yes:
          Else,1731$,Yes;
1730$     ASSIGN:      Decide 1700.NumberOut True=Decide 1700.NumberOut True + 1:NEXT(2$);
1731$     ASSIGN:      Decide 1700.NumberOut False=Decide 1700.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 349
;

1732$     CREATE,      1,SecondstoBaseTime(11800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.59,0.05,0.72,0.1,0.89,0.2,1.25,0.3,1.79,0.4,2.75,0.5,
4.16,0.6,6.44,0.7,9.70,0.8,14.55,0.9,23.10,0.95,32.23,0.98,42.65,0.99,50.22,1,63.43)),
          102:NEXT(1733$);

1733$     ASSIGN:      L.Vehicle Entry1000_1015.NumberOut=L.Vehicle Entry1000_1015.NumberOut
+ 1:NEXT(34$);

;
;
;   Model statements for module: Decide 1701
;

34$      BRANCH,      1:
          If,Entity.CreateTime<12700,1736$,Yes:
          Else,1737$,Yes;
1736$     ASSIGN:      Decide 1701.NumberOut True=Decide 1701.NumberOut True + 1:NEXT(2$);
1737$     ASSIGN:      Decide 1701.NumberOut False=Decide 1701.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 350
;

1738$     CREATE,      1,SecondstoBaseTime(12700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.60,0.05,0.73,0.1,0.90,0.2,1.31,0.3,1.89,0.4,2.92,0.5,
4.42,0.6,6.85,0.7,10.27,0.8,15.38,0.9,24.35,0.95,33.66,0.98,44.14,0.99,51.74,1,65.06)),
          97:NEXT(1739$);

1739$     ASSIGN:      L.Vehicle Entry1015_1030.NumberOut=L.Vehicle Entry1015_1030.NumberOut
+ 1:NEXT(35$);

;
;
;   Model statements for module: Decide 1702

```



```

;
35$      BRANCH,      1:
          If,Entity.CreateTime<13600,1742$,Yes:
          Else,1743$,Yes;
1742$    ASSIGN:      Decide 1702.NumberOut True=Decide 1702.NumberOut True + 1:NEXT(2$);

1743$    ASSIGN:      Decide 1702.NumberOut False=Decide 1702.NumberOut False + 1:NEXT(21$);

;
;
; Model statements for module: Create 351
;

1744$    CREATE,      1,SecondstoBaseTime(13600),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.70,0.1,0.84,0.2,1.07,0.3,1.48,0.4,2.23,0.5,
3.35,0.6,5.13,0.7,7.88,0.8,11.93,0.9,19.18,0.95,27.71,0.98,37.94,0.99,45.41,1,58.31)),
          122:NEXT(1745$);

1745$    ASSIGN:      L.Vehicle Entry1030_1045.NumberOut=L.Vehicle Entry1030_1045.NumberOut
+ 1:NEXT(36$);

;
;
; Model statements for module: Decide 1703
;

36$      BRANCH,      1:
          If,Entity.CreateTime<14500,1748$,Yes:
          Else,1749$,Yes;
1748$    ASSIGN:      Decide 1703.NumberOut True=Decide 1703.NumberOut True + 1:NEXT(2$);

1749$    ASSIGN:      Decide 1703.NumberOut False=Decide 1703.NumberOut False + 1:NEXT(21$);

;
;
; Model statements for module: Create 352
;

1750$    CREATE,      1,SecondstoBaseTime(14500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.71,0.1,0.86,0.2,1.15,0.3,1.62,0.4,2.46,0.5,
3.72,0.6,5.72,0.7,8.71,0.8,13.12,0.9,20.96,0.95,29.76,0.98,40.08,0.99,47.59,1,60.63)),
          112:NEXT(1751$);

1751$    ASSIGN:      L.Vehicle Entry1045_1100.NumberOut=L.Vehicle Entry1045_1100.NumberOut
+ 1:NEXT(37$);

;
;

```

```

;   Model statements for module: Decide 1704
;
37$   BRANCH,      1:
        If,Entity.CreateTime<15400,1754$,Yes:
        Else,1755$,Yes;
1754$   ASSIGN:      Decide 1704.NumberOut True=Decide 1704.NumberOut True + 1:NEXT(2$);

1755$   ASSIGN:      Decide 1704.NumberOut False=Decide 1704.NumberOut False + 1:NEXT(21$);


;
;
;   Model statements for module: Create 353
;

1756$   CREATE,      1,SecondstoBaseTime(15400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.69,0.1,0.82,0.2,1.00,0.3,1.36,0.4,2.03,0.5,
3.03,0.6,4.63,0.7,7.19,0.8,10.92,0.9,17.68,0.95,25.97,0.98,36.13,0.99,43.56,1,56.34)),
        133:NEXT(1757$);

1757$   ASSIGN:      L.Vehicle Entry1100_1115.NumberOut=L.Vehicle Entry1100_1115.NumberOut
+ 1:NEXT(38$);


;
;
;   Model statements for module: Decide 1705
;
38$   BRANCH,      1:
        If,Entity.CreateTime<16300,1760$,Yes:
        Else,1761$,Yes;
1760$   ASSIGN:      Decide 1705.NumberOut True=Decide 1705.NumberOut True + 1:NEXT(2$);

1761$   ASSIGN:      Decide 1705.NumberOut False=Decide 1705.NumberOut False + 1:NEXT(21$);


;
;
;   Model statements for module: Create 354
;

1762$   CREATE,      1,SecondstoBaseTime(16300),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.60,0.05,0.73,0.1,0.91,0.2,1.34,0.3,1.94,0.4,2.99,0.5,
4.53,0.6,7.03,0.7,10.52,0.8,15.74,0.9,24.88,0.95,34.28,0.98,44.79,0.99,52.40,1,65.75)),
        95:NEXT(1763$);

1763$   ASSIGN:      L.Vehicle Entry1115_1130.NumberOut=L.Vehicle Entry1115_1130.NumberOut
+ 1:NEXT(39$);


;

```

```

;
;   Model statements for module: Decide 1706
;
39$      BRANCH,      1:
          If,Entity.CreateTime<17200,1766$,Yes:
          Else,1767$,Yes;
1766$     ASSIGN:      Decide 1706.NumberOut True=Decide 1706.NumberOut True + 1:NEXT(2$);

1767$     ASSIGN:      Decide 1706.NumberOut False=Decide 1706.NumberOut False + 1:NEXT(21$);


;
;
;   Model statements for module: Create 355
;

1768$     CREATE,      1,SecondstoBaseTime(17200),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.59,0.05,0.72,0.1,0.90,0.2,1.28,0.3,1.83,0.4,2.81,0.5,
4.26,0.6,6.60,0.7,9.92,0.8,14.87,0.9,23.59,0.95,32.79,0.98,43.23,0.99,50.81,1,64.06)),
          100:NEXT(1769$);

1769$     ASSIGN:      L.Vehicle Entry1130_1145.NumberOut=L.Vehicle Entry1130_1145.NumberOut
+ 1:NEXT(40$);


;
;
;   Model statements for module: Decide 1707
;
40$      BRANCH,      1:
          If,Entity.CreateTime<18100,1772$,Yes:
          Else,1773$,Yes;
1772$     ASSIGN:      Decide 1707.NumberOut True=Decide 1707.NumberOut True + 1:NEXT(2$);

1773$     ASSIGN:      Decide 1707.NumberOut False=Decide 1707.NumberOut False + 1:NEXT(21$);


;
;
;   Model statements for module: Create 356
;

1774$     CREATE,      1,SecondstoBaseTime(18100),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.70,0.1,0.86,0.2,1.14,0.3,1.59,0.4,2.41,0.5,
3.64,0.6,5.60,0.7,8.53,0.8,12.86,0.9,20.58,0.95,29.32,0.98,39.62,0.99,47.13,1,60.14)),
          114:NEXT(1775$);

1775$     ASSIGN:      L.Vehicle Entry1145_1200.NumberOut=L.Vehicle Entry1145_1200.NumberOut
+ 1:NEXT(41$);

```

```

;
;
;   Model statements for module: Decide 1708
;
41$   BRANCH,      1:
        If,Entity.CreateTime<19000,1778$,Yes:
        Else,1779$,Yes;
1778$   ASSIGN:      Decide 1708.NumberOut True=Decide 1708.NumberOut True + 1:NEXT(2$);

1779$   ASSIGN:      Decide 1708.NumberOut False=Decide 1708.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 357
;

1780$   CREATE,      1,SecondstoBaseTime(19000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.60,0.05,0.74,0.1,0.92,0.2,1.38,0.3,2.00,0.4,3.10,0.5,
4.71,0.6,7.31,0.7,10.91,0.8,16.31,0.9,25.73,0.95,35.26,0.98,45.80,0.99,53.44,1,66.86)),
        91:NEXT(1781$);

1781$   ASSIGN:      L.Vehicle Entry1200_1215.NumberOut=L.Vehicle Entry1200_1215.NumberOut
+ 1:NEXT(42$);

;
;
;   Model statements for module: Decide 1709
;
42$   BRANCH,      1:
        If,Entity.CreateTime<19900,1784$,Yes:
        Else,1785$,Yes;
1784$   ASSIGN:      Decide 1709.NumberOut True=Decide 1709.NumberOut True + 1:NEXT(2$);

1785$   ASSIGN:      Decide 1709.NumberOut False=Decide 1709.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 358
;

1786$   CREATE,      1,SecondstoBaseTime(19900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.70,0.1,0.84,0.2,1.08,0.3,1.49,0.4,2.25,0.5,
3.38,0.6,5.18,0.7,7.96,0.8,12.04,0.9,19.35,0.95,27.90,0.98,38.14,0.99,45.61,1,58.52)),
        121:NEXT(1787$);

1787$   ASSIGN:      L.Vehicle Entry1215_1230.NumberOut=L.Vehicle Entry1215_1230.NumberOut
+ 1:NEXT(43$);

```

```

;
;
;   Model statements for module: Decide 1710
;
43$   BRANCH,      1:
        If,Entity.CreateTime<20800,1790$,Yes:
        Else,1791$,Yes;
1790$   ASSIGN:      Decide 1710.NumberOut True=Decide 1710.NumberOut True + 1:NEXT(2$);

1791$   ASSIGN:      Decide 1710.NumberOut False=Decide 1710.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 359
;

1792$   CREATE,      1,SecondstoBaseTime(20800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.57,0.05,0.68,0.1,0.81,0.2,0.97,0.3,1.30,0.4,1.94,0.5,
2.89,0.6,4.40,0.7,6.88,0.8,10.47,0.9,17.01,0.95,25.20,0.98,35.33,0.99,42.74,1,55.47)),
        138:NEXT(1793$);

1793$   ASSIGN:      L.Vehicle Entry1230_1245.NumberOut=L.Vehicle Entry1230_1245.NumberOut
+ 1:NEXT(44$);

;
;
;   Model statements for module: Decide 1711
;
44$   BRANCH,      1:
        If,Entity.CreateTime<21700,1796$,Yes:
        Else,1797$,Yes;
1796$   ASSIGN:      Decide 1711.NumberOut True=Decide 1711.NumberOut True + 1:NEXT(2$);

1797$   ASSIGN:      Decide 1711.NumberOut False=Decide 1711.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 360
;

1798$   CREATE,      1,SecondstoBaseTime(21700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.69,0.1,0.83,0.2,1.02,0.3,1.39,0.4,2.08,0.5,
3.12,0.6,4.77,0.7,7.39,0.8,11.20,0.9,18.10,0.95,26.46,0.98,36.65,0.99,44.09,1,56.90)),
        130:NEXT(1799$);

1799$   ASSIGN:      L.Vehicle Entry1245_1300.NumberOut=L.Vehicle Entry1245_1300.NumberOut
+ 1:NEXT(45$);

```

```

;
;
;   Model statements for module: Decide 1712
;
45$   BRANCH,      1:
        If,Entity.CreateTime<22600,1802$,Yes:
        Else,1803$,Yes;
1802$   ASSIGN:      Decide 1712.NumberOut True=Decide 1712.NumberOut True + 1:NEXT(2$);

1803$   ASSIGN:      Decide 1712.NumberOut False=Decide 1712.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 361
;

1804$   CREATE,      1,SecondstoBaseTime(22600),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.72,0.1,0.88,0.2,1.22,0.3,1.74,0.4,2.66,0.5,
4.02,0.6,6.21,0.7,9.38,0.8,14.09,0.9,22.42,0.95,31.44,0.98,41.83,0.99,49.38,1,62.54)),
        105:NEXT(1805$);

1805$   ASSIGN:      L.Vehicle Entry1300_1315.NumberOut=L.Vehicle Entry1300_1315.NumberOut
+ 1:NEXT(46$);

;
;
;   Model statements for module: Decide 1713
;
46$   BRANCH,      1:
        If,Entity.CreateTime<23500,1808$,Yes:
        Else,1809$,Yes;
1808$   ASSIGN:      Decide 1713.NumberOut True=Decide 1713.NumberOut True + 1:NEXT(2$);

1809$   ASSIGN:      Decide 1713.NumberOut False=Decide 1713.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 362
;

1810$   CREATE,      1,SecondstoBaseTime(23500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.59,0.05,0.72,0.1,0.89,0.2,1.25,0.3,1.79,0.4,2.75,0.5,
4.16,0.6,6.44,0.7,9.70,0.8,14.55,0.9,23.10,0.95,32.23,0.98,42.65,0.99,50.22,1,63.43)),
        102:NEXT(1811$);

```

```
1811$    ASSIGN:    L.Vehicle Entry1315_1330.NumberOut=L.Vehicle Entry1315_1330.NumberOut
+ 1:NEXT(47$);
```

```
;
;
; Model statements for module: Decide 1714
;
```

```
47$    BRANCH,    1:
                If,Entity.CreateTime<24400,1814$,Yes:
                Else,1815$,Yes;
```

```
1814$    ASSIGN:    Decide 1714.NumberOut True=Decide 1714.NumberOut True + 1:NEXT(2$);
```

```
1815$    ASSIGN:    Decide 1714.NumberOut False=Decide 1714.NumberOut False + 1:NEXT(21$);
```

```
;
;
; Model statements for module: Create 363
;
```

```
1816$    CREATE,    1,SecondstoBaseTime(24400),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.59,0.05,0.72,0.1,0.89,0.2,1.27,0.3,1.81,0.4,2.78,0.5,
4.21,0.6,6.52,0.7,9.81,0.8,14.71,0.9,23.34,0.95,32.51,0.98,42.94,0.99,50.51,1,63.74)),
101:NEXT(1817$);
```

```
1817$    ASSIGN:    L.Vehicle Entry1330_1345.NumberOut=L.Vehicle Entry1330_1345.NumberOut
+ 1:NEXT(48$);
```

```
;
;
; Model statements for module: Decide 1715
;
```

```
48$    BRANCH,    1:
                If,Entity.CreateTime<25300,1820$,Yes:
                Else,1821$,Yes;
```

```
1820$    ASSIGN:    Decide 1715.NumberOut True=Decide 1715.NumberOut True + 1:NEXT(2$);
```

```
1821$    ASSIGN:    Decide 1715.NumberOut False=Decide 1715.NumberOut False + 1:NEXT(21$);
```

```
;
;
; Model statements for module: Create 364
;
```

```
1822$    CREATE,    1,SecondstoBaseTime(25300),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.69,0.1,0.83,0.2,1.01,0.3,1.38,0.4,2.06,0.5,
3.09,0.6,4.72,0.7,7.32,0.8,11.11,0.9,17.96,0.95,26.30,0.98,36.47,0.99,43.91,1,56.71)),
131:NEXT(1823$);
```

```
1823$    ASSIGN:    L.Vehicle Entry1345_1400.NumberOut=L.Vehicle Entry1345_1400.NumberOut
+ 1:NEXT(49$);
```

```
;
;
; Model statements for module: Decide 1716
;
```

```
49$      BRANCH,    1:
                If,Entity.CreateTime<26200,1826$,Yes:
                Else,1827$,Yes;
```

```
1826$    ASSIGN:    Decide 1716.NumberOut True=Decide 1716.NumberOut True + 1:NEXT(2$);
```

```
1827$    ASSIGN:    Decide 1716.NumberOut False=Decide 1716.NumberOut False + 1:NEXT(21$);
```

```
;
;
; Model statements for module: Create 365
;
```

```
1828$    CREATE,    1,SecondstoBaseTime(26200),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.71,0.1,0.86,0.2,1.14,0.3,1.61,0.4,2.44,0.5,
3.68,0.6,5.66,0.7,8.62,0.8,12.99,0.9,20.77,0.95,29.54,0.98,39.85,0.99,47.36,1,60.38)),
113:NEXT(1829$);
```

```
1829$    ASSIGN:    L.Vehicle Entry1400_1415.NumberOut=L.Vehicle Entry1400_1415.NumberOut
+ 1:NEXT(50$);
```

```
;
;
; Model statements for module: Decide 1717
;
```

```
50$      BRANCH,    1:
                If,Entity.CreateTime<27100,1832$,Yes:
                Else,1833$,Yes;
```

```
1832$    ASSIGN:    Decide 1717.NumberOut True=Decide 1717.NumberOut True + 1:NEXT(2$);
```

```
1833$    ASSIGN:    Decide 1717.NumberOut False=Decide 1717.NumberOut False + 1:NEXT(21$);
```

```
;
;
; Model statements for module: Create 366
;
```

```
1834$    CREATE,    1,SecondstoBaseTime(27100),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.49,0.02,0.57,0.05,0.66,0.1,0.78,0.2,0.83,0.3,1.07,0.4,1.54,0.5,
2.27,0.6,3.41,0.7,5.50,0.8,8.48,0.9,14.03,0.95,21.76,0.98,31.75,0.99,39.09,1,51.57)),
```



```

167:NEXT(1835$);

1835$    ASSIGN:    L.Vehicle Entry1415_1430.NumberOut=L.Vehicle Entry1415_1430.NumberOut
+ 1:NEXT(51$);

;
;
;   Model statements for module: Decide 1718
;
51$      BRANCH,    1:
                If,Entity.CreateTime<28000,1838$,Yes:
                Else,1839$,Yes;
1838$    ASSIGN:    Decide 1718.NumberOut True=Decide 1718.NumberOut True + 1:NEXT(2$);
1839$    ASSIGN:    Decide 1718.NumberOut False=Decide 1718.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 367
;
1840$    CREATE,    1,SecondstoBaseTime(28000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.68,0.1,0.82,0.2,0.98,0.3,1.33,0.4,1.97,0.5,
2.95,0.6,4.49,0.7,7.00,0.8,10.65,0.9,17.27,0.95,25.50,0.98,35.64,0.99,43.06,1,55.81)),
                136:NEXT(1841$);

1841$    ASSIGN:    L.Vehicle Entry1430_1445.NumberOut=L.Vehicle Entry1430_1445.NumberOut
+ 1:NEXT(52$);

;
;
;   Model statements for module: Decide 1719
;
52$      BRANCH,    1:
                If,Entity.CreateTime<28900,1844$,Yes:
                Else,1845$,Yes;
1844$    ASSIGN:    Decide 1719.NumberOut True=Decide 1719.NumberOut True + 1:NEXT(2$);
1845$    ASSIGN:    Decide 1719.NumberOut False=Decide 1719.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 368
;
1846$    CREATE,    1,SecondstoBaseTime(28900),Vehicles:

```

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.57,0.05,0.68,0.1,0.81,0.2,0.95,0.3,1.27,0.4,1.89,0.5,2.82,0.6,4.28,0.7,6.71,0.8,10.22,0.9,16.63,0.95,24.76,0.98,34.88,0.99,42.28,1,54.97)),
141:NEXT(1847\$);

1847\$ ASSIGN: L.Vehicle Entry1445_1500.NumberOut=L.Vehicle Entry1445_1500.NumberOut
+ 1:NEXT(53\$);

;
;
; Model statements for module: Decide 1720
;

53\$ BRANCH, 1:
 If,Entity.CreateTime<29800,1850\$,Yes:
 Else,1851\$,Yes;

1850\$ ASSIGN: Decide 1720.NumberOut True=Decide 1720.NumberOut True + 1:NEXT(2\$);

1851\$ ASSIGN: Decide 1720.NumberOut False=Decide 1720.NumberOut False + 1:NEXT(21\$);

;
;
; Model statements for module: Create 369
;

1852\$ CREATE, 1,SecondstoBaseTime(29800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.69,0.1,0.82,0.2,0.99,0.3,1.35,0.4,2.01,0.5,3.00,0.6,4.58,0.7,7.13,0.8,10.83,0.9,17.54,0.95,25.81,0.98,35.97,0.99,43.39,1,56.16)),
134:NEXT(1853\$);

1853\$ ASSIGN: L.Vehicle Entry1500_1515.NumberOut=L.Vehicle Entry1500_1515.NumberOut
+ 1:NEXT(54\$);

;
;
; Model statements for module: Decide 1721
;

54\$ BRANCH, 1:
 If,Entity.CreateTime<30700,1856\$,Yes:
 Else,1857\$,Yes;

1856\$ ASSIGN: Decide 1721.NumberOut True=Decide 1721.NumberOut True + 1:NEXT(2\$);

1857\$ ASSIGN: Decide 1721.NumberOut False=Decide 1721.NumberOut False + 1:NEXT(21\$);

;
;
; Model statements for module: Create 370
;

1858\$ CREATE, 1,SecondstoBaseTime(30700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.69,0.1,0.82,0.2,1.01,0.3,1.37,0.4,2.04,0.5,
3.06,0.6,4.67,0.7,7.25,0.8,11.01,0.9,17.82,0.95,26.13,0.98,36.30,0.99,43.73,1,56.52)),
132:NEXT(1859\$);

1859\$ ASSIGN: L.Vehicle Entry1515_1530.NumberOut=L.Vehicle Entry1515_1530.NumberOut
+ 1:NEXT(55\$);

;
;
;
; Model statements for module: Decide 1722
;

55\$ BRANCH, 1:
 If,Entity.CreateTime<31600,1862\$,Yes:
 Else,1863\$,Yes;

1862\$ ASSIGN: Decide 1722.NumberOut True=Decide 1722.NumberOut True + 1:NEXT(2\$);

1863\$ ASSIGN: Decide 1722.NumberOut False=Decide 1722.NumberOut False + 1:NEXT(21\$);

;
;
;
; Model statements for module: Create 371
;

1864\$ CREATE, 1,SecondstoBaseTime(31600),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.69,0.1,0.83,0.2,1.03,0.3,1.40,0.4,2.10,0.5,
3.15,0.6,4.82,0.7,7.45,0.8,11.30,0.9,18.25,0.95,26.63,0.98,36.82,0.99,44.27,1,57.09)),
129:NEXT(1865\$);

1865\$ ASSIGN: L.Vehicle Entry1530_1545.NumberOut=L.Vehicle Entry1530_1545.NumberOut
+ 1:NEXT(56\$);

;
;
;
; Model statements for module: Decide 1723
;

56\$ BRANCH, 1:
 If,Entity.CreateTime<32500,1868\$,Yes:
 Else,1869\$,Yes;

1868\$ ASSIGN: Decide 1723.NumberOut True=Decide 1723.NumberOut True + 1:NEXT(2\$);

1869\$ ASSIGN: Decide 1723.NumberOut False=Decide 1723.NumberOut False + 1:NEXT(21\$);

;
;
;
; Model statements for module: Create 372
;

```

1870$    CREATE,      1,SecondstoBaseTime(32500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.68,0.1,0.82,0.2,0.99,0.3,1.34,0.4,1.99,0.5,
2.98,0.6,4.54,0.7,7.06,0.8,10.74,0.9,17.40,0.95,25.65,0.98,35.80,0.99,43.23,1,55.98)),
          135:NEXT(1871$);

1871$    ASSIGN:      L.Vehicle Entry1545_1600.NumberOut=L.Vehicle Entry1545_1600.NumberOut
+ 1:NEXT(57$);

;
;
; Model statements for module: Decide 1724
;
57$      BRANCH,      1:
          If,Entity.CreateTime<33400,1874$,Yes:
          Else,1875$,Yes;

1874$    ASSIGN:      Decide 1724.NumberOut True=Decide 1724.NumberOut True + 1:NEXT(2$);

1875$    ASSIGN:      Decide 1724.NumberOut False=Decide 1724.NumberOut False + 1:NEXT(21$);

;
;
; Model statements for module: Create 373
;

1876$    CREATE,      1,SecondstoBaseTime(33400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.71,0.1,0.86,0.2,1.14,0.3,1.61,0.4,2.44,0.5,
3.68,0.6,5.66,0.7,8.62,0.8,12.99,0.9,20.77,0.95,29.54,0.98,39.85,0.99,47.36,1,60.38)),
          113:NEXT(1877$);

1877$    ASSIGN:      L.Vehicle Entry1600_1615.NumberOut=L.Vehicle Entry1600_1615.NumberOut
+ 1:NEXT(58$);

;
;
; Model statements for module: Decide 1725
;
58$      BRANCH,      1:
          If,Entity.CreateTime<34300,1880$,Yes:
          Else,1881$,Yes;

1880$    ASSIGN:      Decide 1725.NumberOut True=Decide 1725.NumberOut True + 1:NEXT(2$);

1881$    ASSIGN:      Decide 1725.NumberOut False=Decide 1725.NumberOut False + 1:NEXT(21$);

;
;
; Model statements for module: Create 374

```

```

;
1882$    CREATE,      1,SecondstoBaseTime(34300),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.69,0.1,0.82,0.2,1.00,0.3,1.36,0.4,2.03,0.5,
3.03,0.6,4.63,0.7,7.19,0.8,10.92,0.9,17.68,0.95,25.97,0.98,36.13,0.99,43.56,1,56.34)),
          133:NEXT(1883$);

1883$    ASSIGN:      L.Vehicle Entry1615_1630.NumberOut=L.Vehicle Entry1615_1630.NumberOut
+ 1:NEXT(59$);

;
;
; Model statements for module: Decide 1726
;
59$      BRANCH,      1:
          If,Entity.CreateTime<35200,1886$,Yes:
          Else,1887$,Yes;
1886$    ASSIGN:      Decide 1726.NumberOut True=Decide 1726.NumberOut True + 1:NEXT(2$);
1887$    ASSIGN:      Decide 1726.NumberOut False=Decide 1726.NumberOut False + 1:NEXT(21$);

;
;
; Model statements for module: Create 375
;

1888$    CREATE,      1,SecondstoBaseTime(35200),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.49,0.02,0.57,0.05,0.67,0.1,0.79,0.2,0.88,0.3,1.15,0.4,1.68,0.5,
2.49,0.6,3.76,0.7,5.98,0.8,9.17,0.9,15.07,0.95,22.96,0.98,33.00,0.99,40.36,1,52.93)),
          155:NEXT(1889$);

1889$    ASSIGN:      L.Vehicle Entry1630_1645.NumberOut=L.Vehicle Entry1630_1645.NumberOut
+ 1:NEXT(60$);

;
;
; Model statements for module: Decide 1727
;
60$      BRANCH,      1:
          If,Entity.CreateTime<36100,1892$,Yes:
          Else,1893$,Yes;
1892$    ASSIGN:      Decide 1727.NumberOut True=Decide 1727.NumberOut True + 1:NEXT(2$);
1893$    ASSIGN:      Decide 1727.NumberOut False=Decide 1727.NumberOut False + 1:NEXT(21$);

;
;

```

```
; Model statements for module: Create 376
;
```

```
1894$ CREATE, 1,SecondstoBaseTime(36100),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.70,0.1,0.85,0.2,1.09,0.3,1.51,0.4,2.27,0.5,
3.42,0.6,5.24,0.7,8.04,0.8,12.15,0.9,19.51,0.95,28.09,0.98,38.34,0.99,45.82,1,58.74)),
120:NEXT(1895$);
```

```
1895$ ASSIGN: L.Vehicle Entry1645_1700.NumberOut=L.Vehicle Entry1645_1700.NumberOut
+ 1:NEXT(61$);
```

```
;
;
; Model statements for module: Decide 1728
;
```

```
61$ BRANCH, 1:
If,Entity.CreateTime<37000,1898$,Yes:
Else,1899$,Yes;
```

```
1898$ ASSIGN: Decide 1728.NumberOut True=Decide 1728.NumberOut True + 1:NEXT(2$);
```

```
1899$ ASSIGN: Decide 1728.NumberOut False=Decide 1728.NumberOut False + 1:NEXT(21$);
```

```
;
;
; Model statements for module: Create 377
;
```

```
1900$ CREATE, 1,SecondstoBaseTime(37000),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.71,0.1,0.87,0.2,1.17,0.3,1.65,0.4,2.52,0.5,
3.80,0.6,5.85,0.7,8.89,0.8,13.38,0.9,21.36,0.95,30.22,0.98,40.56,0.99,48.08,1,61.15)),
110:NEXT(1901$);
```

```
1901$ ASSIGN: L.Vehicle Entry1700_1715.NumberOut=L.Vehicle Entry1700_1715.NumberOut
+ 1:NEXT(62$);
```

```
;
;
; Model statements for module: Decide 1729
;
```

```
62$ BRANCH, 1:
If,Entity.CreateTime<37900,1904$,Yes:
Else,1905$,Yes;
```

```
1904$ ASSIGN: Decide 1729.NumberOut True=Decide 1729.NumberOut True + 1:NEXT(2$);
```

```
1905$ ASSIGN: Decide 1729.NumberOut False=Decide 1729.NumberOut False + 1:NEXT(21$);
```

```
;
```

```
;
; Model statements for module: Create 378
;
```

```
1906$ CREATE, 1,SecondstoBaseTime(37900),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.70,0.1,0.85,0.2,1.09,0.3,1.51,0.4,2.27,0.5,
3.42,0.6,5.24,0.7,8.04,0.8,12.15,0.9,19.51,0.95,28.09,0.98,38.34,0.99,45.82,1,58.74)),
120:NEXT(1907$);
```

```
1907$ ASSIGN: L.Vehicle Entry1715_1730.NumberOut=L.Vehicle Entry1715_1730.NumberOut
+ 1:NEXT(63$);
```

```
;
;
; Model statements for module: Decide 1730
;
```

```
63$ BRANCH, 1:
If,Entity.CreateTime<38800,1910$,Yes:
Else,1911$,Yes;
```

```
1910$ ASSIGN: Decide 1730.NumberOut True=Decide 1730.NumberOut True + 1:NEXT(2$);
```

```
1911$ ASSIGN: Decide 1730.NumberOut False=Decide 1730.NumberOut False + 1:NEXT(21$);
```

```
;
;
; Model statements for module: Create 379
;
```

```
1912$ CREATE, 1,SecondstoBaseTime(38800),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.71,0.1,0.87,0.2,1.16,0.3,1.64,0.4,2.49,0.5,
3.76,0.6,5.79,0.7,8.80,0.8,13.25,0.9,21.16,0.95,29.99,0.98,40.32,0.99,47.84,1,60.89)),
111:NEXT(1913$);
```

```
1913$ ASSIGN: L.Vehicle Entry1730_1745.NumberOut=L.Vehicle Entry1730_1745.NumberOut
+ 1:NEXT(64$);
```

```
;
;
; Model statements for module: Decide 1731
;
```

```
64$ BRANCH, 1:
If,Entity.CreateTime<39700,1916$,Yes:
Else,1917$,Yes;
```

```
1916$ ASSIGN: Decide 1731.NumberOut True=Decide 1731.NumberOut True + 1:NEXT(2$);
```

```
1917$ ASSIGN: Decide 1731.NumberOut False=Decide 1731.NumberOut False + 1:NEXT(21$);
```

```
;
;
; Model statements for module: Create 380
;
```

```
1918$ CREATE, 1,SecondstoBaseTime(39700),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.70,0.1,0.85,0.2,1.12,0.3,1.56,0.4,2.36,0.5,
3.56,0.6,5.47,0.7,8.36,0.8,12.62,0.9,20.21,0.95,28.90,0.98,39.18,0.99,46.67,1,59.65)),
116:NEXT(1919$);
```

```
1919$ ASSIGN: L.Vehicle Entry1745_1800.NumberOut=L.Vehicle Entry1745_1800.NumberOut
+ 1:NEXT(65$);
```

```
;
;
; Model statements for module: Decide 1732
;
```

```
65$ BRANCH, 1:
If,Entity.CreateTime<40600,1922$,Yes:
Else,1923$,Yes;
```

```
1922$ ASSIGN: Decide 1732.NumberOut True=Decide 1732.NumberOut True + 1:NEXT(2$);
```

```
1923$ ASSIGN: Decide 1732.NumberOut False=Decide 1732.NumberOut False + 1:NEXT(21$);
```

```
;
;
; Model statements for module: Create 381
;
```

```
1924$ CREATE, 1,SecondstoBaseTime(40600),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.71,0.1,0.87,0.2,1.19,0.3,1.69,0.4,2.57,0.5,
3.88,0.6,5.99,0.7,9.08,0.8,13.66,0.9,21.77,0.95,30.69,0.98,41.05,0.99,48.59,1,61.69)),
108:NEXT(1925$);
```

```
1925$ ASSIGN: L.Vehicle Entry1800_1815.NumberOut=L.Vehicle Entry1800_1815.NumberOut
+ 1:NEXT(66$);
```

```
;
;
; Model statements for module: Decide 1733
;
```

```
66$ BRANCH, 1:
If,Entity.CreateTime<41500,1928$,Yes:
Else,1929$,Yes;
```

```
1928$ ASSIGN: Decide 1733.NumberOut True=Decide 1733.NumberOut True + 1:NEXT(2$);
```

```
1929$ ASSIGN: Decide 1733.NumberOut False=Decide 1733.NumberOut False + 1:NEXT(21$);
```



```

;
;
;   Model statements for module: Create 382
;

1930$   CREATE,      1,SecondstoBaseTime(41500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.59,0.05,0.71,0.1,0.87,0.2,1.17,0.3,1.65,0.4,2.52,0.5,
3.80,0.6,5.85,0.7,8.89,0.8,13.38,0.9,21.36,0.95,30.22,0.98,40.56,0.99,48.08,1,61.15)),
          110:NEXT(1931$);

1931$   ASSIGN:      L.Vehicle Entry1815_1830.NumberOut=L.Vehicle Entry1815_1830.NumberOut
+ 1:NEXT(67$);

;
;
;   Model statements for module: Decide 1734
;
67$     BRANCH,      1:
          If,Entity.CreateTime<42400,1934$,Yes:
          Else,1935$,Yes;
1934$   ASSIGN:      Decide 1734.NumberOut True=Decide 1734.NumberOut True + 1:NEXT(2$);
1935$   ASSIGN:      Decide 1734.NumberOut False=Decide 1734.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 383
;

1936$   CREATE,      1,SecondstoBaseTime(42400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.02,0.58,0.05,0.70,0.1,0.84,0.2,1.07,0.3,1.48,0.4,2.23,0.5,
3.35,0.6,5.13,0.7,7.88,0.8,11.93,0.9,19.18,0.95,27.71,0.98,37.94,0.99,45.41,1,58.31)),
          122:NEXT(1937$);

1937$   ASSIGN:      L.Vehicle Entry1830_1845.NumberOut=L.Vehicle Entry1830_1845.NumberOut
+ 1:NEXT(68$);

;
;
;   Model statements for module: Decide 1735
;
68$     BRANCH,      1:
          If,Entity.CreateTime<43300,1940$,Yes:
          Else,1941$,Yes;
1940$   ASSIGN:      Decide 1735.NumberOut True=Decide 1735.NumberOut True + 1:NEXT(2$);
1941$   ASSIGN:      Decide 1735.NumberOut False=Decide 1735.NumberOut False + 1:NEXT(21$);

```

```

;
;
; Model statements for module: Create 384
;

1942$    CREATE,      1,SecondstoBaseTime(43300),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.61,0.05,0.76,0.1,0.96,0.2,1.53,0.3,2.27,0.4,3.54,0.5,
5.39,0.6,8.41,0.7,12.43,0.8,18.50,0.9,29.01,0.95,39.05,0.98,49.75,0.99,57.47,1,71.16)),
      81:NEXT(1943$);

1943$    ASSIGN:      L.Vehicle Entry1845_1900.NumberOut=L.Vehicle Entry1845_1900.NumberOut
+ 1:NEXT(69$);

;
;
; Model statements for module: Decide 1736
;
69$      BRANCH,      1:
          If,Entity.CreateTime<44200,1946$,Yes:
          Else,1947$,Yes;
1946$    ASSIGN:      Decide 1736.NumberOut True=Decide 1736.NumberOut True + 1:NEXT(2$);
1947$    ASSIGN:      Decide 1736.NumberOut False=Decide 1736.NumberOut False + 1:NEXT(21$);

;
;
; Model statements for module: Create 385
;

1948$    CREATE,      1,SecondstoBaseTime(44200),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.52,0.02,0.62,0.05,0.77,0.1,0.99,0.2,1.62,0.3,2.43,0.4,3.80,0.5,
5.80,0.6,9.06,0.7,13.34,0.8,19.83,0.9,30.99,0.95,41.33,0.98,52.12,0.99,59.89,1,73.74)),
      76:NEXT(1949$);

1949$    ASSIGN:      L.Vehicle Entry1900_1915.NumberOut=L.Vehicle Entry1900_1915.NumberOut
+ 1:NEXT(70$);

;
;
; Model statements for module: Decide 1737
;
70$      BRANCH,      1:
          If,Entity.CreateTime<45100,1952$,Yes:
          Else,1953$,Yes;
1952$    ASSIGN:      Decide 1737.NumberOut True=Decide 1737.NumberOut True + 1:NEXT(2$);

```

1953\$ ASSIGN: Decide 1737.NumberOut False=Decide 1737.NumberOut False + 1:NEXT(21\$);

;
;
;
; Model statements for module: Create 386
;

1954\$ CREATE, 1,SecondstoBaseTime(45100),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.61,0.05,0.76,0.1,0.97,0.2,1.57,0.3,2.33,0.4,3.64,0.5,
5.55,0.6,8.66,0.7,12.78,0.8,19.01,0.9,29.77,0.95,39.92,0.98,50.66,0.99,58.40,1,72.15)),
79:NEXT(1955\$);

1955\$ ASSIGN: L.Vehicle Entry1915_1930.NumberOut=L.Vehicle Entry1915_1930.NumberOut
+ 1:NEXT(71\$);

;
;
;
; Model statements for module: Decide 1738
;

71\$ BRANCH, 1:
 If,Entity.CreateTime<46000,1958\$,Yes:
 Else,1959\$,Yes;

1958\$ ASSIGN: Decide 1738.NumberOut True=Decide 1738.NumberOut True + 1:NEXT(2\$);

1959\$ ASSIGN: Decide 1738.NumberOut False=Decide 1738.NumberOut False + 1:NEXT(21\$);

;
;
;
; Model statements for module: Create 387
;

1960\$ CREATE, 1,SecondstoBaseTime(46000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.52,0.02,0.63,0.05,0.78,0.1,1.02,0.2,1.75,0.3,2.65,0.4,4.17,0.5,
6.38,0.6,9.98,0.7,14.61,0.8,21.67,0.9,33.74,0.95,44.51,0.98,55.43,0.99,63.27,1,77.34)),
70:NEXT(1961\$);

1961\$ ASSIGN: L.Vehicle Entry1930_1945.NumberOut=L.Vehicle Entry1930_1945.NumberOut
+ 1:NEXT(72\$);

;
;
;
; Model statements for module: Decide 1739
;

72\$ BRANCH, 1:
 If,Entity.CreateTime<46900,1964\$,Yes:
 Else,1965\$,Yes;

1964\$ ASSIGN: Decide 1739.NumberOut True=Decide 1739.NumberOut True + 1:NEXT(2\$);

1965\$ ASSIGN: Decide 1739.NumberOut False=Decide 1739.NumberOut False + 1:NEXT(21\$);

;
;
;
; Model statements for module: Create 388
;

1966\$ CREATE, 1,SecondstoBaseTime(46900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.61,0.05,0.75,0.1,0.95,0.2,1.50,0.3,2.21,0.4,3.44,0.5,
5.24,0.6,8.16,0.7,12.09,0.8,18.02,0.9,28.29,0.95,38.21,0.98,48.88,0.99,56.58,1,70.21)),
83:NEXT(1967\$);

1967\$ ASSIGN: L.Vehicle Entry1945_2000.NumberOut=L.Vehicle Entry1945_2000.NumberOut
+ 1:NEXT(73\$);

;
;
;
; Model statements for module: Decide 1740
;

73\$ BRANCH, 1:
 If,Entity.CreateTime<47800,1970\$,Yes:
 Else,1971\$,Yes;

1970\$ ASSIGN: Decide 1740.NumberOut True=Decide 1740.NumberOut True + 1:NEXT(2\$);

1971\$ ASSIGN: Decide 1740.NumberOut False=Decide 1740.NumberOut False + 1:NEXT(21\$);

;
;
;
; Model statements for module: Create 389
;

1972\$ CREATE, 1,SecondstoBaseTime(47800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.52,0.02,0.63,0.05,0.78,0.1,1.02,0.2,1.75,0.3,2.65,0.4,4.17,0.5,
6.38,0.6,9.98,0.7,14.61,0.8,21.67,0.9,33.74,0.95,44.51,0.98,55.43,0.99,63.27,1,77.34)),
70:NEXT(1973\$);

1973\$ ASSIGN: L.Vehicle Entry2000_2015.NumberOut=L.Vehicle Entry2000_2015.NumberOut
+ 1:NEXT(74\$);

;
;
;
; Model statements for module : Decide 1741
;

74\$ BRANCH, 1:
 If,Entity.CreateTime<48700,1976\$,Yes:
 Else,1977\$,Yes;

```

1976$    ASSIGN:    Decide 1741.NumberOut True=Decide 1741.NumberOut True + 1:NEXT(2$);

1977$    ASSIGN:    Decide 1741.NumberOut False=Decide 1741.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 444
;

1978$    CREATE,    1,SecondstoBaseTime(0),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.51,0.02,0.61,0.05,0.76,0.1,0.97,0.2,1.55,0.3,2.30,0.4,3.59,0.5,
5.47,0.6,8.53,0.7,12.60,0.8,18.76,0.9,29.39,0.95,39.48,0.98,50.20,0.99,57.93,1,71.65)),
10:NEXT(1979$);

1979$    ASSIGN:    Warmup Entry.NumberOut=Warmup Entry.NumberOut + 1:NEXT(75$);

;
;
;   Model statements for module: Decide 1796
;
75$      BRANCH,    1:
                If,Entity.CreateTime<100,1982$,Yes:
                Else,1983$,Yes;
1982$    ASSIGN:    Decide 1796.NumberOut True=Decide 1796.NumberOut True + 1:NEXT(2$);
1983$    ASSIGN:    Decide 1796.NumberOut False=Decide 1796.NumberOut False + 1:NEXT(21$);

;
;
;   Model statements for module: Create 390
;

1984$    CREATE,    1,SecondstoBaseTime(100),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.71,0.02,0.88,0.05,1.16,0.1,1.54,0.2,2.20,0.3,2.88,0.4,3.68,0.5,
4.68,0.6,5.93,0.7,7.46,0.8,9.56,0.9,13.25,0.95,16.91,0.98,20.93,0.99,23.57,1,30.23)),
141:NEXT(1985$);

1985$    ASSIGN:    R.Vehicle Enters645_700.NumberOut=R.Vehicle Enters645_700.NumberOut +
1:NEXT(76$);

;
;
;   Model statements for module: Decide 1742
;
76$      BRANCH,    1:
                If,Entity.CreateTime<1000,1988$,Yes:
                Else,1989$,Yes;

```

```

1988$    ASSIGN:    Decide 1742.NumberOut True=Decide 1742.NumberOut True + 1:NEXT(3$);
1989$    ASSIGN:    Decide 1742.NumberOut False=Decide 1742.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Dispose 10
;
77$      ASSIGN:    Dispose 10.NumberOut=Dispose 10.NumberOut + 1;
1990$    DISPOSE:    No;

;
;
;   Model statements for module: Create 391
;

1991$    CREATE,      1,SecondstoBaseTime(1000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.69,0.02,0.85,0.05,1.12,0.1,1.47,0.2,2.06,0.3,2.68,0.4,3.41,0.5,
4.31,0.6,5.42,0.7,6.77,0.8,8.69,0.9,12.04,0.95,15.18,0.98,19.00,0.99,21.52,1,28.07)),
155:NEXT(1992$);

1992$    ASSIGN:    R.Vehicle Enters700_715.NumberOut=R.Vehicle Enters700_715.NumberOut +
1:NEXT(78$);

;
;
;   Model statements for module: Decide 1743
;
78$      BRANCH,      1:
                If,Entity.CreateTime<1900,1995$,Yes:
                Else,1996$,Yes;
1995$    ASSIGN:    Decide 1743.NumberOut True=Decide 1743.NumberOut True + 1:NEXT(3$);
1996$    ASSIGN:    Decide 1743.NumberOut False=Decide 1743.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 392
;

1997$    CREATE,      1,SecondstoBaseTime(1900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1.08,0.1,1.37,0.2,1.87,0.3,2.40,0.4,3.04,0.5,
3.81,0.6,4.73,0.7,5.85,0.8,7.53,0.9,10.40,0.95,12.84,0.98,16.40,0.99,18.75,1,25.16)),
179:NEXT(1998$);

1998$    ASSIGN:    R.Vehicle Enters715_730.NumberOut=R.Vehicle Enters715_730.NumberOut +
1:NEXT(79$);

```

```

;
;
;   Model statements for module: Decide 1744
;
79$   BRANCH,      1:
        If,Entity.CreateTime<2800,2001$,Yes:
        Else,2002$,Yes;
2001$   ASSIGN:      Decide 1744.NumberOut True=Decide 1744.NumberOut True + 1:NEXT(3$);

2002$   ASSIGN:      Decide 1744.NumberOut False=Decide 1744.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 393
;

2003$   CREATE,      1,SecondstoBaseTime(2800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.39,0.2,1.90,0.3,2.45,0.4,3.11,0.5,
3.90,0.6,4.85,0.7,6.02,0.8,7.75,0.9,10.71,0.95,13.27,0.98,16.89,0.99,19.27,1,25.70)),
174:NEXT(2004$);

2004$   ASSIGN:      R.Vehicle Enters730_745.NumberOut=R.Vehicle Enters730_745.NumberOut +
1:NEXT(80$);

;
;
;   Model statements for module: Decide 1745
;
80$   BRANCH,      1:
        If,Entity.CreateTime<3700,2007$,Yes:
        Else,2008$,Yes;
2007$   ASSIGN:      Decide 1745.NumberOut True=Decide 1745.NumberOut True + 1:NEXT(3$);

2008$   ASSIGN:      Decide 1745.NumberOut False=Decide 1745.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 394
;

2009$   CREATE,      1,SecondstoBaseTime(3700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.69,0.02,0.86,0.05,1.13,0.1,1.48,0.2,2.08,0.3,2.70,0.4,3.45,0.5,
4.36,0.6,5.49,0.7,6.87,0.8,8.81,0.9,12.20,0.95,15.40,0.98,19.26,0.99,21.79,1,29.35)),
153:NEXT(2010$);

```

```

2010$    ASSIGN:    R.Vehicle Enters745_800.NumberOut=R.Vehicle Enters745_800.NumberOut +
1:NEXT(81$);

;
;
;    Model statements for module: Decide 1746
;
81$      BRANCH,    1:
                If,Entity.CreateTime<4600,2013$,Yes:
                Else,2014$,Yes;
2013$    ASSIGN:    Decide 1746.NumberOut True=Decide 1746.NumberOut True + 1:NEXT(3$);
2014$    ASSIGN:    Decide 1746.NumberOut False=Decide 1746.NumberOut False + 1:NEXT(77$);

;
;
;    Model statements for module: Create 395
;
2015$    CREATE,    1,SecondstoBaseTime(4600),Vehicles:
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.83,0.05,1.10,0.1,1.41,0.2,1.94,0.3,2.51,0.4,3.19,0.5,
4.00,0.6,4.99,0.7,6.20,0.8,7.98,0.9,11.03,0.95,13.73,0.98,17.40,0.99,19.81,1,26.27)),
169:NEXT(2016$);

2016$    ASSIGN:    R.Vehicle Enters800_815.NumberOut=R.Vehicle Enters800_815.NumberOut +
1:NEXT(82$);

;
;
;    Model statements for module: Decide 1747
;
82$      BRANCH,    1:
                If,Entity.CreateTime<5500,2019$,Yes:
                Else,2020$,Yes;
2019$    ASSIGN:    Decide 1747.NumberOut True=Decide 1747.NumberOut True + 1:NEXT(3$);
2020$    ASSIGN:    Decide 1747.NumberOut False=Decide 1747.NumberOut False + 1:NEXT(77$);

;
;
;    Model statements for module: Create 396
;
2021$    CREATE,    1,SecondstoBaseTime(5500),Vehicles:
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.83,0.05,1.10,0.1,1.41,0.2,1.94,0.3,2.51,0.4,3.19,0.5,
4.00,0.6,4.99,0.7,6.20,0.8,7.98,0.9,11.03,0.95,13.73,0.98,17.40,0.99,19.81,1,26.27)),
169:NEXT(2022$);

```



```

2022$    ASSIGN:    R.Vehicle Enters815_830.NumberOut=R.Vehicle Enters815_830.NumberOut +
1:NEXT(83$);

;
;
;   Model statements for module: Decide 1748
;
83$      BRANCH,    1:
                If,Entity.CreateTime<6400,2025$,Yes:
                Else,2026$,Yes;
2025$    ASSIGN:    Decide 1748.NumberOut True=Decide 1748.NumberOut True + 1:NEXT(3$);
2026$    ASSIGN:    Decide 1748.NumberOut False=Decide 1748.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 397
;
2027$    CREATE,    1,SecondstoBaseTime(6400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.69,0.02,0.85,0.05,1.12,0.1,1.45,0.2,2.03,0.3,2.62,0.4,3.34,0.5,
4.22,0.6,5.29,0.7,6.60,0.8,8.48,0.9,11.73,0.95,14.74,0.98,18.52,0.99,21.00,1,27.52)),
159:NEXT(2028$);

2028$    ASSIGN:    R.Vehicle Enters830_845.NumberOut=R.Vehicle Enters830_845.NumberOut +
1:NEXT(84$);

;
;
;   Model statements for module: Decide 1749
;
84$      BRANCH,    1:
                If,Entity.CreateTime<7300,2031$,Yes:
                Else,2032$,Yes;
2031$    ASSIGN:    Decide 1749.NumberOut True=Decide 1749.NumberOut True + 1:NEXT(3$);
2032$    ASSIGN:    Decide 1749.NumberOut False=Decide 1749.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 398
;
2033$    CREATE,    1,SecondstoBaseTime(7300),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.87,0.05,1.14,0.1,1.52,0.2,2.15,0.3,2.80,0.4,3.58,0.5,
4.54,0.6,5.74,0.7,7.20,0.8,9.23,0.9,12.79,0.95,16.25,0.98,20.20,0.99,22.80,1,29.41)),

```

```

146:NEXT(2034$);

2034$    ASSIGN:    R.Vehicle Enters845_900.NumberOut=R.Vehicle Enters845_900.NumberOut +
1:NEXT(85$);

;
;
;    Model statements for module: Decide 1750
;
85$      BRANCH,    1:
                If,Entity.CreateTime<8200,2037$,Yes:
                Else,2038$,Yes;
2037$    ASSIGN:    Decide 1750.NumberOut True=Decide 1750.NumberOut True + 1:NEXT(3$);
2038$    ASSIGN:    Decide 1750.NumberOut False=Decide 1750.NumberOut False + 1:NEXT(77$);

;
;
;    Model statements for module: Create 399
;
2039$    CREATE,    1,SecondstoBaseTime(8200),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.86,0.05,1.13,0.1,1.48,0.2,2.09,0.3,2.71,0.4,3.47,0.5,
4.38,0.6,5.52,0.7,6.91,0.8,8.87,0.9,12.28,0.95,15.52,0.98,19.39,0.99,21.93,1,28.50)),
152:NEXT(2040$);

2040$    ASSIGN:    R.Vehicle Enters900_915.NumberOut=R.Vehicle Enters900_915.NumberOut +
1:NEXT(86$);

;
;
;    Model statements for module: Decide 1751
;
86$      BRANCH,    1:
                If,Entity.CreateTime<9100,2043$,Yes:
                Else,2044$,Yes;
2043$    ASSIGN:    Decide 1751.NumberOut True=Decide 1751.NumberOut True + 1:NEXT(3$);
2044$    ASSIGN:    Decide 1751.NumberOut False=Decide 1751.NumberOut False + 1:NEXT(77$);

;
;
;    Model statements for module: Create 400
;
2045$    CREATE,    1,SecondstoBaseTime(9100),Vehicles:

```

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.87,0.05,1.14,0.1,1.52,0.2,2.15,0.3,2.80,0.4,3.58,0.5,4.54,0.6,5.74,0.7,7.20,0.8,9.23,0.9,12.79,0.95,16.25,0.98,20.20,0.99,22.80,1,29.41)),
146:NEXT(2046\$);

2046\$ ASSIGN: R.Vehicle Enters915_930.NumberOut=R.Vehicle Enters915_930.NumberOut +
1:NEXT(87\$);

;
;
; Model statements for module: Decide 1752
;

87\$ BRANCH, 1:
 If,Entity.CreateTime<10000,2049\$,Yes:
 Else,2050\$,Yes;

2049\$ ASSIGN: Decide 1752.NumberOut True=Decide 1752.NumberOut True + 1:NEXT(3\$);

2050\$ ASSIGN: Decide 1752.NumberOut False=Decide 1752.NumberOut False + 1:NEXT(77\$);

;
;
; Model statements for module: Create 401
;

2051\$ CREATE, 1,SecondstoBaseTime(10000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.68,0.02,0.84,0.05,1.11,0.1,1.43,0.2,1.99,0.3,2.58,0.4,3.28,0.5,4.13,0.6,5.16,0.7,6.44,0.8,8.27,0.9,11.44,0.95,14.32,0.98,18.05,0.99,20.51,1,27.00)),
163:NEXT(2052\$);

2052\$ ASSIGN: R.Vehicle Enters930_945.NumberOut=R.Vehicle Enters930_945.NumberOut +
1:NEXT(88\$);

;
;
; Model statements for module: Decide 1753
;

88\$ BRANCH, 1:
 If,Entity.CreateTime<10900,2055\$,Yes:
 Else,2056\$,Yes;

2055\$ ASSIGN: Decide 1753.NumberOut True=Decide 1753.NumberOut True + 1:NEXT(3\$);

2056\$ ASSIGN: Decide 1753.NumberOut False=Decide 1753.NumberOut False + 1:NEXT(77\$);

;
;
; Model statements for module: Create 402
;

```

2057$    CREATE,      1,SecondstoBaseTime(10900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.86,0.05,1.14,0.1,1.50,0.2,2.12,0.3,2.76,0.4,3.52,0.5,
4.46,0.6,5.63,0.7,7.05,0.8,9.04,0.9,12.53,0.95,15.88,0.98,19.79,0.99,22.36,1,28.94)),
149:NEXT(2058$);

2058$    ASSIGN:      R.Vehicle Enters945_1000.NumberOut=R.Vehicle Enters945_1000.NumberOut
+ 1:NEXT(89$);

;
;
;   Model statements for module: Decide 1754
;
89$      BRANCH,      1:
                If,Entity.CreateTime<11800,2061$,Yes:
                Else,2062$,Yes;
2061$    ASSIGN:      Decide 1754.NumberOut True=Decide 1754.NumberOut True + 1:NEXT(3$);
2062$    ASSIGN:      Decide 1754.NumberOut False=Decide 1754.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 403
;

2063$    CREATE,      1,SecondstoBaseTime(11800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.87,0.05,1.14,0.1,1.50,0.2,2.13,0.3,2.77,0.4,3.54,0.5,
4.49,0.6,5.66,0.7,7.10,0.8,9.10,0.9,12.62,0.95,16.00,0.98,19.92,0.99,22.50,1,29.10)),
148:NEXT(2064$);

2064$    ASSIGN:      R.Vehicle Enters1000_1015.NumberOut=R.Vehicle
Enters1000_1015.NumberOut + 1:NEXT(90$);

;
;
;   Model statements for module: Decide 1755
;
90$      BRANCH,      1:
                If,Entity.CreateTime<12700,2067$,Yes:
                Else,2068$,Yes;
2067$    ASSIGN:      Decide 1755.NumberOut True=Decide 1755.NumberOut True + 1:NEXT(3$);
2068$    ASSIGN:      Decide 1755.NumberOut False=Decide 1755.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 404
;

```

```

2069$    CREATE,      1,SecondstoBaseTime(12700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.68,0.02,0.83,0.05,1.10,0.1,1.42,0.2,1.97,0.3,2.55,0.4,3.25,0.5,
4.08,0.6,5.11,0.7,6.36,0.8,8.17,0.9,11.30,0.95,14.12,0.98,17.83,0.99,20.27,1,26.75)),
          165:NEXT(2070$);

2070$    ASSIGN:      R.Vehicle Enters1015_1030.NumberOut=R.Vehicle
Enters1015_1030.NumberOut + 1:NEXT(91$);

;
;
; Model statements for module: Decide 1756
;
91$      BRANCH,      1:
          If,Entity.CreateTime<13600,2073$,Yes:
          Else,2074$,Yes;

2073$    ASSIGN:      Decide 1756.NumberOut True=Decide 1756.NumberOut True + 1:NEXT(3$);

2074$    ASSIGN:      Decide 1756.NumberOut False=Decide 1756.NumberOut False + 1:NEXT(77$);

;
;
; Model statements for module: Create 405
;

2075$    CREATE,      1,SecondstoBaseTime(13600),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1.07,0.1,1.35,0.2,1.83,0.3,2.36,0.4,2.98,0.5,
3.72,0.6,4.60,0.7,5.69,0.8,7.33,0.9,10.12,0.95,12.42,0.98,15.75,0.99,18.27,1,24.65)),
          184:NEXT(2076$);

2076$    ASSIGN:      R.Vehicle Enters1030_1045.NumberOut=R.Vehicle
Enters1030_1045.NumberOut + 1:NEXT(92$);

;
;
; Model statements for module: Decide 1757
;
92$      BRANCH,      1:
          If,Entity.CreateTime<14500,2079$,Yes:
          Else,2080$,Yes;

2079$    ASSIGN:      Decide 1757.NumberOut True=Decide 1757.NumberOut True + 1:NEXT(3$);

2080$    ASSIGN:      Decide 1757.NumberOut False=Decide 1757.NumberOut False + 1:NEXT(77$);

;
;
; Model statements for module: Create 406

```

```

;
2081$    CREATE,      1,SecondstoBaseTime(14500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.83,0.05,1.10,0.1,1.40,0.2,1.93,0.3,2.50,0.4,3.17,0.5,
3.98,0.6,4.96,0.7,6.16,0.8,7.93,0.9,10.96,0.95,13.64,0.98,17.29,0.99,19.70,1,26.15)),
          170:NEXT(2082$);

2082$    ASSIGN:      R.Vehicle Enters1045_1100.NumberOut=R.Vehicle
Enters1045_1100.NumberOut + 1:NEXT(93$);

;
;
; Model statements for module: Decide 1758
;
93$      BRANCH,      1:
          If,Entity.CreateTime<15400,2085$,Yes:
          Else,2086$,Yes;
2085$    ASSIGN:      Decide 1758.NumberOut True=Decide 1758.NumberOut True + 1:NEXT(3$);
2086$    ASSIGN:      Decide 1758.NumberOut False=Decide 1758.NumberOut False + 1:NEXT(77$);

;
;
; Model statements for module: Create 407
;

2087$    CREATE,      1,SecondstoBaseTime(15400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1.08,0.1,1.36,0.2,1.85,0.3,2.38,0.4,3.02,0.5,
3.77,0.6,4.68,0.7,5.78,0.8,7.45,0.9,10.29,0.95,12.67,0.98,16.22,0.99,18.55,1,24.95)),
          181:NEXT(2088$);

2088$    ASSIGN:      R.Vehicle Enters1100_1115.NumberOut=R.Vehicle
Enters1100_1115.NumberOut + 1:NEXT(94$);

;
;
; Model statements for module: Decide 1759
;
94$      BRANCH,      1:
          If,Entity.CreateTime<16300,2091$,Yes:
          Else,2092$,Yes;
2091$    ASSIGN:      Decide 1759.NumberOut True=Decide 1759.NumberOut True + 1:NEXT(3$);
2092$    ASSIGN:      Decide 1759.NumberOut False=Decide 1759.NumberOut False + 1:NEXT(77$);

;
;

```

```
; Model statements for module: Create 408
;
```

```
2093$ CREATE, 1,SecondstoBaseTime(16300),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.86,0.05,1.14,0.1,1.50,0.2,2.12,0.3,2.76,0.4,3.52,0.5,
4.46,0.6,5.63,0.7,7.05,0.8,9.04,0.9,12.53,0.95,15.88,0.98,19.79,0.99,22.36,1,28.94)),
149:NEXT(2094$);
```

```
2094$ ASSIGN: R.Vehicle Enters1115_1130.NumberOut=R.Vehicle
Enters1115_1130.NumberOut + 1:NEXT(95$);
```

```
;
;
; Model statements for module: Decide 1760
;
```

```
95$ BRANCH, 1:
If,Entity.CreateTime<17200,2097$,Yes:
Else,2098$,Yes;
```

```
2097$ ASSIGN: Decide 1760.NumberOut True=Decide 1760.NumberOut True + 1:NEXT(3$);
```

```
2098$ ASSIGN: Decide 1760.NumberOut False=Decide 1760.NumberOut False + 1:NEXT(77$);
```

```
;
;
; Model statements for module: Create 409
;
```

```
2099$ CREATE, 1,SecondstoBaseTime(17200),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.68,0.02,0.84,0.05,1.11,0.1,1.43,0.2,1.98,0.3,2.56,0.4,3.26,0.5,
4.11,0.6,5.13,0.7,6.40,0.8,8.22,0.9,11.37,0.95,14.22,0.98,17.94,0.99,20.39,1,26.88)),
164:NEXT(2100$);
```

```
2100$ ASSIGN: R.Vehicle Enters1130_1145.NumberOut=R.Vehicle
Enters1130_1145.NumberOut + 1:NEXT(96$);
```

```
;
;
; Model statements for module: Decide 1761
;
```

```
96$ BRANCH, 1:
If,Entity.CreateTime<18100,2103$,Yes:
Else,2104$,Yes;
```

```
2103$ ASSIGN: Decide 1761.NumberOut True=Decide 1761.NumberOut True + 1:NEXT(3$);
```

```
2104$ ASSIGN: Decide 1761.NumberOut False=Decide 1761.NumberOut False + 1:NEXT(77$);
```

```
;
```

```
;
; Model statements for module: Create 410
;
```

```
2105$ CREATE, 1,SecondstoBaseTime(18100),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1.08,0.1,1.36,0.2,1.85,0.3,2.38,0.4,3.02,0.5,
3.77,0.6,4.68,0.7,5.78,0.8,7.45,0.9,10.29,0.95,12.67,0.98,16.22,0.99,18.55,1,24.95)),
181:NEXT(2106$);
```

```
2106$ ASSIGN: R.Vehicle Enters1145_1200.NumberOut=R.Vehicle
Enters1145_1200.NumberOut + 1:NEXT(97$);
```

```
;
;
; Model statements for module: Decide 1762
;
```

```
97$ BRANCH, 1:
If,Entity.CreateTime<19000,2109$,Yes:
Else,2110$,Yes;
```

```
2109$ ASSIGN: Decide 1762.NumberOut True=Decide 1762.NumberOut True + 1:NEXT(3$);
```

```
2110$ ASSIGN: Decide 1762.NumberOut False=Decide 1762.NumberOut False + 1:NEXT(77$);
```

```
;
;
; Model statements for module: Create 411
;
```

```
2111$ CREATE, 1,SecondstoBaseTime(19000),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.69,0.02,0.86,0.05,1.13,0.1,1.48,0.2,2.08,0.3,2.70,0.4,3.45,0.5,
4.36,0.6,5.49,0.7,6.87,0.8,8.81,0.9,12.20,0.95,15.40,0.98,19.26,0.99,21.79,1,29.35)),
153:NEXT(2112$);
```

```
2112$ ASSIGN: R.Vehicle Enters1200_1215.NumberOut=R.Vehicle
Enters1200_1215.NumberOut + 1:NEXT(98$);
```

```
;
;
; Model statements for module: Decide 1763
;
```

```
98$ BRANCH, 1:
If,Entity.CreateTime<19900,2115$,Yes:
Else,2116$,Yes;
```

```
2115$ ASSIGN: Decide 1763.NumberOut True=Decide 1763.NumberOut True + 1:NEXT(3$);
```

```
2116$ ASSIGN: Decide 1763.NumberOut False=Decide 1763.NumberOut False + 1:NEXT(77$);
```



```

;
;
;   Model statements for module: Create 412
;

2117$   CREATE,      1,SecondstoBaseTime(19900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.83,0.05,1.10,0.1,1.40,0.2,1.93,0.3,2.50,0.4,3.17,0.5,
3.98,0.6,4.96,0.7,6.16,0.8,7.93,0.9,10.96,0.95,13.64,0.98,17.29,0.99,19.70,1,26.15)),
          170:NEXT(2118$);

2118$   ASSIGN:      R.Vehicle Enters1215_1230.NumberOut=R.Vehicle
Enters1215_1230.NumberOut + 1:NEXT(99$);

;
;
;   Model statements for module: Decide 1764
;
99$     BRANCH,      1:
          If,Entity.CreateTime<20800,2121$,Yes:
          Else,2122$,Yes;
2121$   ASSIGN:      Decide 1764.NumberOut True=Decide 1764.NumberOut True + 1:NEXT(3$);
2122$   ASSIGN:      Decide 1764.NumberOut False=Decide 1764.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 413
;

2123$   CREATE,      1,SecondstoBaseTime(20800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.81,0.05,1.08,0.1,1.37,0.2,1.88,0.3,2.41,0.4,3.06,0.5,
3.83,0.6,4.75,0.7,5.88,0.8,7.57,0.9,10.46,0.95,12.92,0.98,16.50,0.99,18.85,1,25.26)),
          178:NEXT(2124$);

2124$   ASSIGN:      R.Vehicle Enters1230_1245.NumberOut=R.Vehicle
Enters1230_1245.NumberOut + 1:NEXT(100$);

;
;
;   Model statements for module: Decide 1765
;
100$    BRANCH,      1:
          If,Entity.CreateTime<21700,2127$,Yes:
          Else,2128$,Yes;
2127$   ASSIGN:      Decide 1765.NumberOut True=Decide 1765.NumberOut True + 1:NEXT(3$);
2128$   ASSIGN:      Decide 1765.NumberOut False=Decide 1765.NumberOut False + 1:NEXT(77$);

```

```

;
;
;   Model statements for module: Create 414
;

2129$   CREATE,      1,SecondstoBaseTime(21700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.83,0.05,1.10,0.1,1.40,0.2,1.93,0.3,2.50,0.4,3.17,0.5,
3.98,0.6,4.96,0.7,6.16,0.8,7.93,0.9,10.96,0.95,13.64,0.98,17.29,0.99,19.70,1,26.15)),
          170:NEXT(2130$);

2130$   ASSIGN:      R.Vehicle Enters1245_1300.NumberOut=R.Vehicle
Enters1245_1300.NumberOut + 1:NEXT(101$);

;
;
;   Model statements for module: Decide 1766
;
101$    BRANCH,      1:
          If,Entity.CreateTime<22600,2133$,Yes:
          Else,2134$,Yes;
2133$   ASSIGN:      Decide 1766.NumberOut True=Decide 1766.NumberOut True + 1:NEXT(3$);
2134$   ASSIGN:      Decide 1766.NumberOut False=Decide 1766.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 415
;

2135$   CREATE,      1,SecondstoBaseTime(22600),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.68,0.02,0.84,0.05,1.11,0.1,1.44,0.2,2.01,0.3,2.60,0.4,3.31,0.5,
4.17,0.6,5.23,0.7,6.52,0.8,8.37,0.9,11.59,0.95,14.53,0.98,18.28,0.99,20.75,1,27.26)),
          161:NEXT(2136$);

2136$   ASSIGN:      R.Vehicle Enters1300_1315.NumberOut=R.Vehicle
Enters1300_1315.NumberOut + 1:NEXT(102$);

;
;
;   Model statements for module: Decide 1767
;
102$    BRANCH,      1:
          If,Entity.CreateTime<23500,2139$,Yes:
          Else,2140$,Yes;
2139$   ASSIGN:      Decide 1767.NumberOut True=Decide 1767.NumberOut True + 1:NEXT(3$);
2140$   ASSIGN:      Decide 1767.NumberOut False=Decide 1767.NumberOut False + 1:NEXT(77$);

```

```

;
;
;   Model statements for module: Create 416
;

2141$   CREATE,      1,SecondstoBaseTime(23500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.83,0.05,1.10,0.1,1.40,0.2,1.93,0.3,2.50,0.4,3.17,0.5,
3.98,0.6,4.96,0.7,6.16,0.8,7.93,0.9,10.96,0.95,13.64,0.98,17.29,0.99,19.70,1,26.15)),
          170:NEXT(2142$);

2142$   ASSIGN:      R.Vehicle Enters1315_1330.NumberOut=R.Vehicle
Enters1315_1330.NumberOut + 1:NEXT(103$);

;
;
;   Model statements for module: Decide 1768
;

103$    BRANCH,      1:
          If,Entity.CreateTime<24400,2145$,Yes:
          Else,2146$,Yes;

2145$    ASSIGN:      Decide 1768.NumberOut True=Decide 1768.NumberOut True + 1:NEXT(3$);

2146$    ASSIGN:      Decide 1768.NumberOut False=Decide 1768.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 417
;

2147$    CREATE,      1,SecondstoBaseTime(24400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.87,0.05,1.14,0.1,1.52,0.2,2.15,0.3,2.80,0.4,3.58,0.5,
4.54,0.6,5.74,0.7,7.20,0.8,9.23,0.9,12.79,0.95,16.25,0.98,20.20,0.99,22.80,1,29.41)),
          146:NEXT(2148$);

2148$    ASSIGN:      R.Vehicle Enters1330_1345.NumberOut=R.Vehicle
Enters1330_1345.NumberOut + 1:NEXT(104$);

;
;
;   Model statements for module: Decide 1769
;

104$    BRANCH,      1:
          If,Entity.CreateTime<25300,2151$,Yes:
          Else,2152$,Yes;

2151$    ASSIGN:      Decide 1769.NumberOut True=Decide 1769.NumberOut True + 1:NEXT(3$);

```

2152\$ ASSIGN: Decide 1769.NumberOut False=Decide 1769.NumberOut False + 1:NEXT(77\$);

;
;
;
; Model statements for module: Create 418
;

2153\$ CREATE, 1,SecondstoBaseTime(25300),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.38,0.2,1.89,0.3,2.43,0.4,3.09,0.5,
3.86,0.6,4.80,0.7,5.95,0.8,7.66,0.9,10.58,0.95,13.09,0.98,16.69,0.99,19.06,1,25.48)),
176:NEXT(2154\$);

2154\$ ASSIGN: R.Vehicle Enters1345_1400.NumberOut=R.Vehicle
Enters1345_1400.NumberOut + 1:NEXT(105\$);

;
;
;
; Model statements for module: Decide 1770
;

105\$ BRANCH, 1:
 If,Entity.CreateTime<26200,2157\$,Yes:
 Else,2158\$,Yes;

2157\$ ASSIGN: Decide 1770.NumberOut True=Decide 1770.NumberOut True + 1:NEXT(3\$);

2158\$ ASSIGN: Decide 1770.NumberOut False=Decide 1770.NumberOut False + 1:NEXT(77\$);

;
;
;
; Model statements for module: Create 419
;

2159\$ CREATE, 1,SecondstoBaseTime(26200),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.39,0.2,1.92,0.3,2.48,0.4,3.14,0.5,
3.94,0.6,4.91,0.7,6.09,0.8,7.84,0.9,10.83,0.95,13.45,0.98,17.09,0.99,19.48,1,25.92)),
172:NEXT(2160\$);

2160\$ ASSIGN: R.Vehicle Enters1400_1415.NumberOut=R.Vehicle
Enters1400_1415.NumberOut + 1:NEXT(106\$);

;
;
;
; Model statements for module: Decide 1771
;

106\$ BRANCH, 1:
 If,Entity.CreateTime<27100,2163\$,Yes:
 Else,2164\$,Yes;

2163\$ ASSIGN: Decide 1771.NumberOut True=Decide 1771.NumberOut True + 1:NEXT(3\$);

2164\$ ASSIGN: Decide 1771.NumberOut False=Decide 1771.NumberOut False + 1:NEXT(77\$);

;
;
;
; Model statements for module: Create 420
;

2165\$ CREATE, 1,SecondstoBaseTime(27100),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1.08,0.1,1.36,0.2,1.85,0.3,2.38,0.4,3.02,0.5,
3.77,0.6,4.68,0.7,5.78,0.8,7.45,0.9,10.29,0.95,12.67,0.98,16.22,0.99,18.55,1,24.95)),
181:NEXT(2166\$);

2166\$ ASSIGN: R.Vehicle Enters1415_1430.NumberOut=R.Vehicle
Enters1415_1430.NumberOut + 1:NEXT(107\$);

;
;
;
; Model statements for module: Decide 1772
;

107\$ BRANCH, 1:
 If,Entity.CreateTime<28000,2169\$,Yes:
 Else,2170\$,Yes;

2169\$ ASSIGN: Decide 1772.NumberOut True=Decide 1772.NumberOut True + 1:NEXT(3\$);

2170\$ ASSIGN: Decide 1772.NumberOut False=Decide 1772.NumberOut False + 1:NEXT(77\$);

;
;
;
; Model statements for module: Create 421
;

2171\$ CREATE, 1,SecondstoBaseTime(28000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.38,0.2,1.90,0.3,2.44,0.4,3.10,0.5,
3.88,0.6,4.83,0.7,5.98,0.8,7.70,0.9,10.65,0.95,13.18,0.98,16.79,0.99,19.16,1,25.59)),
175:NEXT(2172\$);

2172\$ ASSIGN: R.Vehicle Enters1430_1445.NumberOut=R.Vehicle
Enters1430_1445.NumberOut + 1:NEXT(108\$);

;
;
;
; Model statements for module: Decide 1773
;

108\$ BRANCH, 1:
 If,Entity.CreateTime<28900,2175\$,Yes:
 Else,2176\$,Yes;

```

2175$    ASSIGN:    Decide 1773.NumberOut True=Decide 1773.NumberOut True + 1:NEXT(3$);

2176$    ASSIGN:    Decide 1773.NumberOut False=Decide 1773.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 422
;

2177$    CREATE,    1,SecondstoBaseTime(28900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.38,0.2,1.89,0.3,2.43,0.4,3.09,0.5,
3.86,0.6,4.80,0.7,5.95,0.8,7.66,0.9,10.58,0.95,13.09,0.98,16.69,0.99,19.06,1,25.48)),
176:NEXT(2178$);

2178$    ASSIGN:    R.Vehicle Enters1445_1500.NumberOut=R.Vehicle
Enters1445_1500.NumberOut + 1:NEXT(109$);

;
;
;   Model statements for module: Decide 1774
;

109$     BRANCH,    1:
                If,Entity.CreateTime<29800,2181$,Yes:
                Else,2182$,Yes;

2181$    ASSIGN:    Decide 1774.NumberOut True=Decide 1774.NumberOut True + 1:NEXT(3$);

2182$    ASSIGN:    Decide 1774.NumberOut False=Decide 1774.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 423
;

2183$    CREATE,    1,SecondstoBaseTime(29800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.38,0.2,1.89,0.3,2.43,0.4,3.09,0.5,
3.86,0.6,4.80,0.7,5.95,0.8,7.66,0.9,10.58,0.95,13.09,0.98,16.69,0.99,19.06,1,25.48)),
176:NEXT(2184$);

2184$    ASSIGN:    R.Vehicle Enters1500_1515.NumberOut=R.Vehicle
Enters1500_1515.NumberOut + 1:NEXT(110$);

;
;
;   Model statements for module: Decide 1775
;

110$     BRANCH,    1:
                If,Entity.CreateTime<30700,2187$,Yes:

```

```

Else,2188$,Yes;
2187$    ASSIGN:    Decide 1775.NumberOut True=Decide 1775.NumberOut True + 1:NEXT(3$);

2188$    ASSIGN:    Decide 1775.NumberOut False=Decide 1775.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 424
;

2189$    CREATE,    1,SecondstoBaseTime(30700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1.07,0.1,1.35,0.2,1.83,0.3,2.36,0.4,2.98,0.5,
3.72,0.6,4.60,0.7,5.69,0.8,7.33,0.9,10.12,0.95,12.42,0.98,15.75,0.99,18.27,1,24.65)),
184:NEXT(2190$);

2190$    ASSIGN:    R.Vehicle Enters1515_1530.NumberOut=R.Vehicle
Enters1515_1530.NumberOut + 1:NEXT(111$);

;
;
;   Model statements for module: Decide 1776
;

111$    BRANCH,    1:
                If,Entity.CreateTime<31600,2193$,Yes:
                Else,2194$,Yes;
2193$    ASSIGN:    Decide 1776.NumberOut True=Decide 1776.NumberOut True + 1:NEXT(3$);

2194$    ASSIGN:    Decide 1776.NumberOut False=Decide 1776.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 425
;

2195$    CREATE,    1,SecondstoBaseTime(31600),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1.08,0.1,1.36,0.2,1.85,0.3,2.38,0.4,3.02,0.5,
3.77,0.6,4.68,0.7,5.78,0.8,7.45,0.9,10.29,0.95,12.67,0.98,16.22,0.99,18.55,1,24.95)),
181:NEXT(2196$);

2196$    ASSIGN:    R.Vehicle Enters1530_1545.NumberOut=R.Vehicle
Enters1530_1545.NumberOut + 1:NEXT(112$);

;
;
;   Model statements for module: Decide 1777
;

112$    BRANCH,    1:

```

```

                If,Entity.CreateTime<32500,2199$,Yes:
                Else,2200$,Yes;
2199$    ASSIGN:    Decide 1777.NumberOut True=Decide 1777.NumberOut True + 1:NEXT(3$);

2200$    ASSIGN:    Decide 1777.NumberOut False=Decide 1777.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 426
;

2201$    CREATE,    1,SecondstoBaseTime(32500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.81,0.05,1.08,0.1,1.37,0.2,1.88,0.3,2.41,0.4,3.06,0.5,
3.83,0.6,4.75,0.7,5.88,0.8,7.57,0.9,10.46,0.95,12.92,0.98,16.50,0.99,18.85,1,25.26)),
178:NEXT(2202$);

2202$    ASSIGN:    R.Vehicle Enters1545_1600.NumberOut=R.Vehicle
Enters1545_1600.NumberOut + 1:NEXT(113$);

;
;
;   Model statements for module: Decide 1778
;

113$    BRANCH,    1:
                If,Entity.CreateTime<33400,2205$,Yes:
                Else,2206$,Yes;
2205$    ASSIGN:    Decide 1778.NumberOut True=Decide 1778.NumberOut True + 1:NEXT(3$);

2206$    ASSIGN:    Decide 1778.NumberOut False=Decide 1778.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 427
;

2207$    CREATE,    1,SecondstoBaseTime(33400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.86,0.05,1.13,0.1,1.49,0.2,2.11,0.3,2.74,0.4,3.50,0.5,
4.43,0.6,5.59,0.7,7.01,0.8,8.98,0.9,12.45,0.95,15.76,0.98,19.65,0.99,22.21,1,28.79)),
150:NEXT(2208$);

2208$    ASSIGN:    R.Vehicle Enters1600_1615.NumberOut=R.Vehicle
Enters1600_1615.NumberOut + 1:NEXT(114$);

;
;
;   Model statements for module: Decide 1779
;

```



```

114$    BRANCH,    1:
          If,Entity.CreateTime<34300,2211$,Yes:
          Else,2212$,Yes;
2211$    ASSIGN:    Decide 1779.NumberOut True=Decide 1779.NumberOut True + 1:NEXT(3$);
2212$    ASSIGN:    Decide 1779.NumberOut False=Decide 1779.NumberOut False + 1:NEXT(77$);

;
;
;    Model statements for module: Create 428
;

2213$    CREATE,    1,SecondstoBaseTime(34300),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.80,0.05,1.07,0.1,1.34,0.2,1.82,0.3,2.34,0.4,2.96,0.5,
3.69,0.6,4.56,0.7,5.62,0.8,7.25,0.9,10.01,0.95,12.27,0.98,15.77,0.99,18.08,1,24.45)),
          186:NEXT(2214$);

2214$    ASSIGN:    R.Vehicle Enters1615_1630.NumberOut=R.Vehicle
Enters1615_1630.NumberOut + 1:NEXT(115$);

;
;
;    Model statements for module: Decide 1780
;

115$    BRANCH,    1:
          If,Entity.CreateTime<35200,2217$,Yes:
          Else,2218$,Yes;
2217$    ASSIGN:    Decide 1780.NumberOut True=Decide 1780.NumberOut True + 1:NEXT(3$);
2218$    ASSIGN:    Decide 1780.NumberOut False=Decide 1780.NumberOut False + 1:NEXT(77$);

;
;
;    Model statements for module: Create 429
;

2219$    CREATE,    1,SecondstoBaseTime(35200),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.80,0.05,1.07,0.1,1.33,0.2,1.80,0.3,2.31,0.4,2.92,0.5,
3.64,0.6,4.49,0.7,5.53,0.8,7.13,0.9,9.84,0.95,12.04,0.98,15.52,0.99,17.80,1,24.16)),
          189:NEXT(2220$);

2220$    ASSIGN:    R.Vehicle Enters1630_1645.NumberOut=R.Vehicle
Enters1630_1645.NumberOut + 1:NEXT(116$);

;
;
;    Model statements for module: Decide 1781

```

```

;
116$    BRANCH,    1:
          If,Entity.CreateTime<36100,2223$,Yes:
          Else,2224$,Yes;
2223$    ASSIGN:    Decide 1781.NumberOut True=Decide 1781.NumberOut True + 1:NEXT(3$);
2224$    ASSIGN:    Decide 1781.NumberOut False=Decide 1781.NumberOut False + 1:NEXT(77$);

;
;
;    Model statements for module: Create 430
;

2225$    CREATE,    1,SecondstoBaseTime(36100),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1.07,0.1,1.35,0.2,1.83,0.3,2.36,0.4,2.98,0.5,
3.72,0.6,4.60,0.7,5.69,0.8,7.33,0.9,10.12,0.95,12.42,0.98,15.75,0.99,18.27,1,24.65)),
          185:NEXT(2226$);

2226$    ASSIGN:    R.Vehicle Enters1645_1700.NumberOut=R.Vehicle
Enters1645_1700.NumberOut + 1:NEXT(117$);

;
;
;    Model statements for module: Decide 1782
;

117$    BRANCH,    1:
          If,Entity.CreateTime<37000,2229$,Yes:
          Else,2230$,Yes;
2229$    ASSIGN:    Decide 1782.NumberOut True=Decide 1782.NumberOut True + 1:NEXT(3$);
2230$    ASSIGN:    Decide 1782.NumberOut False=Decide 1782.NumberOut False + 1:NEXT(77$);

;
;
;    Model statements for module: Create 431
;

2231$    CREATE,    1,SecondstoBaseTime(37000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.68,0.02,0.83,0.05,1.10,0.1,1.42,0.2,1.97,0.3,2.55,0.4,3.25,0.5,
4.08,0.6,5.11,0.7,6.36,0.8,8.17,0.9,11.30,0.95,14.12,0.98,17.83,0.99,20.27,1,26.75)),
          165:NEXT(2232$);

2232$    ASSIGN:    R.Vehicle Enters1700_1715.NumberOut=R.Vehicle
Enters1700_1715.NumberOut + 1:NEXT(118$);

;
;

```

```

;   Model statements for module: Decide 1783
;
118$   BRANCH,      1:
           If,Entity.CreateTime<37900,2235$,Yes:
           Else,2236$,Yes;
2235$   ASSIGN:      Decide 1783.NumberOut True=Decide 1783.NumberOut True + 1:NEXT(3$);

2236$   ASSIGN:      Decide 1783.NumberOut False=Decide 1783.NumberOut False + 1:NEXT(77$);


;
;
;   Model statements for module: Create 432
;

2237$   CREATE,      1,SecondstoBaseTime(37900),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.39,0.2,1.92,0.3,2.48,0.4,3.14,0.5,
3.94,0.6,4.91,0.7,6.09,0.8,7.84,0.9,10.83,0.95,13.45,0.98,17.09,0.99,19.48,1,25.92)),
           172:NEXT(2238$);

2238$   ASSIGN:      R.Vehicle Enters1715_1730.NumberOut=R.Vehicle
Enters1715_1730.NumberOut + 1:NEXT(119$);


;
;
;   Model statements for module: Decide 1784
;
119$   BRANCH,      1:
           If,Entity.CreateTime<38800,2241$,Yes:
           Else,2242$,Yes;
2241$   ASSIGN:      Decide 1784.NumberOut True=Decide 1784.NumberOut True + 1:NEXT(3$);

2242$   ASSIGN:      Decide 1784.NumberOut False=Decide 1784.NumberOut False + 1:NEXT(77$);


;
;
;   Model statements for module: Create 433
;

2243$   CREATE,      1,SecondstoBaseTime(38800),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.39,0.2,1.92,0.3,2.48,0.4,3.14,0.5,
3.94,0.6,4.91,0.7,6.09,0.8,7.84,0.9,10.83,0.95,13.45,0.98,17.09,0.99,19.48,1,25.92)),
           173:NEXT(2244$);

2244$   ASSIGN:      R.Vehicle Enters1730_1745.NumberOut=R.Vehicle
Enters1730_1745.NumberOut + 1:NEXT(120$);


;

```

```

;
;   Model statements for module: Decide 1785
;
120$   BRANCH,      1:
        If,Entity.CreateTime<39700,2247$,Yes:
        Else,2248$,Yes;
2247$   ASSIGN:      Decide 1785.NumberOut True=Decide 1785.NumberOut True + 1:NEXT(3$);
2248$   ASSIGN:      Decide 1785.NumberOut False=Decide 1785.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 434
;
2249$   CREATE,      1,SecondstoBaseTime(39700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.86,0.05,1.13,0.1,1.48,0.2,2.09,0.3,2.71,0.4,3.47,0.5,
4.38,0.6,5.52,0.7,6.91,0.8,8.87,0.9,12.28,0.95,15.52,0.98,19.39,0.99,21.93,1,28.50)),
        152:NEXT(2250$);

2250$   ASSIGN:      R.Vehicle Enters1745_1800.NumberOut=R.Vehicle
Enters1745_1800.NumberOut + 1:NEXT(121$);

;
;
;   Model statements for module: Decide 1786
;
121$   BRANCH,      1:
        If,Entity.CreateTime<40600,2253$,Yes:
        Else,2254$,Yes;
2253$   ASSIGN:      Decide 1786.NumberOut True=Decide 1786.NumberOut True + 1:NEXT(3$);
2254$   ASSIGN:      Decide 1786.NumberOut False=Decide 1786.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 435
;
2255$   CREATE,      1,SecondstoBaseTime(40600),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.69,0.02,0.85,0.05,1.12,0.1,1.45,0.2,2.03,0.3,2.62,0.4,3.34,0.5,
4.22,0.6,5.29,0.7,6.60,0.8,8.48,0.9,11.73,0.95,14.74,0.98,18.52,0.99,21.00,1,27.52)),
        159:NEXT(2256$);

2256$   ASSIGN:      R.Vehicle Enters1800_1815.NumberOut=R.Vehicle
Enters1800_1815.NumberOut + 1:NEXT(122$);

```

```

;
;
;   Model statements for module: Decide 1787
;
122$   BRANCH,      1:
        If,Entity.CreateTime<41500,2259$,Yes:
        Else,2260$,Yes;
2259$   ASSIGN:      Decide 1787.NumberOut True=Decide 1787.NumberOut True + 1:NEXT(3$);

2260$   ASSIGN:      Decide 1787.NumberOut False=Decide 1787.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 436
;

2261$   CREATE,      1,SecondstoBaseTime(41500),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.71,0.02,0.88,0.05,1.16,0.1,1.54,0.2,2.20,0.3,2.88,0.4,3.68,0.5,
4.68,0.6,5.93,0.7,7.46,0.8,9.56,0.9,13.25,0.95,16.91,0.98,20.93,0.99,23.57,1,30.23)),
        141:NEXT(2262$);

2262$   ASSIGN:      R.Vehicle Enters1815_1830.NumberOut=R.Vehicle
Enters1815_1830.NumberOut + 1:NEXT(123$);

;
;
;   Model statements for module: Decide 1788
;
123$   BRANCH,      1:
        If,Entity.CreateTime<42400,2265$,Yes:
        Else,2266$,Yes;
2265$   ASSIGN:      Decide 1788.NumberOut True=Decide 1788.NumberOut True + 1:NEXT(3$);

2266$   ASSIGN:      Decide 1788.NumberOut False=Decide 1788.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 437
;

2267$   CREATE,      1,SecondstoBaseTime(42400),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.71,0.02,0.88,0.05,1.15,0.1,1.53,0.2,2.17,0.3,2.83,0.4,3.62,0.5,
4.60,0.6,5.81,0.7,7.30,0.8,9.36,0.9,12.98,0.95,16.51,0.98,20.49,0.99,23.10,1,29.73)),
        144:NEXT(2268$);

2268$   ASSIGN:      R.Vehicle Enters1830_1845.NumberOut=R.Vehicle
Enters1830_1845.NumberOut + 1:NEXT(124$);

```

```

;
;
;   Model statements for module: Decide 1789
;
124$   BRANCH,      1:
        If,Entity.CreateTime<43300,2271$,Yes:
        Else,2272$,Yes;
2271$   ASSIGN:      Decide 1789.NumberOut True=Decide 1789.NumberOut True + 1:NEXT(3$);

2272$   ASSIGN:      Decide 1789.NumberOut False=Decide 1789.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 438
;

2273$   CREATE,      1,SecondstoBaseTime(43300),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.71,0.02,0.88,0.05,1.16,0.1,1.54,0.2,2.20,0.3,2.88,0.4,3.68,0.5,
4.68,0.6,5.93,0.7,7.46,0.8,9.56,0.9,13.25,0.95,16.91,0.98,20.93,0.99,23.57,1,30.23)),
        141:NEXT(2274$);

2274$   ASSIGN:      R.Vehicle Enters1845_1900.NumberOut=R.Vehicle
Enters1845_1900.NumberOut + 1:NEXT(125$);

;
;
;   Model statements for module: Decide 1790
;
125$   BRANCH,      1:
        If,Entity.CreateTime<44200,2277$,Yes:
        Else,2278$,Yes;
2277$   ASSIGN:      Decide 1790.NumberOut True=Decide 1790.NumberOut True + 1:NEXT(3$);

2278$   ASSIGN:      Decide 1790.NumberOut False=Decide 1790.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 439
;

2279$   CREATE,      1,SecondstoBaseTime(44200),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.71,0.02,0.89,0.05,1.16,0.1,1.56,0.2,2.23,0.3,2.91,0.4,3.73,0.5,
4.74,0.6,6.01,0.7,7.57,0.8,9.69,0.9,13.45,0.95,17.19,0.98,21.24,0.99,23.90,1,30.57)),
        139:NEXT(2280$);

2280$   ASSIGN:      R.Vehicle Enters1900_1915.NumberOut=R.Vehicle
Enters1900_1915.NumberOut + 1:NEXT(126$);

```

```

;
;
;   Model statements for module: Decide 1791
;
126$   BRANCH,      1:
        If,Entity.CreateTime<45100,2283$,Yes:
        Else,2284$,Yes;
2283$   ASSIGN:      Decide 1791.NumberOut True=Decide 1791.NumberOut True + 1:NEXT(3$);

2284$   ASSIGN:      Decide 1791.NumberOut False=Decide 1791.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 440
;

2285$   CREATE,      1,SecondstoBaseTime(45100),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.73,0.02,0.91,0.05,1.19,0.1,1.61,0.2,2.34,0.3,3.06,0.4,3.94,0.5,
5.03,0.6,6.41,0.7,8.10,0.8,10.36,0.9,14.39,0.95,18.53,0.98,22.73,0.99,25.50,1,32.25)),
130:NEXT(2286$);

2286$   ASSIGN:      R.Vehicle Enters1915_1930.NumberOut=R.Vehicle
Enters1915_1930.NumberOut + 1:NEXT(127$);

;
;
;   Model statements for module: Decide 1792
;
127$   BRANCH,      1:
        If,Entity.CreateTime<46000,2289$,Yes:
        Else,2290$,Yes;
2289$   ASSIGN:      Decide 1792.NumberOut True=Decide 1792.NumberOut True + 1:NEXT(3$);

2290$   ASSIGN:      Decide 1792.NumberOut False=Decide 1792.NumberOut False + 1:NEXT(77$);

;
;
;   Model statements for module: Create 441
;

2291$   CREATE,      1,SecondstoBaseTime(46000),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.74,0.02,0.93,0.05,1.21,0.1,1.67,0.2,2.45,0.3,3.22,0.4,4.15,0.5,
5.32,0.6,6.81,0.7,8.64,0.8,11.04,0.9,15.35,0.95,19.90,0.98,24.25,0.99,27.11,1,33.94)),
122:NEXT(2292$);

```

```
2292$    ASSIGN:    R.Vehicle Enters1930_1945.NumberOut=R.Vehicle
Enters1930_1945.NumberOut + 1:NEXT(128$);
```

```
;
;
; Model statements for module: Decide 1793
;
```

```
128$    BRANCH,    1:
                If,Entity.CreateTime<46900,2295$,Yes:
                Else,2296$,Yes;
```

```
2295$    ASSIGN:    Decide 1793.NumberOut True=Decide 1793.NumberOut True + 1:NEXT(3$);
```

```
2296$    ASSIGN:    Decide 1793.NumberOut False=Decide 1793.NumberOut False + 1:NEXT(77$);
```

```
;
;
; Model statements for module: Create 442
;
```

```
2297$    CREATE,    1,SecondstoBaseTime(46900),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.75,0.02,0.94,0.05,1.22,0.1,1.69,0.2,2.48,0.3,3.26,0.4,4.21,0.5,
5.40,0.6,6.92,0.7,8.79,0.8,11.22,0.9,15.61,0.95,20.27,0.98,24.66,0.99,27.55,1,34.40)),
120:NEXT(2298$);
```

```
2298$    ASSIGN:    R.Vehicle Enters1945_2000.NumberOut=R.Vehicle
Enters1945_2000.NumberOut + 1:NEXT(129$);
```

```
;
;
; Model statements for module: Decide 1794
;
```

```
129$    BRANCH,    1:
                If,Entity.CreateTime<47800,2301$,Yes:
                Else,2302$,Yes;
```

```
2301$    ASSIGN:    Decide 1794.NumberOut True=Decide 1794.NumberOut True + 1:NEXT(3$);
```

```
2302$    ASSIGN:    Decide 1794.NumberOut False=Decide 1794.NumberOut False + 1:NEXT(77$);
```

```
;
;
; Model statements for module: Create 443
;
```

```
2303$    CREATE,    1,SecondstoBaseTime(47800),Vehicles:
```

```
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.76,0.02,0.95,0.05,1.23,0.1,1.72,0.2,2.54,0.3,3.35,0.4,4.33,0.5,
5.57,0.6,7.15,0.7,9.10,0.8,11.61,0.9,16.15,0.95,21.05,0.98,25.52,0.99,28.47,1,35.37)),
116:NEXT(2304$);
```


2304\$ ASSIGN: R.Vehicle Enters2000_2015.NumberOut=R.Vehicle
Enters2000_2015.NumberOut + 1:NEXT(130\$);

;
;
;
;
;

Model statements for module: Decide 1795

130\$ BRANCH, 1:

 If,Entity.CreateTime<48700,2307\$,Yes:

 Else,2308\$,Yes;

2307\$ ASSIGN: Decide 1795.NumberOut True=Decide 1795.NumberOut True + 1:NEXT(3\$);

2308\$ ASSIGN: Decide 1795.NumberOut False=Decide 1795.NumberOut False + 1:NEXT(77\$);

;
;
;
;
;

Model statements for module: Create 445

2309\$ CREATE, 1,SecondstoBaseTime(0),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.71,0.02,0.88,0.05,1.16,0.1,1.54,0.2,2.20,0.3,2.88,0.4,3.68,0.5,
4.68,0.6,5.93,0.7,7.46,0.8,9.56,0.9,13.25,0.95,16.91,0.98,20.93,0.99,23.57,1,30.23)),
20:NEXT(2310\$);

2310\$ ASSIGN: WarmupRunRight.NumberOut=WarmupRunRight.NumberOut +
1:NEXT(131\$);

;
;
;
;
;

Model statements for module: Decide 1797

131\$ BRANCH, 1:

 If,Entity.CreateTime<100,2313\$,Yes:

 Else,2314\$,Yes;

2313\$ ASSIGN: Decide 1797.NumberOut True=Decide 1797.NumberOut True + 1:NEXT(3\$);

2314\$ ASSIGN: Decide 1797.NumberOut False=Decide 1797.NumberOut False + 1:NEXT(77\$);

;
;
;
;
;

Model statements for module: Station 40

132\$ STATION, LStation6450;

2317\$ DELAY: 0.0,,VA:NEXT(154\$);

```

;
;
;   Model statements for module: Decide 746
;
154$   BRANCH,      1:
        If,Entity.Type==LCar,2318$,Yes:
        Else,2319$,Yes;
2318$   ASSIGN:      checkvehicletype.NumberOut True=checkvehicletype.NumberOut True +
1:NEXT(288$);

2319$   ASSIGN:      checkvehicletype.NumberOut False=checkvehicletype.NumberOut False +
1:NEXT(289$);

;
;
;   Model statements for module: Assign 220
;
288$   ASSIGN:      VTU(LAGV,LCar#)=NORM(83,4):NEXT(155$);

;
;
;   Model statements for module: Decide 747
;
155$   BRANCH,      1:
        If,SpeedatRStation5750 >= VTU(LAGV,LCar#),2320$,Yes:
        Else,2321$,Yes;
2320$   ASSIGN:      Decide 747.NumberOut True=Decide 747.NumberOut True + 1:NEXT(157$);
2321$   ASSIGN:      Decide 747.NumberOut False=Decide 747.NumberOut False + 1:NEXT(183$);

;
;
;   Model statements for module: Decide 774
;
157$   BRANCH,      1:
        If,SpeedatRStation5750>75,158$,Yes:
        If,SpeedatRStation5750 > 55 && SpeedatRStation5750 <= 75,171$,Yes:
        If,SpeedatRStation5750 >35 && SpeedatRStation5750 <=55,174$,Yes:
        If,SpeedatRStation5750 >10 && SpeedatRStation5750 <=35,177$,Yes:
        Else,180$,Yes;

;
;
;   Model statements for module: Decide 852
;
180$   BRANCH,      1:
        If,NZ(RL3,1) &&NZ(RL3,2) && NZ(RL2,28) == 0,2324$,Yes:
        Else,2325$,Yes;
2324$   ASSIGN:      CheckGapL6450Speed10.NumberOut
True=CheckGapL6450Speed10.NumberOut True + 1:NEXT(181$);

```

2325\$ ASSIGN: CheckGapL6450Speed10.NumberOut
False=CheckGapL6450Speed10.NumberOut False + 1:NEXT(182\$);

181\$ TRANSPORT: LAGV,RStation6700;

182\$ TRANSPORT: LAGV,LStation6700;

;
;
; Model statements for module: Decide 775
;

158\$ BRANCH, 1:
 If,
 NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5)
&&NZ(RL3,6) &&NZ(RL3,7) && NZ(RL2,28) && NZ(RL2,27) == 0,
 2326\$,Yes:
 Else,2327\$,Yes;

2326\$ ASSIGN: CheckGapL6450Speed75.NumberOut
True=CheckGapL6450Speed75.NumberOut True + 1:NEXT(159\$);

2327\$ ASSIGN: CheckGapL6450Speed75.NumberOut
False=CheckGapL6450Speed75.NumberOut False + 1:NEXT(160\$);

159\$ TRANSPORT: LAGV,RStation6700;

160\$ TRANSPORT: LAGV,LStation6700;

;
;
; Model statements for module: Decide 849
;

171\$ BRANCH, 1:
 If,NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5)
&&NZ(RL3,6) && NZ(RL2,28) == 0,2328\$,
 Yes:
 Else,2329\$,Yes;

2328\$ ASSIGN: CheckGapL6450Speed55_75.NumberOut
True=CheckGapL6450Speed55_75.NumberOut True + 1:NEXT(172\$);

2329\$ ASSIGN: CheckGapL6450Speed55_75.NumberOut
False=CheckGapL6450Speed55_75.NumberOut False + 1:NEXT(173\$);

172\$ TRANSPORT: LAGV,RStation6700;

173\$ TRANSPORT: LAGV,LStation6700;

;
;
; Model statements for module: Decide 850

```

;
174$    BRANCH,      1:
          If,NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5) &&
NZ(RL2,28) == 0,2330$,Yes:
          Else,2331$,Yes;
2330$    ASSIGN:      CheckGapL6450Speed35_55.NumberOut
True=CheckGapL6450Speed35_55.NumberOut True + 1:NEXT(175$);

2331$    ASSIGN:      CheckGapL6450Speed35_55.NumberOut
False=CheckGapL6450Speed35_55.NumberOut False + 1:NEXT(176$);

175$    TRANSPORT:    LAGV,RStation6700;

176$    TRANSPORT:    LAGV,LStation6700;

;
;
;    Model statements for module: Decide 851
;
177$    BRANCH,      1:
          If,NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) && NZ(RL2,28) == 0,2332$,Yes:
          Else,2333$,Yes;
2332$    ASSIGN:      CheckGapL6450Speed10_35.NumberOut
True=CheckGapL6450Speed10_35.NumberOut True + 1:NEXT(178$);

2333$    ASSIGN:      CheckGapL6450Speed10_35.NumberOut
False=CheckGapL6450Speed10_35.NumberOut False + 1:NEXT(179$);

178$    TRANSPORT:    LAGV,RStation6700;

179$    TRANSPORT:    LA GV,LStation6700;

;
;
;    Model statements for module: Decide 853
;
183$    BRANCH,      1:
          If,SpeedatRStation5750>75,184$,Yes:
          If,SpeedatRStation5750 > 55 && SpeedatRStation5750 <= 75,187$,Yes:
          If,SpeedatRStation5750 >35 && SpeedatRStation5750 <=55,190$,Yes:
          If,SpeedatRStation5750 >10 && SpeedatRStation5750 <=35,193$,Yes:
          Else,196$,Yes;

;
;
;    Model statements for module: Decide 858
;
196$    BRANCH,      1:
          If,NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) == 0,2336$,Yes:
          Else,2337$,Yes;

```

2336\$ ASSIGN: CheckGapL6450Speed10less.NumberOut
 True=CheckGapL6450Speed10less.NumberOut True + 1:NEXT(197\$);

2337\$ ASSIGN: CheckGapL6450Speed10less.NumberOut
 False=CheckGapL6450Speed10less.NumberOut False + 1:NEXT(198\$);

197\$ TRANSPORT: LAGV,RStation6700;

198\$ TRANSPORT: LAGV,LStation6700;

;
 ;

; Model statements for module: Decide 854

;

184\$ BRANCH, 1:
 If,
 NZ(RL2,22) &&NZ(RL2,23) &&NZ(RL2,24) &&NZ(RL2,25) &&NZ(RL2,26)
 &&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) && NZ(RL3,2) == 0,
 2338\$,Yes;
 Else,2339\$,Yes;

2338\$ ASSIGN: CheckGapL6450Speed75less.NumberOut
 True=CheckGapL6450Speed75less.NumberOut True + 1:NEXT(185\$);

2339\$ ASSIGN: CheckGapL6450Speed75less.NumberOut
 False=CheckGapL6450Speed75less.NumberOut False + 1:NEXT(186\$);

185\$ TRANSPORT: LAGV,RStation6700;

186\$ TRANSPORT: LAGV,LStation6700;

;
 ;

; Model statements for module: Decide 855

;

187\$ BRANCH, 1:
 If,NZ(RL2,24) &&NZ(RL2,25) &&NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) &&
 NZ(RL3,1) && NZ(RL3,2) == 0,
 2340\$,Yes;
 Else,2341\$,Yes;

2340\$ ASSIGN: CheckGapL6450Speed55_75less.NumberOut
 True=CheckGapL6450Speed55_75less.NumberOut True + 1
 :NEXT(188\$);

2341\$ ASSIGN: CheckGapL6450Speed55_75less.NumberOut
 False=CheckGapL6450Speed55_75less.NumberOut False + 1
 :NEXT(189\$);

188\$ TRANSPORT: LAGV,RStation6700;

189\$ TRANSPORT: LAGV,LStation6700;

```

;
;
;   Model statements for module: Decide 856
;
190$   BRANCH,      1:
        If,NZ(RL2,25) &&NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) ==
0,2342$,Yes:
        Else,2343$,Yes;
2342$   ASSIGN:      CheckGapL6450Speed35_55less.NumberOut
True=CheckGapL6450Speed35_55less.NumberOut True + 1
        :NEXT(191$);

2343$   ASSIGN:      CheckGapL6450Speed35_55less.NumberOut
False=CheckGapL6450Speed35_55less.NumberOut False + 1
        :NEXT(192$);

191$   TRANSPORT:    LAGV,RStation6700;

192$   TRANSPORT:    LAGV,LStation6700;

;
;
;   Model statements for module: Decide 857
;
193$   BRANCH,      1:
        If,NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) == 0,2344$,Yes:
        Else,2345$,Yes;
2344$   ASSIGN:      CheckGapL6450Speed10_35less.NumberOut
True=CheckGapL6450Speed10_35less.NumberOut True + 1
        :NEXT(194$);

2345$   ASSIGN:      CheckGapL6450Speed10_35less.NumberOut
False=CheckGapL6450Speed10_35less.NumberOut False + 1
        :NEXT(195$);

194$   TRANSPORT:    LAGV,RStation6700;

195$   TRANSPORT:    LAGV,LStation6700;

;
;
;   Model statements for module: Assign 221
;
289$   ASSIGN:      VTU(LAGVT,LTruck#)=NORM(83,4):NEXT(156$);

;
;
;   Model statements for module: Decide 748
;

```

```

156$    BRANCH,      1:
        If,SpeedatRStation5750 >= VTU(LAGVT,LTruck#),2346$,Yes:
        Else,2347$,Yes;
2346$    ASSIGN:      Decide 748.NumberOut True=Decide 748.NumberOut True + 1:NEXT(199$);
2347$    ASSIGN:      Decide 748.NumberOut False=Decide 748.NumberOut False + 1:NEXT(205$);

;
;
;   Model statements for module: Decide 860
;
199$    BRANCH,      1:
        If,SpeedatRStation5750>75,200$,Yes:
        If,SpeedatRStation5750 > 55 && SpeedatRStation5750 <= 75,201$,Yes:
        If,SpeedatRStation5750 >35 && SpeedatRStation5750 <=55,202$,Yes:
        If,SpeedatRStation5750 >10 && SpeedatRStation5750 <=35,203$,Yes:
        Else,204$,Yes;

;
;
;   Model statements for module: Decide 865
;
204$    BRANCH,      1:
        If,NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL2,28)==
0,2350$,Yes:
        Else,2351$,Yes;
2350$    ASSIGN:      CheckGapL6450Speed10t.NumberOut
True=CheckGapL6450Speed10t.NumberOut True + 1:NEXT(169$);

2351$    ASSIGN:      CheckGapL6450Speed10t.NumberOut
False=CheckGapL6450Speed10t.NumberOut False + 1:NEXT(170$);

169$    TRANSPORT:    LAGVT,RStation6700;

170$    TRANSPORT:    LAGVT,LStation6700;

;
;
;   Model statements for module: Decide 861
;
200$    BRANCH,      1:
        If,
        NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5)
&&NZ(RL3,6) &&NZ(RL3,7) &&NZ(RL3,8) &&NZ(RL3,9) && NZ(RL2,28) && NZ(RL2,27) == 0,
2352$,Yes:
        Else,2353$,Yes;
2352$    ASSIGN:      CheckGapL6450Speed75t.NumberOut
True=CheckGapL6450Speed75t.NumberOut True + 1:NEXT(161$);

2353$    ASSIGN:      CheckGapL6450Speed75t.NumberOut
False=CheckGapL6450Speed75t.NumberOut False + 1:NEXT(162$);

```

```

161$    TRANSPORT:  LAGVT,RStation6700;

162$    TRANSPORT:  LAGVT,LStation6700;

;
;
;   Model statements for module: Decide 862
;
201$    BRANCH,      1:
        If,
            NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5)
&&NZ(RL3,6) &&NZ(RL3,7) &&NZ(RL3,8) && NZ(RL2,28)== 0,
            2354$,Yes:
            Else,2355$,Yes;
2354$    ASSIGN:    CheckGapL6450Speed55_75t.NumberOut
True=CheckGapL6450Speed55_75t.NumberOut True + 1:NEXT(164$);

2355$    ASSIGN:    CheckGapL6450Speed55_75t.NumberOut
False=CheckGapL6450Speed55_75t.NumberOut False + 1:NEXT(163$);

164$    TRANSPORT:  LAGVT,RStation6700;

163$    TRANSPORT:  LAGVT,LStation6700;

;
;
;   Model statements for module: Decide 863
;
202$    BRANCH,      1:
        If,
            NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5)
&&NZ(RL3,6) &&NZ(RL3,7) &&NZ(RL2,28)== 0,
            2356$,Yes:
            Else,2357$,Yes;
2356$    ASSIGN:    CheckGapL6450Speed35_55t.NumberOut
True=CheckGapL6450Speed35_55t.NumberOut True + 1:NEXT(165$);

2357$    ASSIGN:    CheckGapL6450Speed35_55t.NumberOut
False=CheckGapL6450Speed35_55t.NumberOut False + 1:NEXT(166$);

165$    TRANSPORT:  LAGVT,RStation6700;

166$    TRANSPORT:  LAGVT,LStation6700;

;
;
;   Model statements for module: Decide 864
;
203$    BRANCH,      1:

```



```

                If,NZ(RL3,1) &&NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5)
&&NZ(RL3,6) &&NZ(RL2,28)== 0,2358$,Yes:
                Else,2359$,Yes;
2358$    ASSIGN:    CheckGapL6450Speed10_35t.NumberOut
True=CheckGapL6450Speed10_35t.NumberOut True + 1:NEXT(167$);

2359$    ASSIGN:    CheckGapL6450Speed10_35t.NumberOut
False=CheckGapL6450Speed10_35t.NumberOut False + 1:NEXT(168$);

167$    TRANSPORT:  LAGVT,RStation6700;

168$    TRANSPORT:  LAGVT,LStation6700;

;
;
;    Model statements for module: Decide 866
;
205$    BRANCH,      1:
                If,SpeedatRStation5750>75,206$,Yes:
                If,SpeedatRStation5750 > 55 && SpeedatRStation5750 <= 75,207$,Yes:
                If,SpeedatRStation5750 >35 && SpeedatRStation5750 <=55,208$,Yes:
                If,SpeedatRStation5750 >10 && SpeedatRStation5750 <=35,209$,Yes:
                Else,210$,Yes;

;
;
;    Model statements for module: Decide 871
;
210$    BRANCH,      1:
                If,NZ(RL2,25) &&NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) ==
0,2362$,Yes:
                Else,2363$,Yes;
2362$    ASSIGN:    CheckGapL6450Speed10lesst.NumberOut
True=CheckGapL6450Speed10lesst.NumberOut True + 1:NEXT(219$);

2363$    ASSIGN:    CheckGapL6450Speed10lesst.NumberOut
False=CheckGapL6450Speed10lesst.NumberOut False + 1:NEXT(220$);

219$    TRANSPORT:  LAGVT,RStation6700;

220$    TRANSPORT:  LAGVT,LStation6700;

;
;
;    Model statements for module: Decide 867
;
206$    BRANCH,      1:
                If,
                NZ(RL2,20) &&NZ(RL2,21) &&NZ(RL2,22) &&NZ(RL2,23) &&NZ(RL2,24)
&&NZ(RL2,25) &&NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) && NZ(RL3,2) == 0,
2364$,Yes:

```

```

                Else,2365$,Yes;
2364$    ASSIGN:    CheckGapL6450Speed75lesst.NumberOut
True=CheckGapL6450Speed75lesst.NumberOut True + 1:NEXT(211$);

2365$    ASSIGN:    CheckGapL6450Speed75lesst.NumberOut
False=CheckGapL6450Speed75lesst.NumberOut False + 1:NEXT(212$);

211$    TRANSPORT:  LAGVT,RStation6700;

212$    TRANSPORT:  LAGVT,LStation6700;

;
;
;    Model statements for module: Decide 868
;
207$    BRANCH,      1:
        If,
            NZ(RL2,21) &&NZ(RL2,22) &&NZ(RL2,23) &&NZ(RL2,24) &&NZ(RL2,25)
&&NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) == 0,
            2366$,Yes:
            Else,2367$,Yes;
2366$    ASSIGN:    CheckGapL6450Speed55_75lesst.NumberOut
True=CheckGapL6450Speed55_75lesst.NumberOut True + 1
:NEXT(213$);

2367$    ASSIGN:    CheckGapL6450Speed55_75lesst.NumberOut
False=CheckGapL6450Speed55_75lesst.NumberOut False + 1
:NEXT(214$);

213$    TRANSPORT:  LAGVT,RStation6700;

214$    TRANSPORT:  LAGVT,LStation6700;

;
;
;    Model statements for module: Decide 869
;
208$    BRANCH,      1:
        If,
            NZ(RL2,22) &&NZ(RL2,23) &&NZ(RL2,24) &&NZ(RL2,25) &&NZ(RL2,26)
&&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) == 0,
            2368$,Yes:
            Else,2369$,Yes;
2368$    ASSIGN:    CheckGapL6450Speed35_55lesst.NumberOut
True=CheckGapL6450Speed35_55lesst.NumberOut True + 1
:NEXT(215$);

2369$    ASSIGN:    CheckGapL6450Speed35_55lesst.NumberOut
False=CheckGapL6450Speed35_55lesst.NumberOut False + 1
:NEXT(216$);

```

```

215$    TRANSPORT:  LAGVT,RStation6700;

216$    TRANSPORT:  LAGVT,LStation6700;

;
;
;   Model statements for module: Decide 870
;
209$    BRANCH,      1:
                If,NZ(RL2,24) &&NZ(RL2,25) &&NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) &&
NZ(RL3,1) == 0,2370$,Yes;
                Else,2371$,Yes;
2370$    ASSIGN:     CheckGapL6450Speed10_35lesst.NumberOut
True=CheckGapL6450Speed10_35lesst.NumberOut True + 1
                :NEXT(217$);

2371$    ASSIGN:     CheckGapL6450Speed10_35lesst.NumberOut
False=CheckGapL6450Speed10_35lesst.NumberOut False + 1
                :NEXT(218$);

217$    TRANSPORT:  LAGVT,RStation6700;

218$    TRANSPORT:  LAGVT,LStation6700;

;
;
;   Model statements for module: Station 41
;

133$    STATION,     RStation6450;
2374$    DELAY:      0.0,,VA:NEXT(148$);

;
;
;   Model statements for module: Decide 719
;
148$    BRANCH,      1:
                If,Entity.Type==RCar,2375$,Yes;
                Else,2376$,Yes;
2375$    ASSIGN:     Decide1VehicleTypeR6450.NumberOut
True=Decide1VehicleTypeR6450.NumberOut True + 1:NEXT(290$);

2376$    ASSIGN:     Decide1VehicleTypeR6450.NumberOut
False=Decide1VehicleTypeR6450.NumberOut False + 1:NEXT(291$);

;
;
;   Model statements for module: Assign 222
;

```

```

290$    ASSIGN:    VTU(RAGV,RCar#)=NORM(73,4):NEXT(134$);

;
;
;   Model statements for module: Assign 35
;
134$    ASSIGN:    SpeedatRStation6450=VTU(RAGV,RCar#):NEXT(136$);

;
;
;   Model statements for module: Decide 88
;
136$    BRANCH,    1:
                If,Entity.Type==RCar,2377$,Yes:
                Else,2378$,Yes;
2377$    ASSIGN:    Decide2VehicleTypeR6450.NumberOut
True=Decide2VehicleTypeR6450.NumberOut True + 1:NEXT(135$);

2378$    ASSIGN:    Decide2VehicleTypeR6450.NumberOut
False=Decide2VehicleTypeR6450.NumberOut False + 1:NEXT(137$);

135$    TRANSPORT:  RAGV,RStation6700;

137$    TRANSPORT:  RAGVT,RStation6700;

;
;
;   Model statements for module: Assign 223
;
291$    ASSIGN:    VTU(RAGVT,Rtruck#)=NORM(73,4):NEXT(147$);

;
;
;   Model statements for module: Assign 66
;
147$    ASSIGN:    SpeedatRStation6450=VTU(RAGVT,RTruck#):NEXT(136$);

;
;
;   Model statements for module: Station 42
;
138$    STATION,    LStation6700;
2381$    DELAY:    0.0,,VA:NEXT(221$);

;
;

```

```

;   Model statements for module: Decide 872
;
221$   BRANCH,      1:
           If,Entity.Type==LCar,2382$,Yes:
           Else,2383$,Yes;
2382$   ASSIGN:      checkvehicletype6700.NumberOut True=checkvehicletype6700.NumberOut
True + 1:NEXT(222$);

2383$   ASSIGN:      checkvehicletype6700.NumberOut False=checkvehicletype6700.NumberOut
False + 1:NEXT(223$);

;
;
;   Model statements for module: Decide 873
;
222$   BRANCH,      1:
           If,SpeedatRStation6450 >= VTU(LAGV,LCar#),2384$,Yes:
           Else,2385$,Yes;
2384$   ASSIGN:      Decide 873.NumberOut True=Decide 873.NumberOut True + 1:NEXT(224$);

2385$   ASSIGN:      Decide 873.NumberOut False=Decide 873.NumberOut False + 1:NEXT(250$);

;
;
;   Model statements for module: Decide 875
;
224$   BRANCH,      1:
           If,SpeedatRStation6450>75,225$,Yes:
           If,SpeedatRStation6450 > 55 && SpeedatRStation6450 <= 75,238$,Yes:
           If,SpeedatRStation6450 >35 && SpeedatRStation6450 <=55,241$,Yes:
           If,SpeedatRStation6450 >10 && SpeedatRStation6450 <=35,244$,Yes:
           Else,247$,Yes;

;
;
;   Model statements for module: Decide 880
;
247$   BRANCH,      1:
           If,NZ(RL4,1) &&NZ(RL4,2) && NZ(RL3,10) == 0,2388$,Yes:
           Else,2389$,Yes;
2388$   ASSIGN:      CheckGapL6700Speed10.NumberOut
True=CheckGapL6700Speed10.NumberOut True + 1:NEXT(248$);

2389$   ASSIGN:      CheckGapL6700Speed10.NumberOut
False=CheckGapL6700Speed10.NumberOut False + 1:NEXT(249$);

248$   TRANSPORT:    LAGV,RStation6950;

249$   TRANSPORT:    LAGV,LStation6950;

```

```

;
;
;   Model statements for module: Decide 876
;
225$   BRANCH,      1:
        If,
            NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL4,5)
&&NZ(RL4,6) &&NZ(RL4,7) && NZ(RL3,9) && NZ(RL3,10) == 0,
            2390$,Yes:
            Else,2391$,Yes;
2390$   ASSIGN:      CheckGapL6700Speed75.NumberOut
True=CheckGapL6700Speed75.NumberOut True + 1:NEXT(226$);

2391$   ASSIGN:      CheckGapL6700Speed75.NumberOut
False=CheckGapL6700Speed75.NumberOut False + 1:NEXT(227$);

226$   TRANSPORT:   LAGV,RStation6950;

227$   TRANSPORT:   LAGV,LStation6950;

;
;
;   Model statements for module: Decide 877
;
238$   BRANCH,      1:
        If,NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL4,5)
&&NZ(RL4,6) && NZ(RL3,10) == 0,2392$,
        Yes:
        Else,2393$,Yes;
2392$   ASSIGN:      CheckGapL6700Speed55_75.NumberOut
True=CheckGapL6700Speed55_75.NumberOut True + 1:NEXT(239$);

2393$   ASSIGN:      CheckGapL6700Speed55_75.NumberOut
False=CheckGapL6700Speed55_75.NumberOut False + 1:NEXT(240$);

239$   TRANSPORT:   LAGV,RStation6950;

240$   TRANSPORT:   LAGV,LStation6950;

;
;
;   Model statements for module: Decide 878
;
241$   BRANCH,      1:
        If,NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL4,5) &&
NZ(RL3,10) == 0,2394$,Yes:
        Else,2395$,Yes;
2394$   ASSIGN:      CheckGapL6700Speed35_55.NumberOut
True=CheckGapL6700Speed35_55.NumberOut True + 1:NEXT(242$);

```

```

2395$    ASSIGN:    CheckGapL6700Speed35_55.NumberOut
False=CheckGapL6700Speed35_55.NumberOut False + 1:NEXT(243$);

242$    TRANSPORT:  LAGV,RStation6950;

243$    TRANSPORT:  LAGV,LStation6950;

;
;
;   Model statements for module: Decide 879
;
244$    BRANCH,      1:
                If,NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) && NZ(RL3,10) == 0,2396$,Yes:
                Else,2397$,Yes;
2396$    ASSIGN:    CheckGapL6700Speed10_35.NumberOut
True=CheckGapL6700Speed10_35.NumberOut True + 1:NEXT(245$);

2397$    ASSIGN:    CheckGapL6700Speed10_35.NumberOut
False=CheckGapL6700Speed10_35.NumberOut False + 1:NEXT(246$);

245$    TRANSPORT:  LAGV,RStation6950;

246$    TRANSPORT:  LAGV,LStation6950;

;
;
;   Model statements for module: Decide 881
;
250$    BRANCH,      1:
                If,SpeedatRStation6450>75,251$,Yes:
                If,SpeedatRStation6450 > 55 && SpeedatRStation6450 <= 75,254$,Yes:
                If,SpeedatRStation6450 >35 && SpeedatRStation6450 <=55,257$,Yes:
                If,SpeedatRStation6450 >10 && SpeedatRStation6450 <=35,260$,Yes:
                Else,263$,Yes;

;
;
;   Model statements for module: Decide 886
;
263$    BRANCH,      1:
                If,NZ(RL3,9) && NZ(RL3,10) && NZ(RL4,1) == 0,2400$,Yes:
                Else,2401$,Yes;
2400$    ASSIGN:    CheckGapL6700Speed10less.NumberOut
True=CheckGapL6700Speed10less.NumberOut True + 1:NEXT(264$);

2401$    ASSIGN:    CheckGapL6700Speed10less.NumberOut
False=CheckGapL6700Speed10less.NumberOut False + 1:NEXT(265$);

264$    TRANSPORT:  LAGV,RStation6950;

265$    TRANSPORT:  LAGV,LStation6950;

```

```

;
;
;   Model statements for module: Decide 882
;
251$   BRANCH,      1:
        If,
            NZ(RL3,4) &&NZ(RL3,5) &&NZ(RL3,6) &&NZ(RL3,7) &&NZ(RL3,8)
&&NZ(RL3,9) && NZ(RL3,10) && NZ(RL4,1) &&NZ(RL4,2) == 0,
            2402$,Yes:
            Else,2403$,Yes;
2402$   ASSIGN:      CheckGapL6700Speed75less.NumberOut
True=CheckGapL6700Speed75less.NumberOut True + 1:NEXT(252$);

2403$   ASSIGN:      CheckGapL6700Speed75less.NumberOut
False=CheckGapL6700Speed75less.NumberOut False + 1:NEXT(253$);

252$   TRANSPORT:    LAGV,RStation6950;

253$   TRANSPORT:    LAGV,LStation6950;

;
;
;   Model statements for module: Decide 883
;
254$   BRANCH,      1:
        If,NZ(RL3,5) &&NZ(RL3,6) &&NZ(RL3,7) &&NZ(RL3,8) &&NZ(RL3,9) &&
NZ(RL3,10) && NZ(RL4,1) == 0,2404$,
        Yes:
        Else,2405$,Yes;
2404$   ASSIGN:      CheckGapL6700Speed55_75less.NumberOut
True=CheckGapL6700Speed55_75less.NumberOut True + 1
:NEXT(255$);

2405$   ASSIGN:      CheckGapL6700Speed55_75less.NumberOut
False=CheckGapL6700Speed55_75less.NumberOut False + 1
:NEXT(256$);

255$   TRANSPORT:    LAGV,RStation6950;

256$   TRANSPORT:    LAGV,LStation6950;

;
;
;   Model statements for module: Decide 884
;
257$   BRANCH,      1:
        If,NZ(RL3,6) &&NZ(RL3,7) &&NZ(RL3,8) &&NZ(RL3,9) && NZ(RL3,10) &&
NZ(RL4,1) == 0,2406$,Yes:
        Else,2407$,Yes;

```


2406\$ ASSIGN: CheckGapL6700Speed35_55less.NumberOut
 True=CheckGapL6700Speed35_55less.NumberOut True + 1
 :NEXT(258\$);

2407\$ ASSIGN: CheckGapL6700Speed35_55less.NumberOut
 False=CheckGapL6700Speed35_55less.NumberOut False + 1
 :NEXT(259\$);

258\$ TRANSPORT: LAGV,RStation6950;

259\$ TRANSPORT: LAGV,LStation6950;

;
 ;
 ; Model statements for module: Decide 885
 ;
 260\$ BRANCH, 1:
 If,NZ(RL3,8) &&NZ(RL3,9) && NZ(RL3,10) && NZ(RL4,1) == 0,2408\$,Yes:
 Else,2409\$,Yes;

2408\$ ASSIGN: CheckGapL6700Speed10_35less.NumberOut
 True=CheckGapL6700Speed10_35less.NumberOut True + 1
 :NEXT(261\$);

2409\$ ASSIGN: CheckGapL6700Speed10_35less.NumberOut
 False=CheckGapL6700Speed10_35less.NumberOut False + 1
 :NEXT(262\$);

261\$ TRANSPORT: LAGV,RStation6950;

262\$ TRANSPORT: LAGV,LStation6950;

;
 ;
 ; Model statements for module: Decide 874
 ;
 223\$ BRANCH, 1:
 If,SpeedatRStation6450 >= VTU(LAGVT,LTruck#),2410\$,Yes:
 Else,2411\$,Yes;
 2410\$ ASSIGN: Decide 874.NumberOut True=Decide 874.NumberOut True + 1:NEXT(266\$);
 2411\$ ASSIGN: Decide 874.NumberOut False=Decide 874.NumberOut False + 1:NEXT(272\$);

;
 ;
 ; Model statements for module: Decide 887
 ;
 266\$ BRANCH, 1:
 If,SpeedatRStation6450>75,267\$,Yes:
 If,SpeedatRStation6450 > 55 && SpeedatRStation6450 <= 75,268\$,Yes:
 If,SpeedatRStation6450 >35 && SpeedatRStation6450 <=55,269\$,Yes:

```

If,SpeedatRStation6450 >10 && SpeedatRStation6450 <=35,270$,Yes:
Else,271$,Yes;

;
;
; Model statements for module: Decide 892
;
271$    BRANCH,    1:
        If,NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL3,10) ==
0,2414$,Yes:
        Else,2415$,Yes;
2414$    ASSIGN:    CheckGapL6700Speed10t.NumberOut
True=CheckGapL6700Speed10t.NumberOut True + 1:NEXT(236$);

2415$    ASSIGN:    CheckGapL6700Speed10t.NumberOut
False=CheckGapL6700Speed10t.NumberOut False + 1:NEXT(237$);

236$    TRANSPORT:  LAGVT,RStation6950;

237$    TRANSPORT:  LAGVT,LStation6950;

;
;
; Model statements for module: Decide 888
;
267$    BRANCH,    1:
        If,
        NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL4,5)
&&NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL4,8) &&NZ(RL4,9) && NZ(RL3,9) && NZ(RL3,10) == 0,
2416$,Yes:
        Else,2417$,Yes;
2416$    ASSIGN:    CheckGapL6700Speed75t.NumberOut
True=CheckGapL6700Speed75t.NumberOut True + 1:NEXT(228$);

2417$    ASSIGN:    CheckGapL6700Speed75t.NumberOut
False=CheckGapL6700Speed75t.NumberOut False + 1:NEXT(229$);

228$    TRANSPORT:  LAGVT,RStation6950;

229$    TRANSPORT:  LAGVT,LStation6950;

;
;
; Model statements for module: Decide 889
;
268$    BRANCH,    1:
        If,
        NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL4,5)
&&NZ(RL4,6) &&NZ(RL4,7) && NZ(RL4,8) && NZ(RL3,10) == 0,
2418$,Yes:
        Else,2419$,Yes;

```

2418\$ ASSIGN: CheckGapL6700Speed55_75t.NumberOut
 True=CheckGapL6700Speed55_75t.NumberOut True + 1:NEXT(231\$);

2419\$ ASSIGN: CheckGapL6700Speed55_75t.NumberOut
 False=CheckGapL6700Speed55_75t.NumberOut False + 1:NEXT(230\$);

231\$ TRANSPORT: LAGVT,RStation6950;

230\$ TRANSPORT: LAGVT,LStation6950;

;
 ;

; Model statements for module: Decide 890

;

269\$ BRANCH, 1:
 If,
 NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL4,5)
 &&NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL3,10) == 0,
 2420\$,Yes;
 Else,2421\$,Yes;

2420\$ ASSIGN: CheckGapL6700Speed35_55t.NumberOut
 True=CheckGapL6700Speed35_55t.NumberOut True + 1:NEXT(232\$);

2421\$ ASSIGN: CheckGapL6700Speed35_55t.NumberOut
 False=CheckGapL6700Speed35_55t.NumberOut False + 1:NEXT(233\$);

232\$ TRANSPORT: LAGVT,RStation6950;

233\$ TRANSPORT: LAGVT,LStation6950;

;
 ;

; Model statements for module: Decide 891

;

270\$ BRANCH, 1:
 If,NZ(RL4,1) &&NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL4,5)
 &&NZ(RL3,10) == 0,2422\$,Yes;
 Else,2423\$,Yes;

2422\$ ASSIGN: CheckGapL6700Speed10_35t.NumberOut
 True=CheckGapL6700Speed10_35t.NumberOut True + 1:NEXT(234\$);

2423\$ ASSIGN: CheckGapL6700Speed10_35t.NumberOut
 False=CheckGapL6700Speed10_35t.NumberOut False + 1:NEXT(235\$);

234\$ TRANSPORT: LAGVT,RStation6950;

235\$ TRANSPORT: LAGVT,LStation6950;

;
 ;

```

;   Model statements for module: Decide 893
;
272$   BRANCH,      1:
        If,SpeedatRStation6450>75,273$,Yes:
        If,SpeedatRStation6450 > 55 && SpeedatRStation6450 <= 75,274$,Yes:
        If,SpeedatRStation6450 >35 && SpeedatRStation6450 <=55,275$,Yes:
        If,SpeedatRStation6450 >10 && SpeedatRStation6450 <=35,276$,Yes:
        Else,277$,Yes;

;
;
;   Model statements for module: Decide 898
;
277$   BRANCH,      1:
        If,NZ(RL3,7) &&NZ(RL3,8) &&NZ(RL3,9) && NZ(RL3,10) && NZ(RL4,1) ==
0,2426$,Yes:
        Else,2427$,Yes;
2426$   ASSIGN:      CheckGapL6700Speed10lesst.NumberOut
True=CheckGapL6700Speed10lesst.NumberOut True + 1:NEXT(286$);

2427$   ASSIGN:      CheckGapL6700Speed10lesst.NumberOut
False=CheckGapL6700Speed10lesst.NumberOut False + 1:NEXT(287$);

286$   TRANSPORT:    LAGVT,RStation6950;

287$   TRANSPORT:    LAGVT,LStation6950;

;
;
;   Model statements for module: Decide 894
;
273$   BRANCH,      1:
        If,
        NZ(RL3,2) &&NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5) &&NZ(RL3,6)
&&NZ(RL3,7) &&NZ(RL3,8) &&NZ(RL3,9) && NZ(RL3,10) && NZ(RL4,1) &&NZ(RL4,2) == 0,
2428$,Yes:
        Else,2429$,Yes;
2428$   ASSIGN:      CheckGapL6700Speed75lesst.NumberOut
True=CheckGapL6700Speed75lesst.NumberOut True + 1:NEXT(278$);

2429$   ASSIGN:      CheckGapL6700Speed75lesst.NumberOut
False=CheckGapL6700Speed75lesst.NumberOut False + 1:NEXT(279$);

278$   TRANSPORT:    LAGVT,RStation6950;

279$   TRANSPORT:    LAGVT,LStation6950;

;
;
;   Model statements for module: Decide 895
;

```

```

274$    BRANCH,    1:
        If,
            NZ(RL3,3) &&NZ(RL3,4) &&NZ(RL3,5) &&NZ(RL3,6) &&NZ(RL3,7)
&&NZ(RL3,8) &&NZ(RL3,9) && NZ(RL3,10) && NZ(RL4,1) == 0,
            2430$,Yes:
            Else,2431$,Yes;
2430$    ASSIGN:    CheckGapL6700Speed55_75lesst.NumberOut
True=CheckGapL6700Speed55_75lesst.NumberOut True + 1
            :NEXT(280$);

2431$    ASSIGN:    CheckGapL6700Speed55_75lesst.NumberOut
False=CheckGapL6700Speed55_75lesst.NumberOut False + 1
            :NEXT(281$);

280$    TRANSPORT:  LAGVT,RStation6950;

281$    TRANSPORT:  LAGVT,LStation6950;

;
;
;    Model statements for module: Decide 896
;
275$    BRANCH,    1:
        If,
            NZ(RL3,4) &&NZ(RL3,5) &&NZ(RL3,6) &&NZ(RL3,7) &&NZ(RL3,8)
&&NZ(RL3,9) && NZ(RL3,10) && NZ(RL4,1) == 0,
            2432$,Yes:
            Else,2433$,Yes;
2432$    ASSIGN:    CheckGapL6700Speed35_55lesst.NumberOut
True=CheckGapL6700Speed35_55lesst.NumberOut True + 1
            :NEXT(282$);

2433$    ASSIGN:    CheckGapL6700Speed35_55lesst.NumberOut
False=CheckGapL6700Speed35_55lesst.NumberOut False + 1
            :NEXT(283$);

282$    TRANSPORT:  LAGVT,RStation6950;

283$    TRANSPORT:  LAGVT,LStation6950;

;
;
;    Model statements for module: Decide 897
;
276$    BRANCH,    1:
            IF,NZ(RL3,6) &&NZ(RL3,7) &&NZ(RL3,8) &&NZ(RL3,9) && NZ(RL3,10) &&
NZ(RL4,1) == 0,2434$,Yes:
            Else,2435$,Yes;
2434$    ASSIGN:    CheckGapL6700Speed10_35lesst.NumberOut
True=CheckGapL6700Speed10_35lesst.NumberOut True + 1
            :NEXT(284$);

```

```

2435$    ASSIGN:    CheckGapL6700Speed10_35lesst.NumberOut
False=CheckGapL6700Speed10_35lesst.NumberOut False + 1
          :NEXT(285$);

```

```

284$    TRANSPORT:  LAGVT,RStation6950;

```

```

285$    TRANSPORT:  LAGVT,LStation6950;

```

```

;
;
;    Model statements for module: Station 43
;

```

```

139$    STATION,    RStation6700;
2438$    DELAY:    0.0,,VA:NEXT(153$);

```

```

;
;
;    Model statements for module: Decide 721
;

```

```

153$    BRANCH,    1:
          If,Entity.Type==RCar,149$,Yes:
          If,Entity.Type==RTruck,150$,Yes:
          If,Entity.Type==LCar,151$,Yes:
          Else,152$,Yes;

```

```

;
;
;    Model statements for module: Assign 70
;
152$    ASSIGN:    SpeedatRStation6700=VTU(LAGVT,LTruck#):NEXT(141$);

```

```

;
;
;    Model statements for module: Decide 111
;

```

```

141$    BRANCH,    1:
          If,Entity.Type==RCar,2441$,Yes:
          Else,2442$,Yes;
2441$    ASSIGN:    Decide2VehicleTypeR6700.NumberOut
True=Decide2VehicleTypeR6700.NumberOut True + 1:NEXT(140$);

```

```

2442$    ASSIGN:    Decide2VehicleTypeR6700.NumberOut
False=Decide2VehicleTypeR6700.NumberOut False + 1:NEXT(143$);

```

```

140$    TRANSPORT:  RAGV,RStation6950;

```

```

;

```

```

;
;   Model statements for module: Decide 536
;
143$   BRANCH,      1:
        If,Entity.Type==RTruck,2443$,Yes:
        Else,2444$,Yes;
2443$   ASSIGN:      Decide3VehicleTypeR6700.NumberOut
True=Decide3VehicleTypeR6700.NumberOut True + 1:NEXT(142$);

2444$   ASSIGN:      Decide3VehicleTypeR6700.NumberOut
False=Decide3VehicleTypeR6700.NumberOut False + 1:NEXT(144$);

142$   TRANSPORT:   RAGVT,RStation6950;

;
;
;   Model statements for module: Decide 537
;
144$   BRANCH,      1:
        If,Entity.Type==LCar,2445$,Yes:
        Else,2446$,Yes;
2445$   ASSIGN:      Decide4VehicleTypeR6700.NumberOut
True=Decide4VehicleTypeR6700.NumberOut True + 1:NEXT(145$);

2446$   ASSIGN:      Decide4VehicleTypeR6700.NumberOut
False=Decide4VehicleTypeR6700.NumberOut False + 1:NEXT(146$);

145$   TRANSPORT:   LAGV,RStation6950;

146$   TRANSPORT:   LAGVT,RStation6950;

;
;
;   Model statements for module: Assign 67
;
149$   ASSIGN:      SpeedatRStation6700=VTU(RAGV,RCar#):NEXT(141$);

;
;
;   Model statements for module: Assign 68
;
150$   ASSIGN:      SpeedatRStation6700=VTU(RAGVT,RTruck#):NEXT(141$);

;
;
;   Model statements for module: Assign 69
;
151$   ASSIGN:      SpeedatRStation6700=VTU(LAGV,LCar#):NEXT(141$);

```

```

;
;
; Model statements for module: Station 45
;

292$    STATION,    RStation6950;
2449$    DELAY:    0.0,,VA:NEXT(341$);

;
;
; Model statements for module: Decide 723
;
341$    BRANCH,    1:
            If,Entity.Type==RCar,337$,Yes:
            If,Entity.Type==RTruck,338$,Yes:
            If,Entity.Type==LCar,339$,Yes:
            Else,340$,Yes;

;
;
; Model statements for module: Assign 78
;
340$    ASSIGN:    SpeedatRStation6950=VTU(LAGVT,LTruck#):NEXT(305$);

;
;
; Model statements for module: Decide 541
;
305$    BRANCH,    1:
            If,Entity.Type==RCar,2452$,Yes:
            Else,2453$,Yes;
2452$    ASSIGN:    Decide 541.NumberOut True=Decide 541.NumberOut True + 1:NEXT(304$);
2453$    ASSIGN:    Decide 541.NumberOut False=Decide 541.NumberOut False + 1:NEXT(307$);
304$    TRANSPORT:    RAGV,RStation7050;

;
;
; Model statements for module: Decide 542
;
307$    BRANCH,    1:
            If,Entity.Type==RTruck,2454$,Yes:
            Else,2455$,Yes;
2454$    ASSIGN:    Decide 542.NumberOut True=Decide 542.NumberOut True + 1:NEXT(306$);
2455$    ASSIGN:    Decide 542.NumberOut False=Decide 542.NumberOut False + 1:NEXT(308$);
306$    TRANSPORT:    RAGVT,RStation7050;

```



```

;
;
;   Model statements for module: Decide 543
;
308$   BRANCH,      1:
        If,Entity.Type==LCar,2456$,Yes:
        Else,2457$,Yes;
2456$   ASSIGN:      Decide 543.NumberOut True=Decide 543.NumberOut True + 1:NEXT(309$);
2457$   ASSIGN:      Decide 543.NumberOut False=Decide 543.NumberOut False + 1:NEXT(310$);

309$   TRANSPORT:    LAGV,RStation7050;
310$   TRANSPORT:    LAGVT,RStation7050;

;
;
;   Model statements for module: Assign 75
;
337$   ASSIGN:      SpeedatRStation6950=VTU(RAGV,RCar#):NEXT(305$);

;
;
;   Model statements for module: Assign 76
;
338$   ASSIGN:      SpeedatRStation6950=VTU(RAGVT,RTruck#):NEXT(305$);

;
;
;   Model statements for module: Assign 77
;
339$   ASSIGN:      SpeedatRStation6950=VTU(LAGV,LCar#):NEXT(305$);

;
;
;   Model statements for module: Station 47
;
293$   STATION,      RStation7050;
2460$   DELAY:      0.0,,VA:NEXT(336$);

;
;
;   Model statements for module: Decide 722
;
336$   BRANCH,      1:

```

```

        If,Entity.Type==RCar,332$,Yes:
        If,Entity.Type==RTruck,333$,Yes:
        If,Entity.Type==LCar,334$,Yes:
        Else,335$,Yes;

;
;
;   Model statements for module: Assign 74
;
335$   ASSIGN:   SpeedatRStation7050=VTU(LAGVT,LTruck#):NEXT(312$);

;
;
;   Model statements for module: Decide 544
;
312$   BRANCH,   1:
        If,Entity.Type==RCar,2463$,Yes:
        Else,2464$,Yes;
2463$   ASSIGN:   Decide 544.NumberOut True=Decide 544.NumberOut True + 1:NEXT(311$);
2464$   ASSIGN:   Decide 544.NumberOut False=Decide 544.NumberOut False + 1:NEXT(314$);
311$   TRANSPORT:  RAGV,RStation7150;

;
;
;   Model statements for module: Decide 545
;
314$   BRANCH,   1:
        If,Entity.Type==RTruck,2465$,Yes:
        Else,2466$,Yes;
2465$   ASSIGN:   Decide 545.NumberOut True=Decide 545.NumberOut True + 1:NEXT(313$);
2466$   ASSIGN:   Decide 545.NumberOut False=Decide 545.NumberOut False + 1:NEXT(315$);
313$   TRANSPORT:  RAGVT,RStation7150;

;
;
;   Model statements for module: Decide 546
;
315$   BRANCH,   1:
        If,Entity.Type==LCar,2467$,Yes:
        Else,2468$,Yes;
2467$   ASSIGN:   Decide 546.NumberOut True=Decide 546.NumberOut True + 1:NEXT(316$);
2468$   ASSIGN:   Decide 546.NumberOut False=Decide 546.NumberOut False + 1:NEXT(317$);
316$   TRANSPORT:  LAGV,RStation7150;

```

```

317$    TRANSPORT:   LAGVT,RStation7150;

;
;
;   Model statements for module: Assign 71
;
332$    ASSIGN:      SpeedatRStation7050=VTU(RAGV,RCar#):NEXT(312$);

;
;
;   Model statements for module: Assign 72
;
333$    ASSIGN:      SpeedatRStation7050=VTU(RAGVT,RTruck#):NEXT(312$);

;
;
;   Model statements for module: Assign 73
;
334$    ASSIGN:      SpeedatRStation7050=VTU(LAGV,LCar#):NEXT(312$);

;
;
;   Model statements for module: Station 49
;

294$    STATION,     RStation7150;
2471$   DELAY:       0.0,,VA:NEXT(346$);

;
;
;   Model statements for module: Decide 724
;
346$    BRANCH,      1:
                If,Entity.Type==RCar,342$,Yes:
                If,Entity.Type==RTruck,343$,Yes:
                If,Entity.Type==LCar,344$,Yes:
                Else,345$,Yes;

;
;
;   Model statements for module: Assign 82
;
345$    ASSIGN:      SpeedatRStation7150=VTU(LAGVT,LTruck#):NEXT(298$);

;
;
;   Model statements for module: Decide 538

```

```

;
298$    BRANCH,    1:
          If,Entity.Type==RCar,2474$,Yes:
          Else,2475$,Yes;
2474$    ASSIGN:    Decide 538.NumberOut True=Decide 538.NumberOut True + 1:NEXT(297$);
2475$    ASSIGN:    Decide 538.NumberOut False=Decide 538.NumberOut False + 1:NEXT(300$);
297$    TRANSPORT:  RAGV,RStation7250;

```

```

;
;
; Model statements for module: Decide 539
;
300$    BRANCH,    1:
          If,Entity.Type==RTruck,2476$,Yes:
          Else,2477$,Yes;
2476$    ASSIGN:    Decide 539.NumberOut True=Decide 539.NumberOut True + 1:NEXT(299$);
2477$    ASSIGN:    Decide 539.NumberOut False=Decide 539.NumberOut False + 1:NEXT(301$);
299$    TRANSPORT:  RAGVT,RStation7250;

```

```

;
;
; Model statements for module: Decide 540
;
301$    BRANCH,    1:
          If,Entity.Type==LCar,2478$,Yes:
          Else,2479$,Yes;
2478$    ASSIGN:    Decide 540.NumberOut True=Decide 540.NumberOut True + 1:NEXT(302$);
2479$    ASSIGN:    Decide 540.NumberOut False=Decide 540.NumberOut False + 1:NEXT(303$);
302$    TRANSPORT:  LAGV,RStation7250;
303$    TRANSPORT:  LAGVT,RStation7250;

```

```

;
;
; Model statements for module: Assign 79
;
342$    ASSIGN:    SpeedatRStation7150=VTU(RAGV,RCar#):NEXT(298$);

```

```

;
;
; Model statements for module: Assign 80
;
343$    ASSIGN:    SpeedatRStation7150=VTU(RAGVT,RTruck#):NEXT(298$);

```

```

;
;
; Model statements for module: Assign 81
;
344$    ASSIGN:    SpeedatRStation7150=VTU(LAGV,LCar#):NEXT(298$);

;
;
; Model statements for module: Station 51
;
295$    STATION,    RStation7250;
2482$    DELAY:    0.0,,VA:NEXT(351$);

;
;
; Model statements for module: Decide 725
;
351$    BRANCH,    1:
                If,Entity.Type==RCar,347$,Yes:
                If,Entity.Type==RTruck,348$,Yes:
                If,Entity.Type==LCar,349$,Yes:
                Else,350$,Yes;

;
;
; Model statements for module: Assign 86
;
350$    ASSIGN:    SpeedatRStation7250=VTU(LAGVT,LTruck#):NEXT(319$);

;
;
; Model statements for module: Decide 547
;
319$    BRANCH,    1:
                If,Entity.Type==RCar,2485$,Yes:
                Else,2486$,Yes;
2485$    ASSIGN:    Decide 547.NumberOut True=Decide 547.NumberOut True + 1:NEXT(318$);
2486$    ASSIGN:    Decide 547.NumberOut False=Decide 547.NumberOut False + 1:NEXT(321$);
318$    TRANSPORT:    RAGV,RStation7350;

;
;
; Model statements for module: Decide 548
;

```

```

321$    BRANCH,    1:
        If,Entity.Type==RTruck,2487$,Yes:
        Else,2488$,Yes;
2487$    ASSIGN:    Decide 548.NumberOut True=Decide 548.NumberOut True + 1:NEXT(320$);
2488$    ASSIGN:    Decide 548.NumberOut False=Decide 548.NumberOut False + 1:NEXT(322$);
320$    TRANSPORT:  RAGVT,RStation7350;

;
;
;   Model statements for module: Decide 549
;
322$    BRANCH,    1:
        If,Entity.Type==LCar,2489$,Yes:
        Else,2490$,Yes;
2489$    ASSIGN:    Decide 549.NumberOut True=Decide 549.NumberOut True + 1:NEXT(323$);
2490$    ASSIGN:    Decide 549.NumberOut False=Decide 549.NumberOut False + 1:NEXT(324$);
323$    TRANSPORT:  LAGV,RStation7350;
324$    TRANSPORT:  LAGVT,RStation7350;

;
;
;   Model statements for module: Assign 83
;
347$    ASSIGN:    SpeedatRStation7250=VTU(RAGV,RCar#):NEXT(319$);

;
;
;   Model statements for module: Assign 84
;
348$    ASSIGN:    SpeedatRStation7250=VTU(RAGVT,RTruck#):NEXT(319$);

;
;
;   Model statements for module: Assign 85
;
349$    ASSIGN:    SpeedatRStation7250=VTU(LAGV,LCar#):NEXT(319$);

;
;
;   Model statements for module: Station 53
;
296$    STATION,    RStation7350;

```

```

2493$    DELAY:    0.0,,VA:NEXT(356$);

;
;
;   Model statements for module: Decide 726
;
356$    BRANCH,    1:
                If,Entity.Type==RCar,352$,Yes:
                If,Entity.Type==RTruck,353$,Yes:
                If,Entity.Type==LCar,354$,Yes:
                Else,355$,Yes;

;
;
;   Model statements for module: Assign 90
;
355$    ASSIGN:    SpeedatRStation7350=VTU(LAGVT,LTruck#):NEXT(326$);

;
;
;   Model statements for module: Decide 550
;
326$    BRANCH,    1:
                If,Entity.Type==RCar,2496$,Yes:
                Else,2497$,Yes;
2496$    ASSIGN:    Decide 550.NumberOut True=Decide 550.NumberOut True + 1:NEXT(325$);
2497$    ASSIGN:    Decide 550.NumberOut False=Decide 550.NumberOut False + 1:NEXT(328$);
325$    TRANSPORT:  RAGV,RStation7450;

;
;
;   Model statements for module: Decide 551
;
328$    BRANCH,    1:
                If,Entity.Type==RTruck,2498$,Yes:
                Else,2499$,Yes;
2498$    ASSIGN:    Decide 551.NumberOut True=Decide 551.NumberOut True + 1:NEXT(327$);
2499$    ASSIGN:    Decide 551.NumberOut False=Decide 551.NumberOut False + 1:NEXT(329$);
327$    TRANSPORT:  RAGVT,RStation7450;

;
;
;   Model statements for module: Decide 552
;
329$    BRANCH,    1:

```

```

                If,Entity.Type==LCar,2500$,Yes:
                Else,2501$,Yes;
2500$    ASSIGN:    Decide 552.NumberOut True=Decide 552.NumberOut True + 1:NEXT(330$);
2501$    ASSIGN:    Decide 552.NumberOut False=Decide 552.NumberOut False + 1:NEXT(331$);
330$    TRANSPORT:  LAGV,RStation7450;
331$    TRANSPORT:  LAGVT,RStation7450;

;
;
; Model statements for module: Assign 87
;
352$    ASSIGN:    SpeedatRStation7350=VTU(RAGV,RCar#):NEXT(326$);

;
;
; Model statements for module: Assign 88
;
353$    ASSIGN:    SpeedatRStation7350=VTU(RAGVT,RTruck#):NEXT(326$);

;
;
; Model statements for module: Assign 89
;
354$    ASSIGN:    SpeedatRStation7350=VTU(LAGV,LCar#):NEXT(326$);

;
;
; Model statements for module: Station 87
;

357$    STATION,    LStation6950;
2504$    DELAY:    0.0,,VA:NEXT(358$);

;
;
; Model statements for module: Decide 899
;
358$    BRANCH,    1:
                If,Entity.Type==LCar,2505$,Yes:
                Else,2506$,Yes;
2505$    ASSIGN:    checkvehicletype6950.NumberOut True=checkvehicletype6950.NumberOut
True + 1:NEXT(359$);
2506$    ASSIGN:    checkvehicletype6950.NumberOut False=checkvehicletype6950.NumberOut
False + 1:NEXT(360$);

```



```

;
;
;   Model statements for module : Decide 900
;
359$   BRANCH,      1:
        If,SpeedatRStation6700 >= VTU(LAGV,LCar#),2507$,Yes:
        Else,2508$,Yes;
2507$   ASSIGN:      Decide 900.NumberOut True=Decide 900.NumberOut True + 1:NEXT(361$);

2508$   ASSIGN:      Decide 900.NumberOut False=Decide 900.NumberOut False + 1:NEXT(387$);

;
;
;   Model statements for module: Decide 902
;
361$   BRANCH,      1:
        If,SpeedatRStation6700>75,362$,Yes:
        If,SpeedatRStation6700 > 55 && SpeedatRStation6700 <= 75,375$,Yes:
        If,SpeedatRStation6700 >35 && SpeedatRStation6700 <=55,378$,Yes:
        If,SpeedatRStation6700 >10 && SpeedatRStation6700 <=35,381$,Yes:
        Else,384$,Yes;

;
;
;   Model statements for module: Decide 907
;
384$   BRANCH,      1:
        If,NZ(RL5,1) &&NZ(RL5,2) && NZ(RL4,10) == 0,2511$,Yes:
        Else,2512$,Yes;
2511$   ASSIGN:      CheckGapL6950Speed10.NumberOut
True=CheckGapL6950Speed10.NumberOut True + 1:NEXT(385$);

2512$   ASSIGN:      CheckGapL6950Speed10.NumberOut
False=CheckGapL6950Speed10.NumberOut False + 1:NEXT(386$);

385$   TRANSPORT:    LAGV,RStation7050;

386$   TRANSPORT:    LAGV,LStation7050;

;
;
;   Model statements for module: Decide 903
;
362$   BRANCH,      1:
        If,
        NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) &&NZ(RL5,4) &&NZ(RL6,1)
&&NZ(RL6,2) &&NZ(RL6,3) && NZ(RL4,9) && NZ(RL4,10) == 0,
        2513$,Yes:
        Else,2514$,Yes;

```

2513\$ ASSIGN: CheckGapL6950Speed75.NumberOut
 True=CheckGapL6950Speed75.NumberOut True + 1:NEXT(363\$);

2514\$ ASSIGN: CheckGapL6950Speed75.NumberOut
 False=CheckGapL6950Speed75.NumberOut False + 1:NEXT(364\$);

363\$ TRANSPORT: LAGV,RStation7050;

364\$ TRANSPORT: LAGV,LStation7050;

;
 ;

; Model statements for module: Decide 904

375\$ BRANCH, 1:
 If,NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) &&NZ(RL5,4) &&NZ(RL6,1)
 &&NZ(RL6,2) && NZ(RL4,10) == 0,2515\$,
 Yes:
 Else,2516\$,Yes;

2515\$ ASSIGN: CheckGapL6950Speed55_75.NumberOut
 True=CheckGapL6950Speed55_75.NumberOut True + 1:NEXT(376\$);

2516\$ ASSIGN: CheckGapL6950Speed55_75.NumberOut
 False=CheckGapL6950Speed55_75.NumberOut False + 1:NEXT(377\$);

376\$ TRANSPORT: LAGV,RStation7050;

377\$ TRANSPORT: LAGV,LStation7050;

;
 ;

; Model statements for module: Decide 905

378\$ BRANCH, 1:
 If,NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) &&NZ(RL5,4) &&NZ(RL6,1) &&
 NZ(RL4,10) == 0,2517\$,Yes:
 Else,2518\$,Yes;

2517\$ ASSIGN: CheckGapL6950Speed35_55.NumberOut
 True=CheckGapL6950Speed35_55.NumberOut True + 1:NEXT(379\$);

2518\$ ASSIGN: CheckGapL6950Speed35_55.NumberOut
 False=CheckGapL6950Speed35_55.NumberOut False + 1:NEXT(380\$);

379\$ TRANSPORT: LAGV,RStation7050;

380\$ TRANSPORT: LAGV,LStation7050;

;
 ;

; Model statements for module: Decide 906

```

;
381$    BRANCH,    1:
        If,NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) && NZ(RL4,10) == 0,2519$,Yes:
        Else,2520$,Yes;
2519$    ASSIGN:    CheckGapL6950Speed10_35.NumberOut
True=CheckGapL6950Speed10_35.NumberOut True + 1:NEXT(382$);

2520$    ASSIGN:    CheckGapL6950Speed10_35.NumberOut
False=CheckGapL6950Speed10_35.NumberOut False + 1:NEXT(383$);

382$    TRANSPORT:  LA GV,RStation7050;

383$    TRANSPORT:  LAGV,LStation7050;

;
;
;    Model statements for module: Decide 908
;
387$    BRANCH,    1:
        If,SpeedatRStation6700>75,388$,Yes:
        If,SpeedatRStation6700 > 55 && SpeedatRStation6700 <= 75,391$,Yes:
        If,SpeedatRStation6700 >35 && SpeedatRStation6700 <=55,394$,Yes:
        If,SpeedatRStation6700 >10 && SpeedatRStation6700 <=35,397$,Yes:
        Else,400$,Yes;

;
;
;    Model statements for module: Decide 913
;
400$    BRANCH,    1:
        If,NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1) == 0,2523$,Yes:
        Else,2524$,Yes;
2523$    ASSIGN:    CheckGapL6950Speed10less.NumberOut
True=CheckGapL6950Speed10less.NumberOut True + 1:NEXT(401$);

2524$    ASSIGN:    CheckGapL6950Speed10less.NumberOut
False=CheckGapL6950Speed10less.NumberOut False + 1:NEXT(402$);

401$    TRANSPORT:  LAGV,RStation7050;

402$    TRANSPORT:  LAGV,LStation7050;

;
;
;    Model statements for module: Decide 909
;
388$    BRANCH,    1:
        If,
        NZ(RL4,4) &&NZ(RL4,5) &&NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL4,8)
&&NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1) &&NZ(RL5,2) == 0,
        2525$,Yes:

```

```

                Else,2526$,Yes;
2525$    ASSIGN:    CheckGapL6950Speed75less.NumberOut
True=CheckGapL6950Speed75less.NumberOut True + 1:NEXT(389$);

2526$    ASSIGN:    CheckGapL6950Speed75less.NumberOut
False=CheckGapL6950Speed75less.NumberOut False + 1:NEXT(390$);

389$    TRANSPORT:  LAGV,RStation7050;

390$    TRANSPORT:  LAGV,LStation7050;

;
;
;    Model statements for module: Decide 910
;
391$    BRANCH,      1:
                If,NZ(RL4,5) &&NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL4,8) &&NZ(RL4,9) &&
NZ(RL4,10) && NZ(RL5,1) == 0,2527$,
                Yes:
                Else,2528$,Yes;
2527$    ASSIGN:    CheckGapL6950Speed55_75less.NumberOut
True=CheckGapL6950Speed55_75less.NumberOut True + 1
                :NEXT(392$);

2528$    ASSIGN:    CheckGapL6950Speed55_75less.NumberOut
False=CheckGapL6950Speed55_75less.NumberOut False + 1
                :NEXT(393$);

392$    TRANSPORT:  LAGV,RStation7050;

393$    TRANSPORT:  LAGV,LStation7050;

;
;
;    Model statements for module: Decide 911
;
394$    BRANCH,      1:
                If,NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL4,8) &&NZ(RL4,9) && NZ(RL4,10) &&
NZ(RL5,1) == 0,2529$,Yes;
                Else,2530$,Yes;
2529$    ASSIGN:    CheckGapL6950Speed35_55less.NumberOut
True=CheckGapL6950Speed35_55less.NumberOut True + 1
                :NEXT(395$);

2530$    ASSIGN:    CheckGapL6950Speed35_55less.NumberOut
False=CheckGapL6950Speed35_55less.NumberOut False + 1
                :NEXT(396$);

395$    TRANSPORT:  LAGV,RStation7050;

396$    TRANSPORT:  LAGV,LStation7050;

```

```

;
;
;   Model statements for module: Decide 912
;
397$   BRANCH,      1:
        If,NZ(RL4,8) &&NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1) == 0,2531$,Yes:
        Else,2532$,Yes;
2531$   ASSIGN:      CheckGapL6950Speed10_35less.NumberOut
True=CheckGapL6950Speed10_35less.NumberOut True + 1
        :NEXT(398$);

2532$   ASSIGN:      CheckGapL6950Speed10_35less.NumberOut
False=CheckGapL6950Speed10_35less.NumberOut False + 1
        :NEXT(399$);

398$   TRANSPORT:    LAGV,RStation7050;

399$   TRANSPORT:    LAGV,LStation7050;

;
;
;   Model statements for module: Decide 901
;
360$   BRANCH,      1:
        If,SpeedatRStation6700 >= VTU(LAGVT,LTruck#),2533$,Yes:
        Else,2534$,Yes;
2533$   ASSIGN:      Decide 901.NumberOut True=Decide 901.NumberOut True + 1:NEXT(403$);

2534$   ASSIGN:      Decide 901.NumberOut False=Decide 901.NumberOut False + 1:NEXT(409$);

;
;
;   Model statements for module: Decide 914
;
403$   BRANCH,      1:
        If,SpeedatRStation6700>75,404$,Yes:
        If,SpeedatRStation6700 > 55 && SpeedatRStation6700 <= 75,405$,Yes:
        If,SpeedatRStation6700 >35 && SpeedatRStation6700 <=55,406$,Yes:
        If,SpeedatRStation6700 >10 && SpeedatRStation6700 <=35,407$,Yes:
        Else,408$,Yes;

;
;
;   Model statements for module: Decide 919
;
408$   BRANCH,      1:
        If,NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) &&NZ(RL5,4) && NZ(RL4,10) ==
0,2537$,Yes:
        Else,2538$,Yes;

```

2537\$ ASSIGN: CheckGapL6950Speed10t.NumberOut
 True=CheckGapL6950Speed10t.NumberOut True + 1:NEXT(373\$);

2538\$ ASSIGN: CheckGapL6950Speed10t.NumberOut
 False=CheckGapL6950Speed10t.NumberOut False + 1:NEXT(374\$);

373\$ TRANSPORT: LAGVT,RStation7050;

374\$ TRANSPORT: LAGVT,LStation7050;

;
 ;

; Model statements for module: Decide 915

;

404\$ BRANCH, 1:
 If,
 NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) &&NZ(RL5,4) &&NZ(RL6,1)
 &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4)&& NZ(RL4,9) && NZ(RL4,10) == 0,
 2539\$,Yes:
 Else,2540\$,Yes;

2539\$ ASSIGN: CheckGapL6950Speed75t.NumberOut
 True=CheckGapL6950Speed75t.NumberOut True + 1:NEXT(365\$);

2540\$ ASSIGN: CheckGapL6950Speed75t.NumberOut
 False=CheckGapL6950Speed75t.NumberOut False + 1:NEXT(366\$);

365\$ TRANSPORT: LAGVT,RStation7050;

366\$ TRANSPORT: LAGVT,LStation7050;

;
 ;

; Model statements for module: Decide 916

;

405\$ BRANCH, 1:
 If,
 NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) &&NZ(RL5,4) &&NZ(RL6,1)
 &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4)&& NZ(RL4,10) == 0,
 2541\$,Yes:
 Else,2542\$,Yes;

2541\$ ASSIGN: CheckGapL6950Speed55_75t.NumberOut
 True=CheckGapL6950Speed55_75t.NumberOut True + 1:NEXT(368\$);

2542\$ ASSIGN: CheckGapL6950Speed55_75t.NumberOut
 False=CheckGapL6950Speed55_75t.NumberOut False + 1:NEXT(367\$);

368\$ TRANSPORT: LAGVT,RStation7050;

367\$ TRANSPORT: LAGVT,LStation7050;

```

;
;
;   Model statements for module: Decide 917
;
406$   BRANCH,      1:
        If,NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) &&NZ(RL5,4) &&NZ(RL6,1)
&&NZ(RL6,2) && NZ(RL4,10) == 0,2543$,
        Yes:
        Else,2544$,Yes;
2543$   ASSIGN:      CheckGapL6950Speed35_55t.NumberOut
True=CheckGapL6950Speed35_55t.NumberOut True + 1:NEXT(369$);

2544$   ASSIGN:      CheckGapL6950Speed35_55t.NumberOut
False=CheckGapL6950Speed35_55t.NumberOut False + 1:NEXT(370$);

369$   TRANSPORT:    LAGVT,RStation7050;

370$   TRANSPORT:    LAGVT,LStation7050;

;
;
;   Model statements for module: Decide 918
;
407$   BRANCH,      1:
        If,NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) &&NZ(RL5,4) &&NZ(RL6,1) &&
NZ(RL4,10) == 0,2545$,Yes:
        Else,2546$,Yes;
2545$   ASSIGN:      CheckGapL6950Speed10_35t.NumberOut
True=CheckGapL6950Speed10_35t.NumberOut True + 1:NEXT(371$);

2546$   ASSIGN:      CheckGapL6950Speed10_35t.NumberOut
False=CheckGapL6950Speed10_35t.NumberOut False + 1:NEXT(372$);

371$   TRANSPORT:    LAGVT,RStation7050;

372$   TRANSPORT:    LAGVT,LStation7050;

;
;
;   Model statements for module: Decide 920
;
409$   BRANCH,      1:
        If,SpeedatRStation6700>75,410$,Yes:
        If,SpeedatRStation6700 > 55 && SpeedatRStation6700 <= 75,411$,Yes:
        If,SpeedatRStation6700 >35 && SpeedatRStation6700 <=55,412$,Yes:
        If,SpeedatRStation6700 >10 && SpeedatRStation6700 <=35,413$,Yes:
        Else,414$,Yes;

;
;
;   Model statements for module: Decide 925

```

```

;
414$    BRANCH,    1:
            If,NZ(RL4,8) &&NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1) == 0,2549$,Yes:
            Else,2550$,Yes;
2549$    ASSIGN:    CheckGapL6950Speed10lesst.NumberOut
True=CheckGapL6950Speed10lesst.NumberOut True + 1:NEXT(423$);

2550$    ASSIGN:    CheckGapL6950Speed10lesst.NumberOut
False=CheckGapL6950Speed10lesst.NumberOut False + 1:NEXT(424$);

423$    TRANSPORT:  LAGVT,RStation7050;

424$    TRANSPORT:  LAGVT,LStation7050;

;
;
;    Model statements for module: Decide 921
;
410$    BRANCH,    1:
            If,
            NZ(RL4,2) &&NZ(RL4,3) &&NZ(RL4,4) &&NZ(RL4,5) &&NZ(RL4,6)
&&NZ(RL4,7) &&NZ(RL4,8) &&NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1) &&NZ(RL5,2) == 0,
            2551$,Yes:
            Else,2552$,Yes;
2551$    ASSIGN:    CheckGapL6950Speed75lesst.NumberOut
True=CheckGapL6950Speed75lesst.NumberOut True + 1:NEXT(415$);

2552$    ASSIGN:    CheckGapL6950Speed75lesst.NumberOut
False=CheckGapL6950Speed75lesst.NumberOut False + 1:NEXT(416$);

415$    TRANSPORT:  LAGVT,RStation7050;

416$    TRANSPORT:  LAGVT,LStation7050;

;
;
;    Model statements for module: Decide 922
;
411$    BRANCH,    1:
            If,
            NZ(RL4,4) &&NZ(RL4,5) &&NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL4,8)
&&NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1) &&NZ(RL5,2) == 0,
            2553$,Yes:
            Else,2554$,Yes;
2553$    ASSIGN:    CheckGapL6950Speed55_75lesst.NumberOut
True=CheckGapL6950Speed55_75lesst.NumberOut True + 1
:NEXT(417$);

2554$    ASSIGN:    CheckGapL6950Speed55_75lesst.NumberOut
False=CheckGapL6950Speed55_75lesst.NumberOut False + 1
:NEXT(418$);

```



```

417$    TRANSPORT:   LAGVT,RStation7050;

418$    TRANSPORT:   LAGVT,LStation7050;

;
;
;   Model statements for module: Decide 923
;
412$    BRANCH,      1:
                If,NZ(RL4,5) &&NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL4,8) &&NZ(RL4,9) &&
NZ(RL4,10) && NZ(RL5,1) == 0,2555$,
                Yes:
                Else,2556$,Yes;
2555$    ASSIGN:     CheckGapL6950Speed35_55lesst.NumberOut
True=CheckGapL6950Speed35_55lesst.NumberOut True + 1
                :NEXT(419$);

2556$    ASSIGN:     CheckGapL6950Speed35_55lesst.NumberOut
False=CheckGapL6950Speed35_55lesst.NumberOut False + 1
                :NEXT(420$);

419$    TRANSPORT:   LAGVT,RStation7050;

420$    TRANSPORT:   LAGVT,LStation7050;

;
;
;   Model statements for module: Decide 924
;
413$    BRANCH,      1:
                If,NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL4,8) &&NZ(RL4,9) && NZ(RL4,10) &&
NZ(RL5,1) == 0,2557$,Yes:
                Else,2558$,Yes;
2557$    ASSIGN:     CheckGapL6950Speed10_35lesst.NumberOut
True=CheckGapL6950Speed10_35lesst.NumberOut True + 1
                :NEXT(421$);

2558$    ASSIGN:     CheckGapL6950Speed10_35lesst.NumberOut
False=CheckGapL6950Speed10_35lesst.NumberOut False + 1
                :NEXT(422$);

421$    TRANSPORT:   LAGVT,RStation7050;

422$    TRANSPORT:   LAGVT,LStation7050;

;
;
;   Model statements for module: Station 88
;

```

```

425$      STATION,      LStation7050;
2561$      DELAY:      0.0,,VA:NEXT(426$);

;
;
;   Model statements for module: Decide 926
;
426$      BRANCH,      1:
                If,Entity.Type==LCar,2562$,Yes:
                Else,2563$,Yes;
2562$      ASSIGN:      checkvehicletype7050.NumberOut True=checkvehicletype7050.NumberOut
True + 1:NEXT(427$);

2563$      ASSIGN:      checkvehicletype7050.NumberOut False=checkvehicletype7050.NumberOut
False + 1:NEXT(428$);

;
;
;   Model statements for module: Decide 927
;
427$      BRANCH,      1:
                If,SpeedatRStation6950 >= VTU(LAGV,LCar#),2564$,Yes:
                Else,2565$,Yes;
2564$      ASSIGN:      Decide 927.NumberOut True=Decide 927.NumberOut True + 1:NEXT(429$);

2565$      ASSIGN:      Decide 927.NumberOut False=Decide 927.NumberOut False + 1:NEXT(455$);

;
;
;   Model statements for module: Decide 929
;
429$      BRANCH,      1:
                If,SpeedatRStation6950>75,430$,Yes:
                If,SpeedatRStation6950 > 55 && SpeedatRStation6950 <= 75,443$,Yes:
                If,SpeedatRStation6950 >35 && SpeedatRStation6950 <=55,446$,Yes:
                If,SpeedatRStation6950 >10 && SpeedatRStation6950 <=35,449$,Yes:
                Else,452$,Yes;

;
;
;   Model statements for module: Decide 934
;
452$      BRANCH,      1:
                If,NZ(RL6,1) &&NZ(RL6,2) && NZ(RL5,4) == 0,2568$,Yes:
                Else,2569$,Yes;
2568$      ASSIGN:      CheckGapL7050Speed10.NumberOut
True=CheckGapL7050Speed10.NumberOut True + 1:NEXT(453$);

```

```

2569$    ASSIGN:    CheckGapL7050Speed10.NumberOut
False=CheckGapL7050Speed10.NumberOut False + 1:NEXT(454$);

```

```

453$    TRANSPORT:  LAGV,RStation7150;

```

```

454$    TRANSPORT:  LAGV,LStation7150;

```

```

;
;
;   Model statements for module: Decide 930
;

```

```

430$    BRANCH,      1:
          If,
            NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4) &&NZ(RL7,1)
&&NZ(RL7,2) &&NZ(RL7,3) && NZ(RL5,3) && NZ(RL5,4) == 0,
            2570$,Yes;
          Else,2571$,Yes;

```

```

2570$    ASSIGN:    CheckGapL7050Speed75.NumberOut
True=CheckGapL7050Speed75.NumberOut True + 1:NEXT(431$);

```

```

2571$    ASSIGN:    CheckGapL7050Speed75.NumberOut
False=CheckGapL7050Speed75.NumberOut False + 1:NEXT(432$);

```

```

431$    TRANSPORT:  LAGV,RStation7150;

```

```

432$    TRANSPORT:  LAGV,LStation7150;

```

```

;
;
;   Model statements for module: Decide 931
;

```

```

443$    BRANCH,      1:
          If,NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4) &&NZ(RL7,1)
&&NZ(RL7,2) && NZ(RL5,4) == 0,2572$,
          Yes:
          Else,2573$,Yes;

```

```

2572$    ASSIGN:    CheckGapL7050Speed55_75.NumberOut
True=CheckGapL7050Speed55_75.NumberOut True + 1:NEXT(444$);

```

```

2573$    ASSIGN:    CheckGapL7050Speed55_75.NumberOut
False=CheckGapL7050Speed55_75.NumberOut False + 1:NEXT(445$);

```

```

444$    TRANSPORT:  LAGV,RStation7150;

```

```

445$    TRANSPORT:  LAGV,LStation7150;

```

```

;
;
;   Model statements for module: Decide 932
;

```

```

446$      BRANCH,      1:
              If,NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4) &&NZ(RL7,1)&&
NZ(RL5,4) == 0,2574$,Yes:
              Else,2575$,Yes;
2574$      ASSIGN:      CheckGapL7050Speed35_55.NumberOut
True=CheckGapL7050Speed35_55.NumberOut True + 1:NEXT(447$);

2575$      ASSIGN:      CheckGapL7050Speed35_55.NumberOut
False=CheckGapL7050Speed35_55.NumberOut False + 1:NEXT(448$);

447$      TRANSPORT:    LAGV,RStation7150;

448$      TRANSPORT:    LAGV,LStation7150;

;
;
;   Model statements for module: Decide 933
;
449$      BRANCH,      1:
              If,NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) && NZ(RL5,4) == 0,2576$,Yes:
              Else,2577$,Yes;
2576$      ASSIGN:      CheckGapL7050Speed10_35.NumberOut
True=CheckGapL7050Speed10_35.NumberOut True + 1:NEXT(450$);

2577$      ASSIGN:      CheckGapL7050Speed10_35.NumberOut
False=CheckGapL7050Speed10_35.NumberOut False + 1:NEXT(451$);

450$      TRANSPORT:    LAGV,RStation7150;

451$      TRANSPORT:    LAGV,LStation7150;

;
;
;   Model statements for module: Decide 935
;
455$      BRANCH,      1:
              If,SpeedatRStation6950>75,456$,Yes:
              If,SpeedatRStation6950 > 55 && SpeedatRStation6950 <= 75,459$,Yes:
              If,SpeedatRStation6950 >35 && SpeedatRStation6950 <=55,462$,Yes:
              If,SpeedatRStation6950 >10 && SpeedatRStation6950 <=35,465$,Yes:
              Else,468$,Yes;

;
;
;   Model statements for module: Decide 940
;
468$      BRANCH,      1:
              If,NZ(RL5,3)&&NZ(RL5,4) &&NZ(RL6,1)== 0,2580$,Yes:
              Else,2581$,Yes;
2580$      ASSIGN:      CheckGapL7050Speed10less.NumberOut
True=CheckGapL7050Speed10less.NumberOut True + 1:NEXT(469$);

```

```

2581$    ASSIGN:    CheckGapL7050Speed10less.NumberOut
False=CheckGapL7050Speed10less.NumberOut False + 1:NEXT(470$);

469$    TRANSPORT:  LAGV,RStation7150;

470$    TRANSPORT:  LAGV,LStation7150;

;
;
;   Model statements for module: Decide 936
;
456$    BRANCH,      1:
        If,
            NZ(RL4,8) &&NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1)
&&NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4) &&NZ(RL6,1)&&NZ(RL6,2) == 0,
2582$,Yes;
        Else,2583$,Yes;
2582$    ASSIGN:    CheckGapL7050Speed75less.NumberOut
True=CheckGapL7050Speed75less.NumberOut True + 1:NEXT(457$);

2583$    ASSIGN:    CheckGapL7050Speed75less.NumberOut
False=CheckGapL7050Speed75less.NumberOut False + 1:NEXT(458$);

457$    TRANSPORT:  LAGV,RStation7150;

458$    TRANSPORT:  LAGV,LStation7150;

;
;
;   Model statements for module: Decide 937
;
459$    BRANCH,      1:
        If,NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1)
&&NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4) &&NZ(RL6,1)== 0,2584$,Yes;
        Else,2585$,Yes;
2584$    ASSIGN:    CheckGapL7050Speed55_75less.NumberOut
True=CheckGapL7050Speed55_75less.NumberOut True + 1
:NEXT(460$);

2585$    ASSIGN:    CheckGapL7050Speed55_75less.NumberOut
False=CheckGapL7050Speed55_75less.NumberOut False + 1
:NEXT(461$);

460$    TRANSPORT:  LAGV,RStation7150;

461$    TRANSPORT:  LAGV,LStation7150;

;
;

```

```

;   Model statements for module: Decide 938
;
462$   BRANCH,      1:
        If,NZ(RL4,10) && NZ(RL5,1) &&NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4)
&&NZ(RL6,1)== 0,2586$,Yes:
        Else,2587$,Yes;
2586$   ASSIGN:      CheckGapL7050Speed35_55less.NumberOut
True=CheckGapL7050Speed35_55less.NumberOut True + 1
        :NEXT(463$);

2587$   ASSIGN:      CheckGapL7050Speed35_55less.NumberOut
False=CheckGapL7050Speed35_55less.NumberOut False + 1
        :NEXT(464$);

463$   TRANSPORT:    LAGV,RStation7150;

464$   TRANSPORT:    LAGV,LStation7150;

;
;
;   Model statements for module: Decide 939
;
465$   BRANCH,      1:
        If,NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4) &&NZ(RL6,1)== 0,2588$,Yes:
        Else,2589$,Yes;
2588$   ASSIGN:      CheckGapL7050Speed10_35less.NumberOut
True=CheckGapL7050Speed10_35less.NumberOut True + 1
        :NEXT(466$);

2589$   ASSIGN:      CheckGapL7050Speed10_35less.NumberOut
False=CheckGapL7050Speed10_35less.NumberOut False + 1
        :NEXT(467$);

466$   TRANSPORT:    LAGV,RStation7150;

467$   TRANSPORT:    LAGV,LStation7150;

;
;
;   Model statements for module: Decide 928
;
428$   BRANCH,      1:
        If,SpeedatRStation6950 >= VTU(LAGVT,LTruck#),2590$,Yes:
        Else,2591$,Yes;
2590$   ASSIGN:      Decide 928.NumberOut True=Decide 928.NumberOut True + 1:NEXT(471$);

2591$   ASSIGN:      Decide 928.NumberOut False=Decide 928.NumberOut False + 1:NEXT(477$);

;
;

```

```

;   Model statements for module: Decide 941
;
471$   BRANCH,      1:
        If,SpeedatRStation6950>75,472$,Yes:
        If,SpeedatRStation6950 > 55 && SpeedatRStation6950 <= 75,473$,Yes:
        If,SpeedatRStation6950 >35 && SpeedatRStation6950 <=55,474$,Yes:
        If,SpeedatRStation6950 >10 && SpeedatRStation6950 <=35,475$,Yes:
        Else,476$,Yes;

;
;
;   Model statements for module: Decide 946
;
476$   BRANCH,      1:
        If,NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4) && NZ(RL5,4) ==
0,2594$,Yes:
        Else,2595$,Yes;
2594$   ASSIGN:      CheckGapL7050Speed10t.NumberOut
True=CheckGapL7050Speed10t.NumberOut True + 1:NEXT(441$);

2595$   ASSIGN:      CheckGapL7050Speed10t.NumberOut
False=CheckGapL7050Speed10t.NumberOut False + 1:NEXT(442$);

441$   TRANSPORT:    LAGVT,RStation7150;

442$   TRANSPORT:    LAGVT,LStation7150;

;
;
;   Model statements for module: Decide 942
;
472$   BRANCH,      1:
        If,
        NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4) &&NZ(RL7,1)
&&NZ(RL7,2) &&NZ(RL7,3)&&NZ(RL7,4) && NZ(RL5,3) && NZ(RL5,4) == 0,
        2596$,Yes:
        Else,2597$,Yes;
2596$   ASSIGN:      CheckGapL7050Speed75t.NumberOut
True=CheckGapL7050Speed75t.NumberOut True + 1:NEXT(433$);

2597$   ASSIGN:      CheckGapL7050Speed75t.NumberOut
False=CheckGapL7050Speed75t.NumberOut False + 1:NEXT(434$);

433$   TRANSPORT:    LAGVT,RStation7150;

434$   TRANSPORT:    LAGVT,LStation7150;

;
;
;   Model statements for module: Decide 943
;

```

```

473$    BRANCH,    1:
        If,
        NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4) &&NZ(RL7,1)
&&NZ(RL7,2) &&NZ(RL7,3)&& NZ(RL5,4) == 0,
        2598$,Yes:
        Else,2599$,Yes;
2598$    ASSIGN:    CheckGapL7050Speed55_75t.NumberOut
True=CheckGapL7050Speed55_75t.NumberOut True + 1:NEXT(436$);

2599$    ASSIGN:    CheckGapL7050Speed55_75t.NumberOut
False=CheckGapL7050Speed55_75t.NumberOut False + 1:NEXT(435$);

436$    TRANSPORT:  LAGVT,RStation7150;

435$    TRANSPORT:  LAGVT,LStation7150;

;
;
; Model statements for module: Decide 944
;
474$    BRANCH,    1:
        If,NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4) &&NZ(RL7,1)
&&NZ(RL7,2) && NZ(RL5,4) == 0,2600$,
        Yes:
        Else,2601$,Yes;
2600$    ASSIGN:    CheckGapL7050Speed35_55t.NumberOut
True=CheckGapL7050Speed35_55t.NumberOut True + 1:NEXT(437$);

2601$    ASSIGN:    CheckGapL7050Speed35_55t.NumberOut
False=CheckGapL7050Speed35_55t.NumberOut False + 1:NEXT(438$);

437$    TRANSPORT:  LAGVT,RStation7150;

438$    TRANSPORT:  LAGVT,LStation7150;

;
;
; Model statements for module: Decide 945
;
475$    BRANCH,    1:
        If,NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) &&NZ(RL6,4) &&NZ(RL7,1) &&
NZ(RL5,4) == 0,2602$,Yes:
        Else,2603$,Yes;
2602$    ASSIGN:    CheckGapL7050Speed10_35t.NumberOut
True=CheckGapL7050Speed10_35t.NumberOut True + 1:NEXT(439$);

2603$    ASSIGN:    CheckGapL7050Speed10_35t.NumberOut
False=CheckGapL7050Speed10_35t.NumberOut False + 1:NEXT(440$);

439$    TRANSPORT:  LAGVT,RStation7150;

```



```

440$    TRANSPORT:   LAGVT,LStation7150;

;
;
;   Model statements for module: Decide 947
;
477$    BRANCH,      1:
          If,SpeedatRStation6950>75,478$,Yes:
          If,SpeedatRStation6950 > 55 && SpeedatRStation6950 <= 75,479$,Yes:
          If,SpeedatRStation6950 >35 && SpeedatRStation6950 <=55,480$,Yes:
          If,SpeedatRStation6950 >10 && SpeedatRStation6950 <=35,481$,Yes:
          Else,482$,Yes;

;
;
;   Model statements for module: Decide 952
;
482$    BRANCH,      1:
          If,NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4) &&NZ(RL6,1) == 0,2606$,Yes:
          Else,2607$,Yes;
2606$    ASSIGN:      CheckGapL7050Speed10lesst.NumberOut
True=CheckGapL7050Speed10lesst.NumberOut True + 1:NEXT(491$);

2607$    ASSIGN:      CheckGapL7050Speed10lesst.NumberOut
False=CheckGapL7050Speed10lesst.NumberOut False + 1:NEXT(492$);

491$    TRANSPORT:   LA GVT,RStation7150;

492$    TRANSPORT:   LAGVT,LStation7150;

;
;
;   Model statements for module: Decide 948
;
478$    BRANCH,      1:
          If,
          NZ(RL4,6) &&NZ(RL4,7) &&NZ(RL4,8) &&NZ(RL4,9) && NZ(RL4,10) &&
NZ(RL5,1) &&NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4) &&NZ(RL6,1)&&NZ(RL6,2) == 0,
          2608$,Yes:
          Else,2609$,Yes;
2608$    ASSIGN:      CheckGapL7050Speed75lesst.NumberOut
True=CheckGapL7050Speed75lesst.NumberOut True + 1:NEXT(483$);

2609$    ASSIGN:      CheckGapL7050Speed75lesst.NumberOut
False=CheckGapL7050Speed75lesst.NumberOut False + 1:NEXT(484$);

483$    TRANSPORT:   LAGVT,RStation7150;

484$    TRANSPORT:   LAGVT,LStation7150;

```

```

;
;
;   Model statements for module: Decide 949
;
479$   BRANCH,      1:
        If,
            NZ(RL4,7) &&NZ(RL4,8) &&NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1)
&&NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4) &&NZ(RL6,1) == 0,
            2610$,Yes:
            Else,2611$,Yes;
2610$   ASSIGN:      CheckGapL7050Speed55_75lesst.NumberOut
True=CheckGapL7050Speed55_75lesst.NumberOut True + 1
        :NEXT(485$);

2611$   ASSIGN:      CheckGapL7050Speed55_75lesst.NumberOut
False=CheckGapL7050Speed55_75lesst.NumberOut False + 1
        :NEXT(486$);

485$   TRANSPORT:    LAGVT,RStation7150;

486$   TRANSPORT:    LAGVT,LStation7150;

;
;
;   Model statements for module: Decide 950
;
480$   BRANCH,      1:
        If,NZ(RL4,9) && NZ(RL4,10) && NZ(RL5,1)
&&NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4) &&NZ(RL6,1) == 0,2612$,
        Yes:
        Else,2613$,Yes;
2612$   ASSIGN:      CheckGapL7050Speed35_55lesst.NumberOut
True=CheckGapL7050Speed35_55lesst.NumberOut True + 1
        :NEXT(487$);

2613$   ASSIGN:      CheckGapL7050Speed35_55lesst.NumberOut
False=CheckGapL7050Speed35_55lesst.NumberOut False + 1
        :NEXT(488$);

487$   TRANSPORT:    LAGVT,RStation7150;

488$   TRANSPORT:    LAGVT,LStation7150;

;
;
;   Model statements for module: Decide 951
;
481$   BRANCH,      1:
        If,NZ(RL4,10) && NZ(RL5,1) &&NZ(RL5,2)&&NZ(RL5,3)&&NZ(RL5,4)
&&NZ(RL6,1) == 0,2614$,Yes:
        Else,2615$,Yes;

```

```

2614$    ASSIGN:    CheckGapL7050Speed10_35lesst.NumberOut
True=CheckGapL7050Speed10_35lesst.NumberOut True + 1
          :NEXT(489$);

```

```

2615$    ASSIGN:    CheckGapL7050Speed10_35lesst.NumberOut
False=CheckGapL7050Speed10_35lesst.NumberOut False + 1
          :NEXT(490$);

```

```

489$     TRANSPORT:  LAGVT,RStation7150;

```

```

490$     TRANSPORT:  LAGVT,LStation7150;

```

```

;
;
;   Model statements for module: Station 89
;

```

```

493$     STATION,    LStation7150;
2618$     DELAY:     0.0,,VA:NEXT(494$);

```

```

;
;
;   Model statements for module: Decide 953
;

```

```

494$     BRANCH,     1:
                If,Entity.Type==LCar,2619$,Yes:
                Else,2620$,Yes;
2619$     ASSIGN:    checkvehicletype7150.NumberOut True=checkvehicletype7150.NumberOut
True + 1:NEXT(495$);

```

```

2620$     ASSIGN:    checkvehicletype7150.NumberOut False=checkvehicletype7150.NumberOut
False + 1:NEXT(496$);

```

```

;
;
;   Model statements for module: Decide 954
;

```

```

495$     BRANCH,     1:
                If,SpeedatRStation7050 >= VTU(LAGV,LCar#),2621$,Yes:
                Else,2622$,Yes;
2621$     ASSIGN:    Decide 954.NumberOut True=Decide 954.NumberOut True + 1:NEXT(497$);
2622$     ASSIGN:    Decide 954.NumberOut False=Decide 954.NumberOut False + 1:NEXT(523$);

```

```

;
;
;   Model statements for module: Decide 956
;
497$     BRANCH,     1:

```

```

If,SpeedatRStation7050>75,498$,Yes:
If,SpeedatRStation7050 > 55 && SpeedatRStation7050 <= 75,511$,Yes:
If,SpeedatRStation7050 >35 && SpeedatRStation7050 <=55,514$,Yes:
If,SpeedatRStation7050 >10 && SpeedatRStation7050 <=35,517$,Yes:
Else,520$,Yes;

;
;
; Model statements for module: Decide 961
;
520$    BRANCH,    1:
        If,NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL6,4) == 0,2625$,Yes:
        Else,2626$,Yes;
2625$    ASSIGN:    CheckGapL7150Speed10.NumberOut
True=CheckGapL7150Speed10.NumberOut True + 1:NEXT(521$);

2626$    ASSIGN:    CheckGapL7150Speed10.NumberOut
False=CheckGapL7150Speed10.NumberOut False + 1:NEXT(522$);

521$    TRANSPORT:  LAGV,RStation7250;

522$    TRANSPORT:  LAGV,LStation7250;

;
;
; Model statements for module: Decide 957
;
498$    BRANCH,    1:
        If,
        NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL7,4) &&NZ(RL8,1)
&&NZ(RL8,2) &&NZ(RL8,3) && NZ(RL6,3) && NZ(RL6,4) == 0,
        2627$,Yes:
        Else,2628$,Yes;
2627$    ASSIGN:    CheckGapL7150Speed75.NumberOut
True=CheckGapL7150Speed75.NumberOut True + 1:NEXT(499$);

2628$    ASSIGN:    CheckGapL7150Speed75.NumberOut
False=CheckGapL7150Speed75.NumberOut False + 1:NEXT(500$);

499$    TRANSPORT:  LAGV,RStation7250;

500$    TRANSPORT:  LAGV,LStation7250;

;
;
; Model statements for module: Decide 958
;
511$    BRANCH,    1:
        If,NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL7,4) &&NZ(RL8,1)
&&NZ(RL8,2) && NZ(RL6,4) == 0,2629$,
        Yes:

```

```

                Else,2630$,Yes;
2629$    ASSIGN:    CheckGapL7150Speed55_75.NumberOut
True=CheckGapL7150Speed55_75.NumberOut True + 1:NEXT(512$);

2630$    ASSIGN:    CheckGapL7150Speed55_75.NumberOut
False=CheckGapL7150Speed55_75.NumberOut False + 1:NEXT(513$);

512$    TRANSPORT:  LAGV,RStation7250;

513$    TRANSPORT:  LAGV,LStation7250;

;
;
;   Model statements for module: Decide 959
;
514$    BRANCH,      1:
                If,NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL7,4) &&NZ(RL8,1) &&
NZ(RL6,4) == 0,2631$,Yes;
                Else,2632$,Yes;
2631$    ASSIGN:    CheckGapL7150Speed35_55.NumberOut
True=CheckGapL7150Speed35_55.NumberOut True + 1:NEXT(515$);

2632$    ASSIGN:    CheckGapL7150Speed35_55.NumberOut
False=CheckGapL7150Speed35_55.NumberOut False + 1:NEXT(516$);

515$    TRANSPORT:  LAGV,RStation7250;

516$    TRANSPORT:  LAGV,LStation7250;

;
;
;   Model statements for module: Decide 960
;
517$    BRANCH,      1:
                If,NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL6,4) == 0,2633$,Yes;
                Else,2634$,Yes;
2633$    ASSIGN:    CheckGapL7150Speed10_35.NumberOut
True=CheckGapL7150Speed10_35.NumberOut True + 1:NEXT(518$);

2634$    ASSIGN:    CheckGapL7150Speed10_35.NumberOut
False=CheckGapL7150Speed10_35.NumberOut False + 1:NEXT(519$);

518$    TRANSPORT:  LAGV,RStation7250;

519$    TRANSPORT:  LAGV,LStation7250;

;
;
;   Model statements for module: Decide 962
;

```

```

523$      BRANCH,      1:
          If,SpeedatRStation7050>75,524$,Yes:
          If,SpeedatRStation7050 > 55 && SpeedatRStation7050 <= 75,527$,Yes:
          If,SpeedatRStation7050 >35 && SpeedatRStation7050 <=55,530$,Yes:
          If,SpeedatRStation7050 >10 && SpeedatRStation7050 <=35,533$,Yes:
          Else,536$,Yes;

;
;
;   Model statements for module: Decide 967
;
536$      BRANCH,      1:
          If,NZ(RL6,3)&&NZ(RL6,4) &&NZ(RL7,1) == 0,2637$,Yes:
          Else,2638$,Yes;
2637$      ASSIGN:      CheckGapL7150Speed10less.NumberOut
True=CheckGapL7150Speed10less.NumberOut True + 1:NEXT(537$);

2638$      ASSIGN:      CheckGapL7150Speed10less.NumberOut
False=CheckGapL7150Speed10less.NumberOut False + 1:NEXT(538$);

537$      TRANSPORT:    LAGV,RStation7250;

538$      TRANSPORT:    LAGV,LStation7250;

;
;
;   Model statements for module: Decide 963
;
524$      BRANCH,      1:
          If,
          NZ(RL5,2) &&NZ(RL5,3) && NZ(RL5,4) && NZ(RL6,1)
&&NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4) &&NZ(RL7,1)&&NZ(RL7,2) == 0,
          2639$,Yes:
          Else,2640$,Yes;
2639$      ASSIGN:      CheckGapL7150Speed75less.NumberOut
True=CheckGapL7150Speed75less.NumberOut True + 1:NEXT(525$);

2640$      ASSIGN:      CheckGapL7150Speed75less.NumberOut
False=CheckGapL7150Speed75less.NumberOut False + 1:NEXT(526$);

525$      TRANSPORT:    LAGV,RStation7250;

526$      TRANSPORT:    LAGV,LStation7250;

;
;
;   Model statements for module: Decide 964
;
527$      BRANCH,      1:
          If,NZ(RL5,3) && NZ(RL5,4) && NZ(RL6,1)
&&NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4) &&NZ(RL7,1) == 0,2641$,Yes:

```

```

                Else,2642$,Yes;
2641$    ASSIGN:    CheckGapL7150Speed55_75less.NumberOut
True=CheckGapL7150Speed55_75less.NumberOut True + 1
                :NEXT(528$);

2642$    ASSIGN:    CheckGapL7150Speed55_75less.NumberOut
False=CheckGapL7150Speed55_75less.NumberOut False + 1
                :NEXT(529$);

528$    TRANSPORT:  LAGV,RStation7250;

529$    TRANSPORT:  LAGV,LStation7250;

;
;
;    Model statements for module: Decide 965
;
530$    BRANCH,      1:
                If,NZ(RL5,4) && NZ(RL6,1) &&NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4)
&&NZ(RL7,1) == 0,2643$,Yes:
                Else,2644$,Yes;
2643$    ASSIGN:    CheckGapL7150Speed35_55less.NumberOut
True=CheckGapL7150Speed35_55less.NumberOut True + 1
                :NEXT(531$);

2644$    ASSIGN:    CheckGapL7150Speed35_55less.NumberOut
False=CheckGapL7150Speed35_55less.NumberOut False + 1
                :NEXT(532$);

531$    TRANSPORT:  LAGV,RStation7250;

532$    TRANSPORT:  LAGV,LStation7250;

;
;
;    Model statements for module: Decide 966
;
533$    BRANCH,      1:
                If,NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4) &&NZ(RL7,1) == 0,2645$,Yes:
                Else,2646$,Yes;
2645$    ASSIGN:    CheckGapL7150Speed10_35less.NumberOut
True=CheckGapL7150Speed10_35less.NumberOut True + 1
                :NEXT(534$);

2646$    ASSIGN:    CheckGapL7150Speed10_35less.NumberOut
False=CheckGapL7150Speed10_35less.NumberOut False + 1
                :NEXT(535$);

534$    TRANSPORT:  LAGV,RStation7250;

535$    TRANSPORT:  LAGV,LStation7250;

```

```

;
;
;   Model statements for module: Decide 955
;
496$   BRANCH,      1:
        If,SpeedatRStation7050 >= VTU(LAGVT,LTruck#),2647$,Yes:
        Else,2648$,Yes;
2647$   ASSIGN:      Decide 955.NumberOut True=Decide 955.NumberOut True + 1:NEXT(539$);

2648$   ASSIGN:      Decide 955.NumberOut False=Decide 955.NumberOut False + 1:NEXT(545$);

;
;
;   Model statements for module: Decide 968
;
539$   BRANCH,      1:
        If,SpeedatRStation7050>75,540$,Yes:
        If,SpeedatRStation7050 > 55 && SpeedatRStation7050 <= 75,541$,Yes:
        If,SpeedatRStation7050 >35 && SpeedatRStation7050 <=55,542$,Yes:
        If,SpeedatRStation7050 >10 && SpeedatRStation7050 <=35,543$,Yes:
        Else,544$,Yes;

;
;
;   Model statements for module: Decide 973
;
544$   BRANCH,      1:
        If,NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL7,4) && NZ(RL6,4) ==
0,2651$,Yes:
        Else,2652$,Yes;
2651$   ASSIGN:      CheckGapL7150Speed10t.NumberOut
True=CheckGapL7150Speed10t.NumberOut True + 1:NEXT(509$);

2652$   ASSIGN:      CheckGapL7150Speed10t.NumberOut
False=CheckGapL7150Speed10t.NumberOut False + 1:NEXT(510$);

509$   TRANSPORT:    LAGVT,RStation7250;

510$   TRANSPORT:    LAGVT,LStation7250;

;
;
;   Model statements for module: Decide 969
;
540$   BRANCH,      1:
        If,
        NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL7,4) &&NZ(RL8,1)
&&NZ(RL8,2) &&NZ(RL8,3) && NZ(RL8,4) &&NZ(RL6,3) && NZ(RL6,4) == 0,
2653$,Yes:

```



```

                Else,2654$,Yes;
2653$    ASSIGN:    CheckGapL7150Speed75t.NumberOut
True=CheckGapL7150Speed75t.NumberOut True + 1:NEXT(501$);

2654$    ASSIGN:    CheckGapL7150Speed75t.NumberOut
False=CheckGapL7150Speed75t.NumberOut False + 1:NEXT(502$);

501$    TRANSPORT:  LAGVT,RStation7250;

502$    TRANSPORT:  LAGVT,LStation7250;

;
;
;    Model statements for module: Decide 970
;
541$    BRANCH,      1:
                If,
                    NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL7,4) &&NZ(RL8,1)
&&NZ(RL8,2) &&NZ(RL8,3) && NZ(RL6,4) == 0,
                    2655$,Yes:
                    Else,2656$,Yes;
2655$    ASSIGN:    CheckGapL7150Speed55_75t.NumberOut
True=CheckGapL7150Speed55_75t.NumberOut True + 1:NEXT(504$);

2656$    ASSIGN:    CheckGapL7150Speed55_75t.NumberOut
False=CheckGapL7150Speed55_75t.NumberOut False + 1:NEXT(503$);

504$    TRANSPORT:  LAGVT,RStation7250;

503$    TRANSPORT:  LAGVT,LStation7250;

;
;
;    Model statements for module: Decide 971
;
542$    BRANCH,      1:
                If,NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL7,4) &&NZ(RL8,1)
&&NZ(RL8,2) && NZ(RL6,4) == 0,2657$,
                    Yes:
                    Else,2658$,Yes;
2657$    ASSIGN:    CheckGapL7150Speed35_55t.NumberOut
True=CheckGapL7150Speed35_55t.NumberOut True + 1:NEXT(505$);

2658$    ASSIGN:    CheckGapL7150Speed35_55t.NumberOut
False=CheckGapL7150Speed35_55t.NumberOut False + 1:NEXT(506$);

505$    TRANSPORT:  LAGVT,RStation7250;

506$    TRANSPORT:  LAGVT,LStation7250;

```

```

;
;
;   Model statements for module: Decide 972
;
543$   BRANCH,      1:
        If,NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) &&NZ(RL7,4) &&NZ(RL8,1) &&
NZ(RL6,4) == 0,2659$,Yes:
        Else,2660$,Yes;
2659$   ASSIGN:      CheckGapL7150Speed10_35t.NumberOut
True=CheckGapL7150Speed10_35t.NumberOut True + 1:NEXT(507$);

2660$   ASSIGN:      CheckGapL7150Speed10_35t.NumberOut
False=CheckGapL7150Speed10_35t.NumberOut False + 1:NEXT(508$);

507$   TRANSPORT:    LAGVT,RStation7250;

508$   TRANSPORT:    LAGVT,LStation7250;

;
;
;   Model statements for module: Decide 974
;
545$   BRANCH,      1:
        If,SpeedatRStation7050>75,546$,Yes:
        If,SpeedatRStation7050 > 55 && SpeedatRStation7050 <= 75,547$,Yes:
        If,SpeedatRStation7050 >35 && SpeedatRStation7050 <=55,548$,Yes:
        If,SpeedatRStation7050 >10 && SpeedatRStation7050 <=35,549$,Yes:
        Else,550$,Yes;

;
;
;   Model statements for module : Decide 979
;
550$   BRANCH,      1:
        If,NZ(RL6,1) &&NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4) &&NZ(RL7,1) ==
0,2663$,Yes:
        Else,2664$,Yes;
2663$   ASSIGN:      CheckGapL7150Speed10lesst.NumberOut
True=CheckGapL7150Speed10lesst.NumberOut True + 1:NEXT(559$);

2664$   ASSIGN:      CheckGapL7150Speed10lesst.NumberOut
False=CheckGapL7150Speed10lesst.NumberOut False + 1:NEXT(560$);

559$   TRANSPORT:    LAGVT,RStation7250;

560$   TRANSPORT:    LAGVT,LStation7250;

;
;
;   Model statements for module: Decide 975
;

```

```

546$    BRANCH,    1:
        If,
            NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) && NZ(RL5,4) && NZ(RL6,1)
&&NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4) &&NZ(RL7,1)&&NZ(RL7,2) == 0,
            2665$,Yes:
            Else,2666$,Yes;
2665$    ASSIGN:    CheckGapL7150Speed75lesst.NumberOut
True=CheckGapL7150Speed75lesst.NumberOut True + 1:NEXT(551$);

2666$    ASSIGN:    CheckGapL7150Speed75lesst.NumberOut
False=CheckGapL7150Speed75lesst.NumberOut False + 1:NEXT(552$);

551$    TRANSPORT:  LAGVT,RStation7250;

552$    TRANSPORT:  LAGVT,LStation7250;

;
;
;    Model statements for module: Decide 976
;
547$    BRANCH,    1:
        If,
            NZ(RL5,1) &&NZ(RL5,2) &&NZ(RL5,3) && NZ(RL5,4) && NZ(RL6,1)
&&NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4) &&NZ(RL7,1) == 0,
            2667$,Yes:
            Else,2668$,Yes;
2667$    ASSIGN:    CheckGapL7150Speed55_75lesst.NumberOut
True=CheckGapL7150Speed55_75lesst.NumberOut True + 1
:NEXT(553$);

2668$    ASSIGN:    CheckGapL7150Speed55_75lesst.NumberOut
False=CheckGapL7150Speed55_75lesst.NumberOut False + 1
:NEXT(554$);

553$    TRANSPORT:  LAGVT,RStation7250;

554$    TRANSPORT:  LAGVT,LStation7250;

;
;
;    Model statements for module: Decide 977
;
548$    BRANCH,    1:
        If,NZ(RL5,3) && NZ(RL5,4) && NZ(RL6,1)
&&NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4) &&NZ(RL7,1) == 0,2669$,Yes:
            Else,2670$,Yes;
2669$    ASSIGN:    CheckGapL7150Speed35_55lesst.NumberOut
True=CheckGapL7150Speed35_55lesst.NumberOut True + 1
:NEXT(555$);

```

```

2670$    ASSIGN:    CheckGapL7150Speed35_55lesst.NumberOut
False=CheckGapL7150Speed35_55lesst.NumberOut False + 1
          :NEXT(556$);

555$     TRANSPORT:  LAGVT,RStation7250;

556$     TRANSPORT:  LAGVT,LStation7250;

;
;
;   Model statements for module: Decide 978
;
549$     BRANCH,     1:
          If,NZ(RL5,4) && NZ(RL6,1) &&NZ(RL6,2)&&NZ(RL6,3)&&NZ(RL6,4)
&&NZ(RL7,1) == 0,2671$,Yes:
          Else,2672$,Yes;
2671$    ASSIGN:    CheckGapL7150Speed10_35lesst.NumberOut
True=CheckGapL7150Speed10_35lesst.NumberOut True + 1
          :NEXT(557$);

2672$    ASSIGN:    CheckGapL7150Speed10_35lesst.NumberOut
False=CheckGapL7150Speed10_35lesst.NumberOut False + 1
          :NEXT(558$);

557$     TRANSPORT:  LAGVT,RStation7250;

558$     TRANSPORT:  LAGVT,LStation7250;

;
;
;   Model statements for module: Station 90
;

561$     STATION,    LStation7250;
2675$    DELAY:      0.0,,VA:NEXT(562$);

;
;
;   Model statements for module: Decide 980
;
562$     BRANCH,     1:
          If,Entity.Type==LCar,2676$,Yes:
          Else,2677$,Yes;
2676$    ASSIGN:    checkvehicletype7250.NumberOut True=checkvehicletype7250.NumberOut
True + 1:NEXT(563$);

2677$    ASSIGN:    checkvehicletype7250.NumberOut False=checkvehicletype7250.NumberOut
False + 1:NEXT(564$);

```

```

;
;
;   Model statements for module: Decide 981
;
563$   BRANCH,      1:
        If,SpeedatRStation7150 >= VTU(LAGV,LCar#),2678$,Yes:
        Else,2679$,Yes;
2678$   ASSIGN:      Decide 981.NumberOut True=Decide 981.NumberOut True + 1:NEXT(565$);

2679$   ASSIGN:      Decide 981.NumberOut False=Decide 981.NumberOut False + 1:NEXT(591$);

;
;
;   Model statements for module: Decide 983
;
565$   BRANCH,      1:
        If,SpeedatRStation7150>75,566$,Yes:
        If,SpeedatRStation7150 > 55 && SpeedatRStation7150 <= 75,579$,Yes:
        If,SpeedatRStation7150 >35 && SpeedatRStation7150<=55,582$,Yes:
        If,SpeedatRStation7150 >10 && SpeedatRStation7150 <=35,585$,Yes:
        Else,588$,Yes;

;
;
;   Model statements for module: Decide 988
;
588$   BRANCH,      1:
        If,NZ(RL8,1) &&NZ(RL8,2) && NZ(RL7,4) == 0,2682$,Yes:
        Else,2683$,Yes;
2682$   ASSIGN:      CheckGapL7250Speed10.NumberOut
True=CheckGapL7250Speed10.NumberOut True + 1:NEXT(589$);

2683$   ASSIGN:      CheckGapL7250Speed10.NumberOut
False=CheckGapL7250Speed10.NumberOut False + 1:NEXT(590$);

589$   TRANSPORT:    LAGV,RStation7350;

590$   TRANSPORT:    LAGV,LStation7350;

;
;
;   Model statements for module: Decide 984
;
566$   BRANCH,      1:
        If,
        NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) &&NZ(RL8,4) &&NZ(RL9,1)
&&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL7,3) && NZ(RL7,4) == 0,
        2684$,Yes:
        Else,2685$,Yes;
2684$   ASSIGN:      CheckGapL7250Speed75.NumberOut
True=CheckGapL7250Speed75.NumberOut True + 1:NEXT(567$);

```

```

2685$    ASSIGN:    CheckGapL7250Speed75.NumberOut
False=CheckGapL7250Speed75.NumberOut False + 1:NEXT(568$);

567$     TRANSPORT:  LAGV,RStation7350;

568$     TRANSPORT:  LAGV,LStation7350;

;
;
;   Model statements for module: Decide 985
;
579$     BRANCH,     1:
                If,NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) &&NZ(RL8,4) &&NZ(RL9,1)
&&NZ(RL9,2) && NZ(RL7,4) == 0,2686$,
                Yes:
                Else,2687$,Yes;
2686$     ASSIGN:    CheckGapL7250Speed55_75.NumberOut
True=CheckGapL7250Speed55_75.NumberOut True + 1:NEXT(580$);

2687$     ASSIGN:    CheckGapL7250Speed55_75.NumberOut
False=CheckGapL7250Speed55_75.NumberOut False + 1:NEXT(581$);

580$     TRANSPORT:  LAGV,RStation7350;

581$     TRANSPORT:  LAGV,LStation7350;

;
;
;   Model statements for module: Decide 986
;
582$     BRANCH,     1:
                If,NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) &&NZ(RL8,4) &&NZ(RL9,1) &&
NZ(RL7,4) == 0,2688$,Yes:
                Else,2689$,Yes;
2688$     ASSIGN:    CheckGapL7250Speed35_55.NumberOut
True=CheckGapL7250Speed35_55.NumberOut True + 1:NEXT(583$);

2689$     ASSIGN:    CheckGapL7250Speed35_55.NumberOut
False=CheckGapL7250Speed35_55.NumberOut False + 1:NEXT(584$);

583$     TRANSPORT:  LAGV,RStation7350;

584$     TRANSPORT:  LAGV,LStation7350;

;
;
;   Model statements for module: Decide 987
;
585$     BRANCH,     1:

```

```

                If,NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) && NZ(RL7,4) == 0,2690$,Yes:
                Else,2691$,Yes;
2690$    ASSIGN:    CheckGapL7250Speed10_35.NumberOut
True=CheckGapL7250Speed10_35.NumberOut True + 1:NEXT(586$);

2691$    ASSIGN:    CheckGapL7250Speed10_35.NumberOut
False=CheckGapL7250Speed10_35.NumberOut False + 1:NEXT(587$);

586$    TRANSPORT:  LAGV,RStation7350;

587$    TRANSPORT:  LAGV,LStation7350;

;
;
;   Model statements for module: Decide 989
;
591$    BRANCH,      1:
                If,SpeedatRStation7150>75,592$,Yes:
                If,SpeedatRStation7150 > 55 && SpeedatRStation7150 <= 75,595$,Yes:
                If,SpeedatRStation7150 >35 && SpeedatRStation7150 <=55,598$,Yes:
                If,SpeedatRStation7150 >10 && SpeedatRStation7150 <=35,601$,Yes:
                Else,604$,Yes;

;
;
;   Model statements for module: Decide 994
;
604$    BRANCH,      1:
                If,NZ(RL7,3)&&NZ(RL7,4) &&NZ(RL8,1)== 0,2694$,Yes:
                Else,2695$,Yes;
2694$    ASSIGN:    CheckGapL7250Speed10less.NumberOut
True=CheckGapL7250Speed10less.NumberOut True + 1:NEXT(605$);

2695$    ASSIGN:    CheckGapL7250Speed10less.NumberOut
False=CheckGapL7250Speed10less.NumberOut False + 1:NEXT(606$);

605$    TRANSPORT:  LAGV,RStation7350;

606$    TRANSPORT:  LAGV,LStation7350;

;
;
;   Model statements for module: Decide 990
;
592$    BRANCH,      1:
                If,
                NZ(RL6,2) &&NZ(RL6,3) && NZ(RL6,4) && NZ(RL7,1)
&&NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4) &&NZ(RL8,1)&&NZ(RL8,2) == 0,
                2696$,Yes:
                Else,2697$,Yes;

```

2696\$ ASSIGN: CheckGapL7250Speed75less.NumberOut
True=CheckGapL7250Speed75less.NumberOut True + 1:NEXT(593\$);

2697\$ ASSIGN: CheckGapL7250Speed75less.NumberOut
False=CheckGapL7250Speed75less.NumberOut False + 1:NEXT(594\$);

593\$ TRANSPORT: LAGV,RStation7350;

594\$ TRANSPORT: LAGV,LStation7350;

;
;

; Model statements for module: Decide 991

;

595\$ BRANCH, 1:
 If,NZ(RL6,3) && NZ(RL6,4) && NZ(RL7,1)
&&NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4) &&NZ(RL8,1)== 0,2698\$,Yes:
 Else,2699\$,Yes;

2698\$ ASSIGN: CheckGapL7250Speed55_75less.NumberOut
True=CheckGapL7250Speed55_75less.NumberOut True + 1
 :NEXT(596\$);

2699\$ ASSIGN: CheckGapL7250Speed55_75less.NumberOut
False=CheckGapL7250Speed55_75less.NumberOut False + 1
 :NEXT(597\$);

596\$ TRANSPORT: LAGV,RStation7350;

597\$ TRANSPORT: LAGV,LStation7350;

;
;

; Model statements for module: Decide 992

;

598\$ BRANCH, 1:
 If,NZ(RL6,4) && NZ(RL7,1) &&NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4)
&&NZ(RL8,1)== 0,2700\$,Yes:
 Else,2701\$,Yes;

2700\$ ASSIGN: CheckGapL7250Speed35_55less.NumberOut
True=CheckGapL7250Speed35_55less.NumberOut True + 1
 :NEXT(599\$);

2701\$ ASSIGN: CheckGapL7250Speed35_55less.NumberOut
False=CheckGapL7250Speed35_55less.NumberOut False + 1
 :NEXT(600\$);

599\$ TRANSPORT: LAGV,RStation7350;

600\$ TRANSPORT: LAGV,LStation7350;


```

;
;
;   Model statements for module: Decide 993
;
601$   BRANCH,      1:
        If,NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4) &&NZ(RL8,1)== 0,2702$,Yes:
        Else,2703$,Yes;
2702$   ASSIGN:      CheckGapL7250Speed10_35less.NumberOut
True=CheckGapL7250Speed10_35less.NumberOut True + 1
        :NEXT(602$);

2703$   ASSIGN:      CheckGapL7250Speed10_35less.NumberOut
False=CheckGapL7250Speed10_35less.NumberOut False + 1
        :NEXT(603$);

602$   TRANSPORT:    LAGV,RStation7350;

603$   TRANSPORT:    LAGV,LStation7350;

;
;
;   Model statements for module: Decide 982
;
564$   BRANCH,      1:
        If,SpeedatRStation7150 >= VTU(LAGVT,LTruck#),2704$,Yes:
        Else,2705$,Yes;
2704$   ASSIGN:      Decide 982.NumberOut True=Decide 982.NumberOut True + 1:NEXT(607$);
2705$   ASSIGN:      Decide 982.NumberOut False=Decide 982.NumberOut False + 1:NEXT(613$);

;
;
;   Model statements for module: Decide 995
;
607$   BRANCH,      1:
        If,SpeedatRStation7150>75,608$,Yes:
        If,SpeedatRStation7150 > 55 && SpeedatRStation7150 <= 75,609$,Yes:
        If,SpeedatRStation7150 >35 && SpeedatRStation7150 <=55,610$,Yes:
        If,SpeedatRStation7150 >10 && SpeedatRStation7150 <=35,611$,Yes:
        Else,612$,Yes;

;
;
;   Model statements for module: Decide 1000
;
612$   BRANCH,      1:
        If,NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) &&NZ(RL8,4) &&NZ(RL7,4) ==
0,2708$,Yes:
        Else,2709$,Yes;
2708$   ASSIGN:      CheckGapL7250Speed10t.NumberOut
True=CheckGapL7250Speed10t.NumberOut True + 1:NEXT(577$);

```

```

2709$    ASSIGN:    CheckGapL7250Speed10t.NumberOut
False=CheckGapL7250Speed10t.NumberOut False + 1:NEXT(578$);

577$     TRANSPORT:  LAGVT,RStation7350;

578$     TRANSPORT:  LAGVT,LStation7350;

;
;
;   Model statements for module: Decide 996
;
608$     BRANCH,     1:
           If,
               NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) &&NZ(RL8,4) &&NZ(RL9,1)
&&NZ(RL9,2) &&NZ(RL9,3) && NZ(RL9,4) &&NZ(RL7,3) && NZ(RL7,4) == 0,
               2710$,Yes:
               Else,2711$,Yes;
2710$     ASSIGN:    CheckGapL7250Speed75t.NumberOut
True=CheckGapL7250Speed75t.NumberOut True + 1:NEXT(569$);

2711$     ASSIGN:    CheckGapL7250Speed75t.NumberOut
False=CheckGapL7250Speed75t.NumberOut False + 1:NEXT(570$);

569$     TRANSPORT:  LAGVT,RStation7350;

570$     TRANSPORT:  LAGVT,LStation7350;

;
;
;   Model statements for module: Decide 997
;
609$     BRANCH,     1:
           If,
               NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) &&NZ(RL8,4) &&NZ(RL9,1)
&&NZ(RL9,2) &&NZ(RL9,3) && NZ(RL7,4) == 0,
               2712$,Yes:
               Else,2713$,Yes;
2712$     ASSIGN:    CheckGapL7250Speed55_75t.NumberOut
True=CheckGapL7250Speed55_75t.NumberOut True + 1:NEXT(572$);

2713$     ASSIGN:    CheckGapL7250Speed55_75t.NumberOut
False=CheckGapL7250Speed55_75t.NumberOut False + 1:NEXT(571$);

572$     TRANSPORT:  LAGVT,RStation7350;

571$     TRANSPORT:  LAGVT,LStation7350;

;
;

```

```

;   Model statements for module: Decide 998
;
610$   BRANCH,      1:
           If,NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) &&NZ(RL8,4) &&NZ(RL9,1)
&&NZ(RL9,2) &&NZ(RL7,4) == 0,2714$,
           Yes:
           Else,2715$,Yes;
2714$   ASSIGN:      CheckGapL7250Speed35_55t.NumberOut
True=CheckGapL7250Speed35_55t.NumberOut True + 1:NEXT(573$);

2715$   ASSIGN:      CheckGapL7250Speed35_55t.NumberOut
False=CheckGapL7250Speed35_55t.NumberOut False + 1:NEXT(574$);

573$   TRANSPORT:    LAGVT,RStation7350;

574$   TRANSPORT:    LAGVT,LStation7350;

;
;
;   Model statements for module: Decide 999
;
611$   BRANCH,      1:
           If,NZ(RL8,1) &&NZ(RL8,2) &&NZ(RL8,3) &&NZ(RL8,4) &&NZ(RL9,1)
&&NZ(RL7,4) == 0,2716$,Yes:
           Else,2717$,Yes;
2716$   ASSIGN:      CheckGapL7250Speed10_35t.NumberOut
True=CheckGapL7250Speed10_35t.NumberOut True + 1:NEXT(575$);

2717$   ASSIGN:      CheckGapL7250Speed10_35t.NumberOut
False=CheckGapL7250Speed10_35t.NumberOut False + 1:NEXT(576$);

575$   TRANSPORT:    LAGVT,RStation7350;

576$   TRANSPORT:    LAGVT,LStation7350;

;
;
;   Model statements for module: Decide 1001
;
613$   BRANCH,      1:
           If,SpeedatRStation7150>75,614$,Yes:
           If,SpeedatRStation7150 > 55 && SpeedatRStation7150<= 75,615$,Yes:
           If,SpeedatRStation7150 >35 && SpeedatRStation7150 <=55,616$,Yes:
           If,SpeedatRStation7150 >10 && SpeedatRStation7150 <=35,617$,Yes:
           Else,618$,Yes;

;
;
;   Model statements for module: Decide 1006
;
618$   BRANCH,      1:

```

```

                                If,NZ(RL7,1) &&NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4) &&NZ(RL8,1)==
0,2720$,Yes:
                                Else,2721$,Yes;
2720$    ASSIGN:    CheckGapL7250Speed10lesst.NumberOut
True=CheckGapL7250Speed10lesst.NumberOut True + 1:NEXT(627$);

2721$    ASSIGN:    CheckGapL7250Speed10lesst.NumberOut
False=CheckGapL7250Speed10lesst.NumberOut False + 1:NEXT(628$);

627$    TRANSPORT:  LAGVT,RStation7350;

628$    TRANSPORT:  LAGVT,LStation7350;

;
;
;   Model statements for module: Decide 1002
;
614$    BRANCH,      1:
                                If,
                                NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) && NZ(RL6,4) && NZ(RL7,1)
&&NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4) &&NZ(RL8,1)&&NZ(RL8,2) == 0,
                                2722$,Yes:
                                Else,2723$,Yes;
2722$    ASSIGN:    CheckGapL7250Speed75lesst.NumberOut
True=CheckGapL7250Speed75lesst.NumberOut True + 1:NEXT(619$);

2723$    ASSIGN:    CheckGapL7250Speed75lesst.NumberOut
False=CheckGapL7250Speed75lesst.NumberOut False + 1:NEXT(620$);

619$    TRANSPORT:  LAGVT,RStation7350;

620$    TRANSPORT:  LAGVT,LStation7350;

;
;
;   Model statements for module: Decide 1003
;
615$    BRANCH,      1:
                                If,
                                NZ(RL6,1) &&NZ(RL6,2) &&NZ(RL6,3) && NZ(RL6,4) && NZ(RL7,1)
&&NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4) &&NZ(RL8,1)== 0,
                                2724$,Yes:
                                Else,2725$,Yes;
2724$    ASSIGN:    CheckGapL7250Speed55_75lesst.NumberOut
True=CheckGapL7250Speed55_75lesst.NumberOut True + 1
                                :NEXT(621$);

2725$    ASSIGN:    CheckGapL7250Speed55_75lesst.NumberOut
False=CheckGapL7250Speed55_75lesst.NumberOut False + 1
                                :NEXT(622$);

```

```

621$    TRANSPORT:  LAGVT,RStation7350;

622$    TRANSPORT:  LAGVT,LStation7350;

;
;
;   Model statements for module: Decide 1004
;
616$    BRANCH,      1:
          If,NZ(RL6,3) && NZ(RL6,4) && NZ(RL7,1)
&&NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4) &&NZ(RL8,1)== 0,2726$,Yes:
          Else,2727$,Yes;
2726$    ASSIGN:      CheckGapL7250Speed35_55lesst.NumberOut
True=CheckGapL7250Speed35_55lesst.NumberOut True + 1
          :NEXT(623$);

2727$    ASSIGN:      CheckGapL7250Speed35_55lesst.NumberOut
False=CheckGapL7250Speed35_55lesst.NumberOut False + 1
          :NEXT(624$);

623$    TRANSPORT:  LAGVT,RStation7350;

624$    TRANSPORT:  LAGVT,LStation7350;

;
;
;   Model statements for module: Decide 1005
;
617$    BRANCH,      1:
          If,NZ(RL6,4) && NZ(RL7,1) &&NZ(RL7,2)&&NZ(RL7,3)&&NZ(RL7,4)
&&NZ(RL8,1)== 0,2728$,Yes:
          Else,2729$,Yes;
2728$    ASSIGN:      CheckGapL7250Speed10_35lesst.NumberOut
True=CheckGapL7250Speed10_35lesst.NumberOut True + 1
          :NEXT(625$);

2729$    ASSIGN:      CheckGapL7250Speed10_35lesst.NumberOut
False=CheckGapL7250Speed10_35lesst.NumberOut False + 1
          :NEXT(626$);

625$    TRANSPORT:  LAGVT,RStation7350;

626$    TRANSPORT:  LAGVT,LStation7350;

;
;
;   Model statements for module: Station 91
;
629$    STATION,      LStation7350;

```

```

2732$    DELAY:    0.0,,VA:NEXT(630$);

;
;
;   Model statements for module: Decide 1007
;
630$    BRANCH,    1:
                If,Entity.Type==LCar,2733$,Yes:
                Else,2734$,Yes;
2733$    ASSIGN:    checkvehicletype7350.NumberOut True=checkvehicletype7350.NumberOut
True + 1:NEXT(631$);

2734$    ASSIGN:    checkvehicletype7350.NumberOut False=checkvehicletype7350.NumberOut
False + 1:NEXT(632$);

;
;
;   Model statements for module: Decide 1008
;
631$    BRANCH,    1:
                If,SpeedatRStation7250 >= VTU(LAGV,LCar#),2735$,Yes:
                Else,2736$,Yes;
2735$    ASSIGN:    Decide 1008.NumberOut True=Decide 1008.NumberOut True + 1:NEXT(633$);

2736$    ASSIGN:    Decide 1008.NumberOut False=Decide 1008.NumberOut False +
1:NEXT(659$);

;
;
;   Model statements for module: Decide 1010
;
633$    BRANCH,    1:
                If,SpeedatRStation7250>75,634$,Yes:
                If,SpeedatRStation7250 > 55 && SpeedatRStation7250 <= 75,647$,Yes:
                If,SpeedatRStation7250 >35 && SpeedatRStation7250<=55,650$,Yes:
                If,SpeedatRStation7250 >10 && SpeedatRStation7250 <=35,653$,Yes:
                Else,656$,Yes;

;
;
;   Model statements for module: Decide 1015
;
656$    BRANCH,    1:
                If,NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL8,4) == 0,2739$,Yes:
                Else,2740$,Yes;
2739$    ASSIGN:    CheckGapL7350Speed10.NumberOut
True=CheckGapL7350Speed10.NumberOut True + 1:NEXT(657$);

2740$    ASSIGN:    CheckGapL7350Speed10.NumberOut
False=CheckGapL7350Speed10.NumberOut False + 1:NEXT(658$);

```

```

657$    TRANSPORT:  LAGV,RStation7450;

658$    TRANSPORT:  LAGV,LStation7450;

;
;
;   Model statements for module: Decide 1011
;
634$    BRANCH,      1:
          If,
              NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL9,4) &&NZ(RL10,1)
&&NZ(RL10,2) &&NZ(RL11,1) && NZ(RL11,2) && NZ(RL8,4) == 0,
              2741$,Yes:
              Else,2742$,Yes;
2741$    ASSIGN:      CheckGapL7350Speed75.NumberOut
True=CheckGapL7350Speed75.NumberOut True + 1:NEXT(635$);

2742$    ASSIGN:      CheckGapL7350Speed75.NumberOut
False=CheckGapL7350Speed75.NumberOut False + 1:NEXT(636$);

635$    TRANSPORT:  LAGV,RStation7450;

636$    TRANSPORT:  LAGV,LStation7450;

;
;
;   Model statements for module: Decide 1012
;
647$    BRANCH,      1:
          If,NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL9,4) &&NZ(RL10,1)
&&NZ(RL10,2) && NZ(RL8,4) == 0,2743$,
          Yes:
          Else,2744$,Yes;
2743$    ASSIGN:      CheckGapL7350Speed55_75.NumberOut
True=CheckGapL7350Speed55_75.NumberOut True + 1:NEXT(648$);

2744$    ASSIGN:      CheckGapL7350Speed55_75.NumberOut
False=CheckGapL7350Speed55_75.NumberOut False + 1:NEXT(649$);

648$    TRANSPORT:  LAGV,RStation7450;

649$    TRANSPORT:  LAGV,LStation7450;

;
;
;   Model statements for module: Decide 1013
;
650$    BRANCH,      1:

```

```

                If,NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL9,4) &&NZ(RL10,1) &&
NZ(RL8,4) == 0,2745$,Yes:
                Else,2746$,Yes;
2745$    ASSIGN:    CheckGapL7350Speed35_55.NumberOut
True=CheckGapL7350Speed35_55.NumberOut True + 1:NEXT(651$);

2746$    ASSIGN:    CheckGapL7350Speed35_55.NumberOut
False=CheckGapL7350Speed35_55.NumberOut False + 1:NEXT(652$);

651$    TRANSPORT:  LAGV,RStation7450;

652$    TRANSPORT:  LAGV,LStation7450;

;
;
;    Model statements for module: Decide 1014
;
653$    BRANCH,      1:
                If,NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) && NZ(RL8,4) == 0,2747$,Yes:
                Else,2748$,Yes;
2747$    ASSIGN:    CheckGapL7350Speed10_35.NumberOut
True=CheckGapL7350Speed10_35.NumberOut True + 1:NEXT(654$);

2748$    ASSIGN:    CheckGapL7350Speed10_35.NumberOut
False=CheckGapL7350Speed10_35.NumberOut False + 1:NEXT(655$);

654$    TRANSPORT:  LAGV,RStation7450;

655$    TRANSPORT:  LAGV,LStation7450;

;
;
;    Model statements for module: Decide 1016
;
659$    BRANCH,      1:
                If,SpeedatRStation7250>75,660$,Yes:
                If,SpeedatRStation7250 > 55 && SpeedatRStation7250 <= 75,663$,Yes:
                If,SpeedatRStation7250 >35 && SpeedatRStation7250 <=55,666$,Yes:
                If,SpeedatRStation7250 >10 && SpeedatRStation7250 <=35,669$,Yes:
                Else,672$,Yes;

;
;
;    Model statements for module: Decide 1021
;
672$    BRANCH,      1:
                If,NZ(RL8,3)&&NZ(RL8,4) &&NZ(RL9,1) == 0,2751$,Yes:
                Else,2752$,Yes;
2751$    ASSIGN:    CheckGapL7350Speed10less.NumberOut
True=CheckGapL7350Speed10less.NumberOut True + 1:NEXT(673$);

```



```

2752$    ASSIGN:    CheckGapL7350Speed10less.NumberOut
False=CheckGapL7350Speed10less.NumberOut False + 1:NEXT(674$);

673$     TRANSPORT:  LAGV,RStation7450;

674$     TRANSPORT:  LAGV,LStation7450;

;
;
;   Model statements for module: Decide 1017
;
660$     BRANCH,     1:
                If,
                    NZ(RL7,2) &&NZ(RL7,3) && NZ(RL7,4) && NZ(RL8,1)
&&NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4) &&NZ(RL9,1)&&NZ(RL9,2) == 0,
                    2753$,Yes;
                Else,2754$,Yes;
2753$     ASSIGN:    CheckGapL7350Speed75less.NumberOut
True=CheckGapL7350Speed75less.NumberOut True + 1:NEXT(661$);

2754$     ASSIGN:    CheckGapL7350Speed75less.NumberOut
False=CheckGapL7350Speed75less.NumberOut False + 1:NEXT(662$);

661$     TRANSPORT:  LAGV,RStation7450;

662$     TRANSPORT:  LAGV,LStation7450;

;
;
;   Model statements for module: Decide 1018
;
663$     BRANCH,     1:
                If,NZ(RL7,3) && NZ(RL7,4) && NZ(RL8,1)
&&NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4) &&NZ(RL9,1) == 0,2755$,Yes;
                Else,2756$,Yes;
2755$     ASSIGN:    CheckGapL7350Speed55_75less.NumberOut
True=CheckGapL7350Speed55_75less.NumberOut True + 1
                :NEXT(664$);

2756$     ASSIGN:    CheckGapL7350Speed55_75less.NumberOut
False=CheckGapL7350Speed55_75less.NumberOut False + 1
                :NEXT(665$);

664$     TRANSPORT:  LAGV,RStation7450;

665$     TRANSPORT:  LAGV,LStation7450;

;
;
;   Model statements for module: Decide 1019

```

```

;
666$    BRANCH,    1:
          If,NZ(RL7,4) && NZ(RL8,1) &&NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4)
&&NZ(RL9,1) == 0,2757$,Yes:
          Else,2758$,Yes;
2757$    ASSIGN:    CheckGapL7350Speed35_55less.NumberOut
True=CheckGapL7350Speed35_55less.NumberOut True + 1
          :NEXT(667$);

2758$    ASSIGN:    CheckGapL7350Speed35_55less.NumberOut
False=CheckGapL7350Speed35_55less.NumberOut False + 1
          :NEXT(668$);

667$    TRANSPORT:  LAGV,RStation7450;

668$    TRANSPORT:  LAGV,LStation7450;

;
;
;   Model statements for module: Decide 1020
;
669$    BRANCH,    1:
          If,NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4) &&NZ(RL9,1) == 0,2759$,Yes:
          Else,2760$,Yes;
2759$    ASSIGN:    CheckGapL7350Speed10_35less.NumberOut
True=CheckGapL7350Speed10_35less.NumberOut True + 1
          :NEXT(670$);

2760$    ASSIGN:    CheckGapL7350Speed10_35less.NumberOut
False=CheckGapL7350Speed10_35less.NumberOut False + 1
          :NEXT(671$);

670$    TRANSPORT:  LAGV,RStation7450;

671$    TRANSPORT:  LAGV,LStation7450;

;
;
;   Model statements for module: Decide 1009
;
632$    BRANCH,    1:
          If,SpeedatRStation7250 >= VTU(LAGVT,LTruck#),2761$,Yes:
          Else,2762$,Yes;
2761$    ASSIGN:    Decide 1009.NumberOut True=Decide 1009.NumberOut True + 1:NEXT(675$);

2762$    ASSIGN:    Decide 1009.NumberOut False=Decide 1009.NumberOut False +
1:NEXT(681$);

;
;

```

```

;   Model statements for module: Decide 1022
;
675$   BRANCH,      1:
        If,SpeedatRStation7250>75,676$,Yes:
        If,SpeedatRStation7250 > 55 && SpeedatRStation7250 <= 75,677$,Yes:
        If,SpeedatRStation7250 >35 && SpeedatRStation7250 <=55,678$,Yes:
        If,SpeedatRStation7250 >10 && SpeedatRStation7250 <=35,679$,Yes:
        Else,680$,Yes;

;
;
;   Model statements for module: Decide 1027
;
680$   BRANCH,      1:
        If,NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL9,4) &&NZ(RL8,4) ==
0,2765$,Yes:
        Else,2766$,Yes;
2765$   ASSIGN:      CheckGapL7350Speed10t.NumberOut
True=CheckGapL7350Speed10t.NumberOut True + 1:NEXT(645$);

2766$   ASSIGN:      CheckGapL7350Speed10t.NumberOut
False=CheckGapL7350Speed10t.NumberOut False + 1:NEXT(646$);

645$   TRANSPORT:    LAGVT,RStation7450;

646$   TRANSPORT:    LAGVT,LStation7450;

;
;
;   Model statements for module: Decide 1023
;
676$   BRANCH,      1:
        If,
        NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL9,4) &&NZ(RL10,1)
&&NZ(RL10,2) &&NZ(RL11,1) && NZ(RL11,2) &&NZ(RL8,3) && NZ(RL8,4) == 0,
2767$,Yes:
        Else,2768$,Yes;
2767$   ASSIGN:      CheckGapL7350Speed75t.NumberOut
True=CheckGapL7350Speed75t.NumberOut True + 1:NEXT(637$);

2768$   ASSIGN:      CheckGapL7350Speed75t.NumberOut
False=CheckGapL7350Speed75t.NumberOut False + 1:NEXT(638$);

637$   TRANSPORT:    LAGVT,RStation7450;

638$   TRANSPORT:    LAGVT,LStation7450;

;
;
;   Model statements for module: Decide 1024
;

```

```

677$    BRANCH,    1:
        If,
        NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL9,4) &&NZ(RL10,1)
&&NZ(RL10,2) &&NZ(RL11,1) && NZ(RL8,4) == 0,
        2769$,Yes:
        Else,2770$,Yes;
2769$    ASSIGN:    CheckGapL7350Speed55_75t.NumberOut
True=CheckGapL7350Speed55_75t.NumberOut True + 1:NEXT(640$);

2770$    ASSIGN:    CheckGapL7350Speed55_75t.NumberOut
False=CheckGapL7350Speed55_75t.NumberOut False + 1:NEXT(639$);

640$    TRANSPORT:  LAGVT,RStation7450;

639$    TRANSPORT:  LAGVT,LStation7450;

;
;
;   Model statements for module: Decide 1025
;
678$    BRANCH,    1:
        If,NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL9,4) &&NZ(RL10,1)
&&NZ(RL10,2) &&NZ(RL8,4) == 0,2771$,
        Yes:
        Else,2772$,Yes;
2771$    ASSIGN:    CheckGapL7350Speed35_55t.NumberOut
True=CheckGapL7350Speed35_55t.NumberOut True + 1:NEXT(641$);

2772$    ASSIGN:    CheckGapL7350Speed35_55t.NumberOut
False=CheckGapL7350Speed35_55t.NumberOut False + 1:NEXT(642$);

641$    TRANSPORT:  LAGVT,RStation7450;

642$    TRANSPORT:  LAGVT,LStation7450;

;
;
;   Model statements for module: Decide 1026
;
679$    BRANCH,    1:
        If,NZ(RL9,1) &&NZ(RL9,2) &&NZ(RL9,3) &&NZ(RL9,4) &&NZ(RL10,1)
&&NZ(RL8,4) == 0,2773$,Yes:
        Else,2774$,Yes;
2773$    ASSIGN:    CheckGapL7350Speed10_35t.NumberOut
True=CheckGapL7350Speed10_35t.NumberOut True + 1:NEXT(643$);

2774$    ASSIGN:    CheckGapL7350Speed10_35t.NumberOut
False=CheckGapL7350Speed10_35t.NumberOut False + 1:NEXT(644$);

643$    TRANSPORT:  LAGVT,RStation7450;

```

```

644$    TRANSPORT:  LAGVT,LStation7450;

;
;
;  Model statements for module: Decide 1028
;
681$    BRANCH,      1:
        If,SpeedatRStation7250>75,682$,Yes:
        If,SpeedatRStation7250 > 55 && SpeedatRStation7250<= 75,683$,Yes:
        If,SpeedatRStation7250 >35 && SpeedatRStation7250 <=55,684$,Yes:
        If,SpeedatRStation7250 >10 && SpeedatRStation7250 <=35,685$,Yes:
        Else,686$,Yes;

;
;
;  Model statements for module: Decide 1033
;
686$    BRANCH,      1:
        If,NZ(RL8,1) &&NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4) &&NZ(RL9,1)==
0,2777$,Yes:
        Else,2778$,Yes;
2777$    ASSIGN:      CheckGapL7350Speed10lesst.NumberOut
True=CheckGapL7350Speed10lesst.NumberOut True + 1:NEXT(695$);

2778$    ASSIGN:      CheckGapL7350Speed10lesst.NumberOut
False=CheckGapL7350Speed10lesst.NumberOut False + 1:NEXT(696$);

695$    TRANSPORT:  LAGVT,RStation7450;

696$    TRANSPORT:  LAGVT,LStation7450;

;
;
;  Model statements for module: Decide 1029
;
682$    BRANCH,      1:
        If,
        NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) && NZ(RL7,4) && NZ(RL8,1)
&&NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4) &&NZ(RL9,1)&&NZ(RL9,2) == 0,
        2779$,Yes:
        Else,2780$,Yes;
2779$    ASSIGN:      CheckGapL7350Speed75lesst.NumberOut
True=CheckGapL7350Speed75lesst.NumberOut True + 1:NEXT(687$);

2780$    ASSIGN:      CheckGapL7350Speed75lesst.NumberOut
False=CheckGapL7350Speed75lesst.NumberOut False + 1:NEXT(688$);

687$    TRANSPORT:  LAGVT,RStation7450;

688$    TRANSPORT:  LAGVT,LStation7450;

```

```

;
;
;   Model statements for module: Decide 1030
;
683$   BRANCH,      1:
        If,
            NZ(RL7,1) &&NZ(RL7,2) &&NZ(RL7,3) && NZ(RL7,4) && NZ(RL8,1)
&&NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4) &&NZ(RL9,1)== 0,
            2781$,Yes:
            Else,2782$,Yes;
2781$   ASSIGN:      CheckGapL7350Speed55_75lesst.NumberOut
True=CheckGapL7350Speed55_75lesst.NumberOut True + 1
        :NEXT(689$);

2782$   ASSIGN:      CheckGapL7350Speed55_75lesst.NumberOut
False=CheckGapL7350Speed55_75lesst.NumberOut False + 1
        :NEXT(690$);

689$   TRANSPORT:    LAGVT,RStation7450;

690$   TRANSPORT:    LAGVT,LStation7450;

;
;
;   Model statements for module: Decide 1031
;
684$   BRANCH,      1:
        If,NZ(RL7,3) && NZ(RL7,4) && NZ(RL8,1)
&&NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4) &&NZ(RL9,1)== 0,2783$,Yes:
            Else,2784$,Yes;
2783$   ASSIGN:      CheckGapL7350Speed35_55lesst.NumberOut
True=CheckGapL7350Speed35_55lesst.NumberOut True + 1
        :NEXT(691$);

2784$   ASSIGN:      CheckGapL7350Speed35_55lesst.NumberOut
False=CheckGapL7350Speed35_55lesst.NumberOut False + 1
        :NEXT(692$);

691$   TRANSPORT:    LAGVT,RStation7450;

692$   TRANSPORT:    LAGVT,LStation7450;

;
;
;   Model statements for module: Decide 1032
;
685$   BRANCH,      1:
        If,NZ(RL7,4) && NZ(RL8,1) &&NZ(RL8,2)&&NZ(RL8,3)&&NZ(RL8,4)
&&NZ(RL9,1)== 0,2785$,Yes:
            Else,2786$,Yes;

```

```

2785$    ASSIGN:    CheckGapL7350Speed10_35lesst.NumberOut
True=CheckGapL7350Speed10_35lesst.NumberOut True + 1
          :NEXT(693$);

```

```

2786$    ASSIGN:    CheckGapL7350Speed10_35lesst.NumberOut
False=CheckGapL7350Speed10_35lesst.NumberOut False + 1
          :NEXT(694$);

```

```

693$     TRANSPORT:  LAGVT,RStation7450;

```

```

694$     TRANSPORT:  LAGVT,LStation7450;

```

```

;
;
;   Model statements for module: Station 101
;

```

```

697$     STATION,    RStation7450;
2789$     DELAY:     0.0,,VA:NEXT(735$);

```

```

;
;
;   Model statements for module: Decide 1288
;

```

```

735$     BRANCH,     1:
                If,Entity.Type==RCar,731$,Yes:
                If,Entity.Type==RTruck,732$,Yes:
                If,Entity.Type==LCar,733$,Yes:
                Else,734$,Yes;

```

```

;
;
;   Model statements for module: Assign 158
;

```

```

734$     ASSIGN:     SpeedatRStation7450=VTU(LAGVT,LTruck#):NEXT(701$);

```

```

;
;
;   Model statements for module: Decide 1277
;

```

```

701$     BRANCH,     1:
                If,Entity.Type==RCar,2792$,Yes:
                Else,2793$,Yes;

```

```

2792$     ASSIGN:     Decide 1277.NumberOut True=Decide 1277.NumberOut True + 1:NEXT(700$);

```

```

2793$     ASSIGN:     Decide 1277.NumberOut False=Decide 1277.NumberOut False +
1:NEXT(703$);

```

```

700$     TRANSPORT:  RAGV,RStation7500,NORM(90,4);

```

```

;
;
;   Model statements for module: Decide 1278
;
703$   BRANCH,      1:
        If,Entity.Type==RTruck,2794$,Yes:
        Else,2795$,Yes;
2794$   ASSIGN:      Decide 1278.NumberOut True=Decide 1278.NumberOut True + 1:NEXT(702$);

2795$   ASSIGN:      Decide 1278.NumberOut False=Decide 1278.NumberOut False +
1:NEXT(704$);

702$   TRANSPORT:    RAGVT,RStation7500,NORM(90,4);

;
;
;   Model statements for module: Decide 1279
;
704$   BRANCH,      1:
        If,Entity.Type==LCar,2796$,Yes:
        Else,2797$,Yes;
2796$   ASSIGN:      Decide 1279.NumberOut True=Decide 1279.NumberOut True + 1:NEXT(705$);

2797$   ASSIGN:      Decide 1279.NumberOut False=Decide 1279.NumberOut False +
1:NEXT(706$);

705$   TRANSPORT:    LAGV,RStation7500,NORM(90,4);

706$   TRANSPORT:    LAGVT,RStation7500,NORM(90,4);

;
;
;   Model statements for module: Assign 155
;
731$   ASSIGN:      SpeedatRStation7450=VTU(RAGV,RCar#):NEXT(701$);

;
;
;   Model statements for module: Assign 156
;
732$   ASSIGN:      SpeedatRStation7450=VTU(RAGVT,RTruck#):NEXT(701$);

;
;
;   Model statements for module: Assign 157
;
733$   ASSIGN:      SpeedatRStation7450=VTU(LAGV,LCar#):NEXT(701$);

```



```

;
;
; Model statements for module: Station 102
;

698$    STATION,    RStation7500;
2800$    DELAY:      0.0,,VA:NEXT(725$);

;
;
; Model statements for module: Decide 1286
;
725$    BRANCH,      1:
          If,Entity.Type==RCar,721$,Yes:
          If,Entity.Type==RTruck,722$,Yes:
          If,Entity.Type==LCar,723$,Yes:
          Else,724$,Yes;

;
;
; Model statements for module: Assign 150
;
724$    ASSIGN:      SpeedatRStation7500=VTU(LAGVT,LTruck#):NEXT(708$);

;
;
; Model statements for module: Decide 1280
;
708$    BRANCH,      1:
          If,Entity.Type==RCar,2803$,Yes:
          Else,2804$,Yes;
2803$    ASSIGN:      Decide 1280.NumberOut True=Decide 1280.NumberOut True + 1:NEXT(707$);

2804$    ASSIGN:      Decide 1280.NumberOut False=Decide 1280.NumberOut False +
1:NEXT(710$);

707$    TRANSPORT:    RAGV,RStation7550;

;
;
; Model statements for module: Decide 1281
;
710$    BRANCH,      1:
          If,Entity.Type==RTruck,2805$,Yes:
          Else,2806$,Yes;
2805$    ASSIGN:      Decide 1281.NumberOut True=Decide 1281.NumberOut True + 1:NEXT(709$);

2806$    ASSIGN:      Decide 1281.NumberOut False=Decide 1281.NumberOut False +
1:NEXT(711$);

```

```

709$    TRANSPORT:  RAGVT,RStation7550;

;
;
;  Model statements for module: Decide 1282
;
711$    BRANCH,      1:
          If,Entity.Type==LCar,2807$,Yes:
          Else,2808$,Yes;
2807$    ASSIGN:      Decide 1282.NumberOut True=Decide 1282.NumberOut True + 1:NEXT(712$);

2808$    ASSIGN:      Decide 1282.NumberOut False=Decide 1282.NumberOut False +
1:NEXT(713$);

712$    TRANSPORT:  LAGV,RStation7550;

713$    TRANSPORT:  LAGVT,RStation7550;

;
;
;  Model statements for module: Assign 147
;
721$    ASSIGN:      SpeedatRStation7500=VTU(RAGV,RCar#):NEXT(708$);

;
;
;  Model statements for module: Assign 148
;
722$    ASSIGN:      SpeedatRStation7500=VTU(RAGVT,RTruck#):NEXT(708$);

;
;
;  Model statements for module: Assign 149
;
723$    ASSIGN:      SpeedatRStation7500=VTU(LAGV,LCar#):NEXT(708$);

;
;
;  Model statements for module: Station 103
;

699$    STATION,      RStation7550;
2811$    DELAY:        0.0,,VA:NEXT(730$);

;
;

```

```

;   Model statements for module: Decide 1287
;
730$   BRANCH,      1:
        If,Entity.Type==RCar,726$,Yes:
        If,Entity.Type==RTruck,727$,Yes:
        If,Entity.Type==LCar,728$,Yes:
        Else,729$,Yes;

;
;
;   Model statements for module: Assign 154
;
729$   ASSIGN:      SpeedatRStation7550=VTU(LAGVT,LTruck#):NEXT(715$);

;
;
;   Model statements for module: Decide 1283
;
715$   BRANCH,      1:
        If,Entity.Type==RCar,2814$,Yes:
        Else,2815$,Yes;
2814$   ASSIGN:      Decide 1283.NumberOut True=Decide 1283.NumberOut True + 1:NEXT(714$);
2815$   ASSIGN:      Decide 1283.NumberOut False=Decide 1283.NumberOut False +
1:NEXT(717$);

714$   TRANSPORT:   RAGV,RStation7600;

;
;
;   Model statements for module: Decide 1284
;
717$   BRANCH,      1:
        If,Entity.Type==RTruck,2816$,Yes:
        Else,2817$,Yes;
2816$   ASSIGN:      Decide 1284.NumberOut True=Decide 1284.NumberOut True + 1:NEXT(716$);
2817$   ASSIGN:      Decide 1284.NumberOut False=Decide 1284.NumberOut False +
1:NEXT(718$);

716$   TRANSPORT:   RAGVT,RStation7600;

;
;
;   Model statements for module: Decide 1285
;
718$   BRANCH,      1:
        If,Entity.Type==LCar,2818$,Yes:
        Else,2819$,Yes;
2818$   ASSIGN:      Decide 1285.NumberOut True=Decide 1285.NumberOut True + 1:NEXT(719$);

```

```

2819$    ASSIGN:    Decide 1285.NumberOut False=Decide 1285.NumberOut False +
1:NEXT(720$);

719$    TRANSPORT:  LAGV,RStation7600;

720$    TRANSPORT:  LAGVT,RStation7600;

;
;
;   Model statements for module: Assign 151
;
726$    ASSIGN:    SpeedatRStation7550=VTU(RAGV,RCar#):NEXT(715$);

;
;
;   Model statements for module: Assign 152
;
727$    ASSIGN:    SpeedatRStation7550=VTU(RAGVT,RTruck#):NEXT(715$);

;
;
;   Model statements for module: Assign 153
;
728$    ASSIGN:    SpeedatRStation7550=VTU(LAGV,LCar#):NEXT(715$);

;
;
;   Model statements for module: Station 104
;

736$    STATION,    LStation7450;
2822$    DELAY:     0.0,,VA:NEXT(737$);

;
;
;   Model statements for module: Decide 1289
;
737$    BRANCH,     1:
                If,Entity.Type==LCar,2823$,Yes:
                Else,2824$,Yes;
2823$    ASSIGN:    checkvehicletype7450.NumberOut True=checkvehicletype7450.NumberOut
True + 1:NEXT(738$);

2824$    ASSIGN:    checkvehicletype7450.NumberOut False=checkvehicletype7450.NumberOut
False + 1:NEXT(739$);

```

```

;
;
;   Model statements for module: Decide 1290
;
738$   BRANCH,      1:
        If,SpeedatRStation7350 >= VTU(LAGV,LCar#),2825$,Yes:
        Else,2826$,Yes;
2825$   ASSIGN:      Decide 1290.NumberOut True=Decide 1290.NumberOut True + 1:NEXT(740$);

2826$   ASSIGN:      Decide 1290.NumberOut False=Decide 1290.NumberOut False +
1:NEXT(766$);

;
;
;   Model statements for module: Decide 1292
;
740$   BRANCH,      1:
        If,SpeedatRStation7350>75,741$,Yes:
        If,SpeedatRStation7350 > 55 && SpeedatRStation7350 <= 75,754$,Yes:
        If,SpeedatRStation7350 >35 && SpeedatRStation7350<=55,757$,Yes:
        If,SpeedatRStation7350 >10 && SpeedatRStation7350 <=35,760$,Yes:
        Else,763$,Yes;

;
;
;   Model statements for module: Decide 1297
;
763$   BRANCH,      1:
        If,NZ(RL10,1) &&NZ(RL10,2) && NZ(RL9,4) == 0,2829$,Yes:
        Else,2830$,Yes;
2829$   ASSIGN:      CheckGapL7450Speed10.NumberOut
True=CheckGapL7450Speed10.NumberOut True + 1:NEXT(764$);

2830$   ASSIGN:      CheckGapL7450Speed10.NumberOut
False=CheckGapL7450Speed10.NumberOut False + 1:NEXT(765$);

764$   TRANSPORT:    LAGV,RStation7500,NORM(101,3);

765$   TRANSPORT:    LAGV,LStation7500,NORM(101,3);

;
;
;   Model statements for module: Decide 1293
;
741$   BRANCH,      1:
        If,
        NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1)
&&NZ(RL12,2) &&NZ(RL13,1) && NZ(RL13,2) && NZ(RL9,4) == 0,
        2831$,Yes:
        Else,2832$,Yes;

```

```

2831$    ASSIGN:    CheckGapL7450Speed75.NumberOut
True=CheckGapL7450Speed75.NumberOut True + 1:NEXT(742$);

2832$    ASSIGN:    CheckGapL7450Speed75.NumberOut
False=CheckGapL7450Speed75.NumberOut False + 1:NEXT(743$);

742$    TRANSPORT:  LAGV,RStation7500,NORM(101,3);

743$    TRANSPORT:  LAGV,LStation7500,NORM(101,3);

;
;
; Model statements for module: Decide 1294
;
754$    BRANCH,      1:
                If,NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1)
&&NZ(RL12,2) && NZ(RL9,4) == 0,
                2833$,Yes;
                Else,2834$,Yes;
2833$    ASSIGN:    CheckGapL7450Speed55_75.NumberOut
True=CheckGapL7450Speed55_75.NumberOut True + 1:NEXT(755$);

2834$    ASSIGN:    CheckGapL7450Speed55_75.NumberOut
False=CheckGapL7450Speed55_75.NumberOut False + 1:NEXT(756$);

755$    TRANSPORT:  LAGV,RStation7500,NORM(101,3);

756$    TRANSPORT:  LAGV,LStation7500,NORM(101,3);

;
;
; Model statements for module: Decide 1295
;
757$    BRANCH,      1:
                If,NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&
NZ(RL9,4) == 0,2835$,Yes;
                Else,2836$,Yes;
2835$    ASSIGN:    CheckGapL7450Speed35_55.NumberOut
True=CheckGapL7450Speed35_55.NumberOut True + 1:NEXT(758$);

2836$    ASSIGN:    CheckGapL7450Speed35_55.NumberOut
False=CheckGapL7450Speed35_55.NumberOut False + 1:NEXT(759$);

758$    TRANSPORT:  LAGV,RStation7500,NORM(101,3);

759$    TRANSPORT:  LAGV,LStation7500,NORM(101,3);

;
;
; Model statements for module: Decide 1296

```

```

;
760$    BRANCH,    1:
          If,NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) && NZ(RL9,4) == 0,2837$,Yes:
          Else,2838$,Yes;
2837$    ASSIGN:    CheckGapL7450Speed10_35.NumberOut
True=CheckGapL7450Speed10_35.NumberOut True + 1:NEXT(761$);

2838$    ASSIGN:    CheckGapL7450Speed10_35.NumberOut
False=CheckGapL7450Speed10_35.NumberOut False + 1:NEXT(762$);

761$    TRANSPORT:  LAGV,RStation7500,NORM(101,3);

762$    TRANSPORT:  LAGV,LStation7500,NORM(101,3);

;
;
;    Model statements for module: Decide 1298
;
766$    BRANCH,    1:
          If,SpeedatRStation7350>75,767$,Yes:
          If,SpeedatRStation7350 > 55 && SpeedatRStation7350 <= 75,770$,Yes:
          If,SpeedatRStation7350>35 && SpeedatRStation7350 <=55,773$,Yes:
          If,SpeedatRStation7350 >10 && SpeedatRStation7350 <=35,776$,Yes:
          Else,779$,Yes;

;
;
;    Model statements for module: Decide 1303
;
779$    BRANCH,    1:
          If,NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1)== 0,2841$,Yes:
          Else,2842$,Yes;
2841$    ASSIGN:    CheckGapL7450Speed10less.NumberOut
True=CheckGapL7450Speed10less.NumberOut True + 1:NEXT(780$);

2842$    ASSIGN:    CheckGapL7450Speed10less.NumberOut
False=CheckGapL7450Speed10less.NumberOut False + 1:NEXT(781$);

780$    TRANSPORT:  LAGV,RStation7500,NORM(101,3);

781$    TRANSPORT:  LAGV,LStation7500,NORM(101,3);

;
;
;    Model statements for module: Decide 1299
;
767$    BRANCH,    1:
          If,
          NZ(RL8,2) && NZ(RL8,3) &&NZ(RL8,4)&&NZ(RL9,1)&&NZ(RL9,2)
&&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2) == 0,
          2843$,Yes:

```

```

                Else,2844$,Yes;
2843$    ASSIGN:    CheckGapL7450Speed75less.NumberOut
True=CheckGapL7450Speed75less.NumberOut True + 1:NEXT(768$);

2844$    ASSIGN:    CheckGapL7450Speed75less.NumberOut
False=CheckGapL7450Speed75less.NumberOut False + 1:NEXT(769$);

768$    TRANSPORT:  LAGV,RStation7500,NORM(101,3);

769$    TRANSPORT:  LAGV,LStation7500,NORM(101,3);

;
;
;    Model statements for module: Decide 1300
;
770$    BRANCH,      1:
                If,NZ(RL8,3) &&NZ(RL8,4)&&NZ(RL9,1)&&NZ(RL9,2)
&&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1)== 0,2845$,Yes:
                Else,2846$,Yes;
2845$    ASSIGN:    CheckGapL7450Speed55_75less.NumberOut
True=CheckGapL7450Speed55_75less.NumberOut True + 1
                :NEXT(771$);

2846$    ASSIGN:    CheckGapL7450Speed55_75less.NumberOut
False=CheckGapL7450Speed55_75less.NumberOut False + 1
                :NEXT(772$);

771$    TRANSPORT:  LAGV,RStation7500,NORM(101,3);

772$    TRANSPORT:  LAGV,LStation7500,NORM(101,3);

;
;
;    Model statements for module: Decide 1301
;
773$    BRANCH,      1:
                If,NZ(RL8,4)&&NZ(RL9,1)&&NZ(RL9,2)
&&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1)== 0,2847$,Yes:
                Else,2848$,Yes;
2847$    ASSIGN:    CheckGapL7450Speed35_55less.NumberOut
True=CheckGapL7450Speed35_55less.NumberOut True + 1
                :NEXT(774$);

2848$    ASSIGN:    CheckGapL7450Speed35_55less.NumberOut
False=CheckGapL7450Speed35_55less.NumberOut False + 1
                :NEXT(775$);

774$    TRANSPORT:  LAGV,RStation7500,NORM(101,3);

775$    TRANSPORT:  LAGV,LStation7500,NORM(101,3);

```



```

;
;
;   Model statements for module: Decide 1302
;
776$   BRANCH,      1:
        If,NZ(RL9,2) &&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1)== 0,2849$,Yes:
        Else,2850$,Yes;
2849$   ASSIGN:      CheckGapL7450Speed10_35less.NumberOut
True=CheckGapL7450Speed10_35less.NumberOut True + 1
        :NEXT(777$);

2850$   ASSIGN:      CheckGapL7450Speed10_35less.NumberOut
False=CheckGapL7450Speed10_35less.NumberOut False + 1
        :NEXT(778$);

777$   TRANSPORT:    LAGV,RStation7500,NORM(101,3);

778$   TRANSPORT:    LAGV,LStation7500,NORM(101,3);

;
;
;   Model statements for module: Decide 1291
;
739$   BRANCH,      1:
        If,SpeedatRStation7350 >= VTU(LAGVT,LTruck#),2851$,Yes:
        Else,2852$,Yes;
2851$   ASSIGN:      Decide 1291.NumberOut True=Decide 1291.NumberOut True + 1:NEXT(782$);

2852$   ASSIGN:      Decide 1291.NumberOut False=Decide 1291.NumberOut False +
1:NEXT(788$);

;
;
;   Model statements for module: Decide 1304
;
782$   BRANCH,      1:
        If,SpeedatRStation7350>75,783$,Yes:
        If,SpeedatRStation7350 > 55 && SpeedatRStation7350 <= 75,784$,Yes:
        If,SpeedatRStation7350 >35 && SpeedatRStation7350 <=55,785$,Yes:
        If,SpeedatRStation7350 >10 && SpeedatRStation7350 <=35,786$,Yes:
        Else,787$,Yes;

;
;
;   Model statements for module: Decide 1309
;
787$   BRANCH,      1:
        If,NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL9,4) ==
0,2855$,Yes:
        Else,2856$,Yes;

```

```

2855$    ASSIGN:    CheckGapL7450Speed10t.NumberOut
True=CheckGapL7450Speed10t.NumberOut True + 1:NEXT(752$);

2856$    ASSIGN:    CheckGapL7450Speed10t.NumberOut
False=CheckGapL7450Speed10t.NumberOut False + 1:NEXT(753$);

752$     TRANSPORT:  LAGVT,RStation7500,NORM(101,3);

753$     TRANSPORT:  LAGVT,LStation7500,NORM(101,3);

;
;
; Model statements for module: Decide 1305
;
783$     BRANCH,     1:
                If,
                    NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1)
&&NZ(RL12,2) &&NZ(RL13,1) && NZ(RL13,2) && NZ(RL9,3) && NZ(RL9,4) == 0,
                    2857$,Yes:
                    Else,2858$,Yes;
2857$     ASSIGN:    CheckGapL7450Speed75t.NumberOut
True=CheckGapL7450Speed75t.NumberOut True + 1:NEXT(744$);

2858$     ASSIGN:    CheckGapL7450Speed75t.NumberOut
False=CheckGapL7450Speed75t.NumberOut False + 1:NEXT(745$);

744$     TRANSPORT:  LAGVT,RStation7500,NORM(101,3);

745$     TRANSPORT:  LAGVT,LStation7500,NORM(101,3);

;
;
; Model statements for module: Decide 1306
;
784$     BRANCH,     1:
                If,
                    NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1)
&&NZ(RL12,2) &&NZ(RL13,1) && NZ(RL9,4) == 0,
                    2859$,Yes:
                    Else,2860$,Yes;
2859$     ASSIGN:    CheckGapL7450Speed55_75t.NumberOut
True=CheckGapL7450Speed55_75t.NumberOut True + 1:NEXT(747$);

2860$     ASSIGN:    CheckGapL7450Speed55_75t.NumberOut
False=CheckGapL7450Speed55_75t.NumberOut False + 1:NEXT(746$);

747$     TRANSPORT:  LAGVT,RStation7500,NORM(101,3);

746$     TRANSPORT:  LAGVT,LStation7500,NORM(101,3);

```

```

;
;
;   Model statements for module: Decide 1307
;
785$   BRANCH,      1:
        If,NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1)
&&NZ(RL12,2) && NZ(RL9,4) == 0,
        2861$,Yes:
        Else,2862$,Yes;
2861$   ASSIGN:      CheckGapL7450Speed35_55t.NumberOut
True=CheckGapL7450Speed35_55t.NumberOut True + 1:NEXT(748$);

2862$   ASSIGN:      CheckGapL7450Speed35_55t.NumberOut
False=CheckGapL7450Speed35_55t.NumberOut False + 1:NEXT(749$);

748$   TRANSPORT:    LAGVT,RStation7500,NORM(101,3);

749$   TRANSPORT:    LAGVT,LStation7500,NORM(101,3);

;
;
;   Model statements for module: Decide 1308
;
786$   BRANCH,      1:
        If,NZ(RL10,1) &&NZ(RL10,2) &&NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1)
&&NZ(RL9,4) == 0,2863$,Yes:
        Else,2864$,Yes;
2863$   ASSIGN:      CheckGapL7450Speed10_35t.NumberOut
True=CheckGapL7450Speed10_35t.NumberOut True + 1:NEXT(750$);

2864$   ASSIGN:      CheckGapL7450Speed10_35t.NumberOut
False=CheckGapL7450Speed10_35t.NumberOut False + 1:NEXT(751$);

750$   TRANSPORT:    LAGVT,RStation7500,NORM(101,3);

751$   TRANSPORT:    LAGVT,LStation7500,NORM(101,3);

;
;
;   Model statements for module: Decide 1310
;
788$   BRANCH,      1:
        If,SpeedatRStation7350>75,789$,Yes:
        If,SpeedatRStation7350 > 55 && SpeedatRStation7350<= 75,790$,Yes:
        If,SpeedatRStation7350 >35 && SpeedatRStation7350 <=55,791$,Yes:
        If,SpeedatRStation7350 >10 && SpeedatRStation7350 <=35,792$,Yes:
        Else,793$,Yes;

;
;
;   Model statements for module: Decide 1315

```

```

;
793$    BRANCH,    1:
          If,NZ(RL9,1)&&NZ(RL9,2) &&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1) ==
0,2867$,Yes:
          Else,2868$,Yes;
2867$    ASSIGN:    CheckGapL7450Speed10lesst.NumberOut
True=CheckGapL7450Speed10lesst.NumberOut True + 1:NEXT(802$);

2868$    ASSIGN:    CheckGapL7450Speed10lesst.NumberOut
False=CheckGapL7450Speed10lesst.NumberOut False + 1:NEXT(803$);

802$    TRANSPORT:  LAGVT,RStation7500,NORM(101,3);

803$    TRANSPORT:  LAGVT,LStation7500,NORM(101,3);

;
;
;    Model statements for module: Decide 1311
;
789$    BRANCH,    1:
          If,
          NZ(RL8,1) &&NZ(RL8,2) && NZ(RL8,3) &&NZ(RL8,4)&&NZ(RL9,1)&&NZ(RL9,2)
&&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2) == 0,
          2869$,Yes:
          Else,2870$,Yes;
2869$    ASSIGN:    CheckGapL7450Speed75lesst.NumberOut
True=CheckGapL7450Speed75lesst.NumberOut True + 1:NEXT(794$);

2870$    ASSIGN:    CheckGapL7450Speed75lesst.NumberOut
False=CheckGapL7450Speed75lesst.NumberOut False + 1:NEXT(795$);

794$    TRANSPORT:  LAGVT,RStation7500,NORM(101,3);

795$    TRANSPORT:  LAGVT,LStation7500,NORM(101,3);

;
;
;    Model statements for module: Decide 1312
;
790$    BRANCH,    1:
          If,
          NZ(RL8,1) &&NZ(RL8,2) && NZ(RL8,3) &&NZ(RL8,4)&&NZ(RL9,1)&&NZ(RL9,2)
&&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1) == 0,
          2871$,Yes:
          Else,2872$,Yes;
2871$    ASSIGN:    CheckGapL7450Speed55_75lesst.NumberOut
True=CheckGapL7450Speed55_75lesst.NumberOut True + 1
:NEXT(796$);

2872$    ASSIGN:    CheckGapL7450Speed55_75lesst.NumberOut
False=CheckGapL7450Speed55_75lesst.NumberOut False + 1

```

```

: NEXT(797$);

796$    TRANSPORT:  LAGVT,RStation7500,NORM(101,3);

797$    TRANSPORT:  LAGVT,LStation7500,NORM(101,3);

;
;
;   Model statements for module: Decide 1313
;
791$    BRANCH,      1:
          If,NZ(RL8,2) && NZ(RL8,3) &&NZ(RL8,4)&&NZ(RL9,1)&&NZ(RL9,2)
&&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1) == 0,
          2873$,Yes:
          Else,2874$,Yes;
2873$    ASSIGN:      CheckGapL7450Speed35_55lesst.NumberOut
True=CheckGapL7450Speed35_55lesst.NumberOut True + 1
          :NEXT(798$);

2874$    ASSIGN:      CheckGapL7450Speed35_55lesst.NumberOut
False=CheckGapL7450Speed35_55lesst.NumberOut False + 1
          :NEXT(799$);

798$    TRANSPORT:  LAGVT,RStation7500,NORM(101,3);

799$    TRANSPORT:  LAGVT,LStation7500,NORM(101,3);

;
;
;   Model statements for module: Decide 1314
;
792$    BRANCH,      1:
          If,NZ(RL8,4)&&NZ(RL9,1)&&NZ(RL9,2)
&&NZ(RL9,3)&&NZ(RL9,4)&&NZ(RL10,1) == 0,2875$,Yes:
          Else,2876$,Yes;
2875$    ASSIGN:      CheckGapL7450Speed10_35lesst.NumberOut
True=CheckGapL7450Speed10_35lesst.NumberOut True + 1
          :NEXT(800$);

2876$    ASSIGN:      CheckGapL7450Speed10_35lesst.NumberOut
False=CheckGapL7450Speed10_35lesst.NumberOut False + 1
          :NEXT(801$);

800$    TRANSPORT:  LAGVT,RStation7500,NORM(101,3);

801$    TRANSPORT:  LAGVT,LStation7500,NORM(101,3);

;
;
;   Model statements for module: Station 105

```

```

;

804$      STATION,      LStation7500;
2879$     DELAY:        0.0,,VA:NEXT(805$);

;
;
;   Model statements for module: Decide 1316
;
805$      BRANCH,      1:
                If,Entity.Type==LCar,2880$,Yes:
                Else,2881$,Yes;
2880$     ASSIGN:      checkvehicletype7500.NumberOut True=checkvehicletype7500.NumberOut
True + 1:NEXT(806$);

2881$     ASSIGN:      checkvehicletype7500.NumberOut False=checkvehicletype7500.NumberOut
False + 1:NEXT(807$);

;
;
;   Model statements for module: Decide 1317
;
806$      BRANCH,      1:
                If,SpeedatRStation7450 >= VTU(LAGV,LCar#),2882$,Yes:
                Else,2883$,Yes;
2882$     ASSIGN:      Decide 1317.NumberOut True=Decide 1317.NumberOut True + 1:NEXT(808$);

2883$     ASSIGN:      Decide 1317.NumberOut False=Decide 1317.NumberOut False +
1:NEXT(834$);

;
;
;   Model statements for module: Decide 1319
;
808$      BRANCH,      1:
                If,SpeedatRStation7450>75,809$,Yes:
                If,SpeedatRStation7450 > 55 && SpeedatRStation7450 <= 75,822$,Yes:
                If,SpeedatRStation7450 >35 && SpeedatRStation7450<=55,825$,Yes:
                If,SpeedatRStation7450 >10 && SpeedatRStation7450 <=35,828$,Yes:
                Else,831$,Yes;

;
;
;   Model statements for module: Decide 1324
;
831$      BRANCH,      1:
                If,NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL10,2) == 0,2886$,Yes:
                Else,2887$,Yes;
2886$     ASSIGN:      CheckGapL7500Speed10.NumberOut
True=CheckGapL7500Speed10.NumberOut True + 1:NEXT(832$);

```

```

2887$    ASSIGN:    CheckGapL7500Speed10.NumberOut
False=CheckGapL7500Speed10.NumberOut False + 1:NEXT(833$);

832$    TRANSPORT:  LAGV,RStation7550;

833$    TRANSPORT:  LAGV,LStation7550;

;
;
;   Model statements for module: Decide 1320
;
809$    BRANCH,      1:
        If,
            NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1)
&&NZ(RL13,2) &&NZ(RL14,1) && NZ(RL14,2) && NZ(RL10,2) == 0,
            2888$,Yes:
            Else,2889$,Yes;
2888$    ASSIGN:    CheckGapL7500Speed75.NumberOut
True=CheckGapL7500Speed75.NumberOut True + 1:NEXT(810$);

2889$    ASSIGN:    CheckGapL7500Speed75.NumberOut
False=CheckGapL7500Speed75.NumberOut False + 1:NEXT(811$);

810$    TRANSPORT:  LAGV,RStation7550;

811$    TRANSPORT:  LAGV,LStation7550;

;
;
;   Model statements for module: Decide 1321
;
822$    BRANCH,      1:
        If,NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1)
&&NZ(RL13,2) &&NZ(RL10,2) == 0,
            2890$,Yes:
            Else,2891$,Yes;
2890$    ASSIGN:    CheckGapL7500Speed55_75.NumberOut
True=CheckGapL7500Speed55_75.NumberOut True + 1:NEXT(823$);

2891$    ASSIGN:    CheckGapL7500Speed55_75.NumberOut
False=CheckGapL7500Speed55_75.NumberOut False + 1:NEXT(824$);

823$    TRANSPORT:  LAGV,RStation7550;

824$    TRANSPORT:  LAGV,LStation7550;

;
;
;   Model statements for module: Decide 1322

```

```

;
825$    BRANCH,    1:
            If,NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1)
&&NZ(RL10,2) == 0,2892$,Yes:
            Else,2893$,Yes;
2892$    ASSIGN:    CheckGapL7500Speed35_55.NumberOut
True=CheckGapL7500Speed35_55.NumberOut True + 1:NEXT(826$);

2893$    ASSIGN:    CheckGapL7500Speed35_55.NumberOut
False=CheckGapL7500Speed35_55.NumberOut False + 1:NEXT(827$);

826$    TRANSPORT:  LAGV,RStation7550;

827$    TRANSPORT:  LAGV,LStation7550;

;
;
;    Model statements for module: Decide 1323
;
828$    BRANCH,    1:
            If,NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL10,2) == 0,2894$,Yes:
            Else,2895$,Yes;
2894$    ASSIGN:    CheckGapL7500Speed10_35.NumberOut
True=CheckGapL7500Speed10_35.NumberOut True + 1:NEXT(829$);

2895$    ASSIGN:    CheckGapL7500Speed10_35.NumberOut
False=CheckGapL7500Speed10_35.NumberOut False + 1:NEXT(830$);

829$    TRANSPORT:  LAGV,RStation7550;

830$    TRANSPORT:  LAGV,LStation7550;

;
;
;    Model statements for module: Decide 1325
;
834$    BRANCH,    1:
            If,SpeedatRStation7450>75,835$,Yes:
            If,SpeedatRStation7450 > 55 && SpeedatRStation7450 <= 75,838$,Yes:
            If,SpeedatRStation7450>35 && SpeedatRStation7450 <=55,841$,Yes:
            If,SpeedatRStation7450 >10 && SpeedatRStation7450 <=35,844$,Yes:
            Else,847$,Yes;

;
;
;    Model statements for module: Decide 1330
;
847$    BRANCH,    1:
            If,NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1)== 0,2898$,Yes:
            Else,2899$,Yes;

```


2898\$ ASSIGN: CheckGapL7500Speed10less.NumberOut
True=CheckGapL7500Speed10less.NumberOut True + 1:NEXT(848\$);

2899\$ ASSIGN: CheckGapL7500Speed10less.NumberOut
False=CheckGapL7500Speed10less.NumberOut False + 1:NEXT(849\$);

848\$ TRANSPORT: LAGV,RStation7550;

849\$ TRANSPORT: LAGV,LStation7550;

;
;

; Model statements for module: Decide 1326

;

835\$ BRANCH, 1:
 If,
 NZ(RL8,4) && NZ(RL9,1) &&NZ(RL9,2)&&NZ(RL9,3)&&NZ(RL9,4)
 &&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2) == 0,
 2900\$,Yes:
 Else,2901\$,Yes;

2900\$ ASSIGN: CheckGapL7500Speed75less.NumberOut
True=CheckGapL7500Speed75less.NumberOut True + 1:NEXT(836\$);

2901\$ ASSIGN: CheckGapL7500Speed75less.NumberOut
False=CheckGapL7500Speed75less.NumberOut False + 1:NEXT(837\$);

836\$ TRANSPORT: LAGV,RStation7550;

837\$ TRANSPORT: LAGV,LStation7550;

;
;

; Model statements for module: Decide 1327

;

838\$ BRANCH, 1:
 If,NZ(RL9,1) &&NZ(RL9,2)&&NZ(RL9,3)&&NZ(RL9,4)
 &&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1)== 0,2902\$,Yes:
 Else,2903\$,Yes;

2902\$ ASSIGN: CheckGapL7500Speed55_75less.NumberOut
True=CheckGapL7500Speed55_75less.NumberOut True + 1
 :NEXT(839\$);

2903\$ ASSIGN: CheckGapL7500Speed55_75less.NumberOut
False=CheckGapL7500Speed55_75less.NumberOut False + 1
 :NEXT(840\$);

839\$ TRANSPORT: LAGV,RStation7550;

840\$ TRANSPORT: LAGV,LStation7550;

```

;
;
;   Model statements for module: Decide 1328
;
841$   BRANCH,      1:
        If,NZ(RL9,2)&&NZ(RL9,3)&&NZ(RL9,4)
&&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1)== 0,2904$,Yes:
        Else,2905$,Yes;
2904$   ASSIGN:      CheckGapL7500Speed35_55less.NumberOut
True=CheckGapL7500Speed35_55less.NumberOut True + 1
        :NEXT(842$);

2905$   ASSIGN:      CheckGapL7500Speed35_55less.NumberOut
False=CheckGapL7500Speed35_55less.NumberOut False + 1
        :NEXT(843$);

842$   TRANSPORT:    LAGV,RStation7550;

843$   TRANSPORT:    LAGV,LStation7550;

;
;
;   Model statements for module: Decide 1329
;
844$   BRANCH,      1:
        If,NZ(RL9,4) &&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1)== 0,2906$,Yes:
        Else,2907$,Yes;
2906$   ASSIGN:      CheckGapL7500Speed10_35less.NumberOut
True=CheckGapL7500Speed10_35less.NumberOut True + 1
        :NEXT(845$);

2907$   ASSIGN:      CheckGapL7500Speed10_35less.NumberOut
False=CheckGapL7500Speed10_35less.NumberOut False + 1
        :NEXT(846$);

845$   TRANSPORT:    LAGV,RStation7550;

846$   TRANSPORT:    LAGV,LStation7550;

;
;
;   Model statements for module: Decide 1318
;
807$   BRANCH,      1:
        If,SpeedatRStation7450 >= VTU(LAGVT,LTruck#),2908$,Yes:
        Else,2909$,Yes;
2908$   ASSIGN:      Decide 1318.NumberOut True=Decide 1318.NumberOut True + 1:NEXT(850$);

2909$   ASSIGN:      Decide 1318.NumberOut False=Decide 1318.NumberOut False +
1:NEXT(856$);

```

```

;
;
; Model statements for module: Decide 1331
;
850$    BRANCH,    1:
        If,SpeedatRStation7450>75,851$,Yes:
        If,SpeedatRStation7450 > 55 && SpeedatRStation7450 <= 75,852$,Yes:
        If,SpeedatRStation7450 >35 && SpeedatRStation7450 <=55,853$,Yes:
        If,SpeedatRStation7450 >10 && SpeedatRStation7450 <=35,854$,Yes:
        Else,855$,Yes;

;
;
; Model statements for module: Decide 1336
;
855$    BRANCH,    1:
        If,NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL12,2) && NZ(RL10,2) ==
0,2912$,Yes:
        Else,2913$,Yes;
2912$    ASSIGN:    CheckGapL7500Speed10t.NumberOut
True=CheckGapL7500Speed10t.NumberOut True + 1:NEXT(820$);

2913$    ASSIGN:    CheckGapL7500Speed10t.NumberOut
False=CheckGapL7500Speed10t.NumberOut False + 1:NEXT(821$);

820$    TRANSPORT:  LAGVT,RStation7550;

821$    TRANSPORT:  LAGVT,LStation7550;

;
;
; Model statements for module: Decide 1332
;
851$    BRANCH,    1:
        If,
        NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1)
&&NZ(RL13,2) &&NZ(RL14,1) && NZ(RL14,2) && NZ(RL10,1)&& NZ(RL10,2) == 0,
        2914$,Yes:
        Else,2915$,Yes;
2914$    ASSIGN:    CheckGapL7500Speed75t.NumberOut
True=CheckGapL7500Speed75t.NumberOut True + 1:NEXT(812$);

2915$    ASSIGN:    CheckGapL7500Speed75t.NumberOut
False=CheckGapL7500Speed75t.NumberOut False + 1:NEXT(813$);

812$    TRANSPORT:  LAGVT,RStation7550;

813$    TRANSPORT:  LAGVT,LStation7550;

;

```

```

;
;   Model statements for module: Decide 1333
;
852$   BRANCH,    1:
        If,
            NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1)
&&NZ(RL13,2) &&NZ(RL14,1) && NZ(RL10,2) == 0,
            2916$,Yes:
            Else,2917$,Yes;
2916$   ASSIGN:    CheckGapL7500Speed55_75t.NumberOut
True=CheckGapL7500Speed55_75t.NumberOut True + 1:NEXT(815$);

2917$   ASSIGN:    CheckGapL7500Speed55_75t.NumberOut
False=CheckGapL7500Speed55_75t.NumberOut False + 1:NEXT(814$);

815$   TRANSPORT:  LAGVT,RStation7550;

814$   TRANSPORT:  LAGVT,LStation7550;

;
;
;   Model statements for module: Decide 1334
;
853$   BRANCH,    1:
        If,NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1)
&&NZ(RL13,2) && NZ(RL10,2) == 0,
            2918$,Yes:
            Else,2919$,Yes;
2918$   ASSIGN:    CheckGapL7500Speed35_55t.NumberOut
True=CheckGapL7500Speed35_55t.NumberOut True + 1:NEXT(816$);

2919$   ASSIGN:    CheckGapL7500Speed35_55t.NumberOut
False=CheckGapL7500Speed35_55t.NumberOut False + 1:NEXT(817$);

816$   TRANSPORT:  LAGVT,RStation7550;

817$   TRANSPORT:  LAGVT,LStation7550;

;
;
;   Model statements for module: Decide 1335
;
854$   BRANCH,    1:
        If,NZ(RL11,1) &&NZ(RL11,2) &&NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&
NZ(RL10,2) == 0,2920$,Yes:
            Else,2921$,Yes;
2920$   ASSIGN:    CheckGapL7500Speed10_35t.NumberOut
True=CheckGapL7500Speed10_35t.NumberOut True + 1:NEXT(818$);

2921$   ASSIGN:    CheckGapL7500Speed10_35t.NumberOut
False=CheckGapL7500Speed10_35t.NumberOut False + 1:NEXT(819$);

```

```

818$    TRANSPORT:   LAGVT,RStation7550;

819$    TRANSPORT:   LAGVT,LStation7550;

;
;
;   Model statements for module: Decide 1337
;
856$    BRANCH,      1:
                If,SpeedatRStation7450>75,857$,Yes:
                If,SpeedatRStation7450 > 55 && SpeedatRStation7450<= 75,858$,Yes:
                If,SpeedatRStation7450 >35 && SpeedatRStation7450 <=55,859$,Yes:
                If,SpeedatRStation7450 >10 && SpeedatRStation7450 <=35,860$,Yes:
                Else,861$,Yes;

;
;
;   Model statements for module: Decide 1342
;
861$    BRANCH,      1:
                If,NZ(RL9,3)&&NZ(RL9,4) &&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1) ==
0,2924$,Yes:
                Else,2925$,Yes;
2924$    ASSIGN:     CheckGapL7500Speed10lesst.NumberOut
True=CheckGapL7500Speed10lesst.NumberOut True + 1:NEXT(870$);

2925$    ASSIGN:     CheckGapL7500Speed10lesst.NumberOut
False=CheckGapL7500Speed10lesst.NumberOut False + 1:NEXT(871$);

870$    TRANSPORT:   LAGVT,RStation7550;

871$    TRANSPORT:   LAGVT,LStation7550;

;
;
;   Model statements for module: Decide 1338
;
857$    BRANCH,      1:
                If,
                NZ(RL8,3) &&NZ(RL8,4) && NZ(RL9,1) &&NZ(RL9,2)&&NZ(RL9,3)&&NZ(RL9,4)
&&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2) == 0,
                2926$,Yes:
                Else,2927$,Yes;
2926$    ASSIGN:     CheckGapL7500Speed75lesst.NumberOut
True=CheckGapL7500Speed75lesst.NumberOut True + 1:NEXT(862$);

2927$    ASSIGN:     CheckGapL7500Speed75lesst.NumberOut
False=CheckGapL7500Speed75lesst.NumberOut False + 1:NEXT(863$);

862$    TRANSPORT:   LAGVT,RStation7550;

```

```

863$      TRANSPORT:   LAGVT,LStation7550;

;
;
;   Model statements for module: Decide 1339
;
858$      BRANCH,      1:
                If,
                NZ(RL8,4) && NZ(RL9,1) &&NZ(RL9,2)&&NZ(RL9,3)&&NZ(RL9,4)
&&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1) == 0,
                2928$,Yes:
                Else,2929$,Yes;
2928$      ASSIGN:      CheckGapL7500Speed55_75lesst.NumberOut
True=CheckGapL7500Speed55_75lesst.NumberOut True + 1
                :NEXT(864$);

2929$      ASSIGN:      CheckGapL7500Speed55_75lesst.NumberOut
False=CheckGapL7500Speed55_75lesst.NumberOut False + 1
                :NEXT(865$);

864$      TRANSPORT:   LAGVT,RStation7550;

865$      TRANSPORT:   LAGVT,LStation7550;

;
;
;   Model statements for module: Decide 1340
;
859$      BRANCH,      1:
                If,NZ(RL9,1) &&NZ(RL9,2)&&NZ(RL9,3)&&NZ(RL9,4)
&&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1) == 0,2930$,Yes:
                Else,2931$,Yes;
2930$      ASSIGN:      CheckGapL7500Speed35_55lesst.NumberOut
True=CheckGapL7500Speed35_55lesst.NumberOut True + 1
                :NEXT(866$);

2931$      ASSIGN:      CheckGapL7500Speed35_55lesst.NumberOut
False=CheckGapL7500Speed35_55lesst.NumberOut False + 1
                :NEXT(867$);

866$      TRANSPORT:   LAGVT,RStation7550;

867$      TRANSPORT:   LAGVT,LStation7550;

;
;
;   Model statements for module: Decide 1341
;
860$      BRANCH,      1:

```

```

                If,NZ(RL9,2)&&NZ(RL9,3)&&NZ(RL9,4)
&&NZ(RL10,1)&&NZ(RL10,2)&&NZ(RL11,1) == 0,2932$,Yes:
                Else,2933$,Yes;
2932$    ASSIGN:    CheckGapL7500Speed10_35lesst.NumberOut
True=CheckGapL7500Speed10_35lesst.NumberOut True + 1
                :NEXT(868$);

2933$    ASSIGN:    CheckGapL7500Speed10_35lesst.NumberOut
False=CheckGapL7500Speed10_35lesst.NumberOut False + 1
                :NEXT(869$);

868$    TRANSPORT:    LAGVT,RStation7550;

869$    TRANSPORT:    LAGVT,LStation7550;

;
;
;    Model statements for module: Station 106
;

872$    STATION,    LStation7550;
2936$    DELAY:    0.0,,VA:NEXT(873$);

;
;
;    Model statements for module: Decide 1343
;
873$    BRANCH,    1:
                If,Entity.Type==LCar,2937$,Yes:
                Else,2938$,Yes;
2937$    ASSIGN:    checkvehicletype7550.NumberOut True=checkvehicletype7550.NumberOut
True + 1:NEXT(874$);

2938$    ASSIGN:    checkvehicletype7550.NumberOut False=checkvehicletype7550.NumberOut
False + 1:NEXT(875$);

;
;
;    Model statements for module: Decide 1344
;
874$    BRANCH,    1:
                If,SpeedatRStation7500 >= VTU(LAGV,LCar#),2939$,Yes:
                Else,2940$,Yes;
2939$    ASSIGN:    Decide 1344.NumberOut True=Decide 1344.NumberOut True + 1:NEXT(876$);

2940$    ASSIGN:    Decide 1344.NumberOut False=Decide 1344.NumberOut False +
1:NEXT(902$);

;

```

```

;
;   Model statements for module: Decide 1346
;
876$   BRANCH,      1:
        If,SpeedatRStation7500>75,877$,Yes:
        If,SpeedatRStation7500 > 55 && SpeedatRStation7500 <= 75,890$,Yes:
        If,SpeedatRStation7500 >35 && SpeedatRStation7500<=55,893$,Yes:
        If,SpeedatRStation7500 >10 && SpeedatRStation7500 <=35,896$,Yes:
        Else,899$,Yes;

;
;
;   Model statements for module: Decide 1351
;
899$   BRANCH,      1:
        If,NZ(RL12,1) &&NZ(RL12,2) && NZ(RL11,2) == 0,2943$,Yes:
        Else,2944$,Yes;
2943$   ASSIGN:      CheckGapL7550Speed10.NumberOut
True=CheckGapL7550Speed10.NumberOut True + 1:NEXT(900$);

2944$   ASSIGN:      CheckGapL7550Speed10.NumberOut
False=CheckGapL7550Speed10.NumberOut False + 1:NEXT(901$);

900$   TRANSPORT:    LAGV,RStation7600;

901$   TRANSPORT:    LAGV,LStation7600;

;
;
;   Model statements for module: Decide 1347
;
877$   BRANCH,      1:
        If,
        NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1)
&&NZ(RL14,2) &&NL(RL15) && NZ(RL11,2) == 0,
        2945$,Yes:
        Else,2946$,Yes;
2945$   ASSIGN:      CheckGapL7550Speed75.NumberOut
True=CheckGapL7550Speed75.NumberOut True + 1:NEXT(878$);

2946$   ASSIGN:      CheckGapL7550Speed75.NumberOut
False=CheckGapL7550Speed75.NumberOut False + 1:NEXT(879$);

878$   TRANSPORT:    LAGV,RStation7600;

879$   TRANSPORT:    LAGV,LStation7600;

;
;
;   Model statements for module: Decide 1348
;

```



```

890$      BRANCH,      1:
                If,NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1)
&&NZ(RL14,2) && NZ(RL11,2) == 0,
                2947$,Yes:
                Else,2948$,Yes;
2947$      ASSIGN:      CheckGapL7550Speed55_75.NumberOut
True=CheckGapL7550Speed55_75.NumberOut True + 1:NEXT(891$);

2948$      ASSIGN:      CheckGapL7550Speed55_75.NumberOut
False=CheckGapL7550Speed55_75.NumberOut False + 1:NEXT(892$);

891$      TRANSPORT:    LAGV,RStation7600;

892$      TRANSPORT:    LAGV,LStation7600;

;
;
;   Model statements for module: Decide 1349
;
893$      BRANCH,      1:
                If,NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&
NZ(RL11,2) == 0,2949$,Yes:
                Else,2950$,Yes;
2949$      ASSIGN:      CheckGapL7550Speed35_55.NumberOut
True=CheckGapL7550Speed35_55.NumberOut True + 1:NEXT(894$);

2950$      ASSIGN:      CheckGapL7550Speed35_55.NumberOut
False=CheckGapL7550Speed35_55.NumberOut False + 1:NEXT(895$);

894$      TRANSPORT:    LAGV,RStation7600;

895$      TRANSPORT:    LAGV,LStation7600;

;
;
;   Model statements for module: Decide 1350
;
896$      BRANCH,      1:
                If,NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) && NZ(RL11,2) == 0,2951$,Yes:
                Else,2952$,Yes;
2951$      ASSIGN:      CheckGapL7550Speed10_35.NumberOut
True=CheckGapL7550Speed10_35.NumberOut True + 1:NEXT(897$);

2952$      ASSIGN:      CheckGapL7550Speed10_35.NumberOut
False=CheckGapL7550Speed10_35.NumberOut False + 1:NEXT(898$);

897$      TRANSPORT:    LAGV,RStation7600;

898$      TRANSPORT:    LAGV,LStation7600;

```

```

;
;
;   Model statements for module: Decide 1352
;
902$   BRANCH,      1:
        If,SpeedatRStation7500>75,903$,Yes:
        If,SpeedatRStation7500 > 55 && SpeedatRStation7500 <= 75,906$,Yes:
        If,SpeedatRStation7500>35 && SpeedatRStation7500 <=55,909$,Yes:
        If,SpeedatRStation7500 >10 && SpeedatRStation7500 <=35,912$,Yes:
        Else,915$,Yes;

;
;
;   Model statements for module: Decide 1357
;
915$   BRANCH,      1:
        If,NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1) == 0,2955$,Yes:
        Else,2956$,Yes;
2955$   ASSIGN:      CheckGapL7550Speed10less.NumberOut
True=CheckGapL7550Speed10less.NumberOut True + 1:NEXT(916$);

2956$   ASSIGN:      CheckGapL7550Speed10less.NumberOut
False=CheckGapL7550Speed10less.NumberOut False + 1:NEXT(917$);

916$   TRANSPORT:    LAGV,RStation7600;

917$   TRANSPORT:    LAGV,LStation7600;

;
;
;   Model statements for module: Decide 1353
;
903$   BRANCH,      1:
        If,
        NZ(RL9,2) && NZ(RL9,3) &&NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2)
&&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1)&&NZ(RL12,2) == 0,
        2957$,Yes:
        Else,2958$,Yes;
2957$   ASSIGN:      CheckGapL7550Speed75less.NumberOut
True=CheckGapL7550Speed75less.NumberOut True + 1:NEXT(904$);

2958$   ASSIGN:      CheckGapL7550Speed75less.NumberOut
False=CheckGapL7550Speed75less.NumberOut False + 1:NEXT(905$);

904$   TRANSPORT:    LAGV,RStation7600;

905$   TRANSPORT:    LAGV,LStation7600;

;
;
;   Model statements for module: Decide 1354

```

```

;
906$    BRANCH,    1:
          If,NZ(RL9,3) &&NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2)
&&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1) == 0,2959$,
          Yes:
          Else,2960$,Yes;
2959$    ASSIGN:    CheckGapL7550Speed55_75less.NumberOut
True=CheckGapL7550Speed55_75less.NumberOut True + 1
          :NEXT(907$);

2960$    ASSIGN:    CheckGapL7550Speed55_75less.NumberOut
False=CheckGapL7550Speed55_75less.NumberOut False + 1
          :NEXT(908$);

907$    TRANSPORT:  LAGV,RStation7600;

908$    TRANSPORT:  LAGV,LStation7600;

;
;
;    Model statements for module: Decide 1355
;
909$    BRANCH,    1:
          If,NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2)
&&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1) == 0,2961$,Yes:
          Else,2962$,Yes;
2961$    ASSIGN:    CheckGapL7550Speed35_55less.NumberOut
True=CheckGapL7550Speed35_55less.NumberOut True + 1
          :NEXT(910$);

2962$    ASSIGN:    CheckGapL7550Speed35_55less.NumberOut
False=CheckGapL7550Speed35_55less.NumberOut False + 1
          :NEXT(911$);

910$    TRANSPORT:  LAGV,RStation7600;

911$    TRANSPORT:  LAGV,LStation7600;

;
;
;    Model statements for module: Decide 1356
;
912$    BRANCH,    1:
          If,NZ(RL10,2) &&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1) == 0,2963$,Yes:
          Else,2964$,Yes;
2963$    ASSIGN:    CheckGapL7550Speed10_35less.NumberOut
True=CheckGapL7550Speed10_35less.NumberOut True + 1
          :NEXT(913$);

2964$    ASSIGN:    CheckGapL7550Speed10_35less.NumberOut
False=CheckGapL7550Speed10_35less.NumberOut False + 1

```

```

: NEXT(914$);

913$    TRANSPORT:   LAGV,RStation7600;

914$    TRANSPORT:   LAGV,LStation7600;

;
;
;   Model statements for module: Decide 1345
;
875$    BRANCH,      1:
                If,SpeedatRStation7500 >= VTU(LAGVT,LTruck#),2965$,Yes:
                Else,2966$,Yes;
2965$    ASSIGN:      Decide 1345.NumberOut True=Decide 1345.NumberOut True + 1:NEXT(918$);

2966$    ASSIGN:      Decide 1345.NumberOut False=Decide 1345.NumberOut False +
1:NEXT(924$);

;
;
;   Model statements for module: Decide 1358
;
918$    BRANCH,      1:
                If,SpeedatRStation7500>75,919$,Yes:
                If,SpeedatRStation7500 > 55 && SpeedatRStation7500 <= 75,920$,Yes:
                If,SpeedatRStation7500 >35 && SpeedatRStation7500 <=55,921$,Yes:
                If,SpeedatRStation7500 >10 && SpeedatRStation7500 <=35,922$,Yes:
                Else,923$,Yes;

;
;
;   Model statements for module: Decide 1363
;
923$    BRANCH,      1:
                If,NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL11,2) ==
0,2969$,Yes:
                Else,2970$,Yes;
2969$    ASSIGN:      CheckGapL7550Speed10t.NumberOut
True=CheckGapL7550Speed10t.NumberOut True + 1:NEXT(888$);

2970$    ASSIGN:      CheckGapL7550Speed10t.NumberOut
False=CheckGapL7550Speed10t.NumberOut False + 1:NEXT(889$);

888$    TRANSPORT:   LAGVT,RStation7600;

889$    TRANSPORT:   LAGVT,LStation7600;

;
;
;   Model statements for module: Decide 1359

```

```

;
919$    BRANCH,    1:
        If,
            NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1)
&&NZ(RL14,2) &&NL(RL15) && NL(RL16) &&NZ(RL11,2) == 0,
            2971$,Yes:
            Else,2972$,Yes;
2971$    ASSIGN:    CheckGapL7550Speed75t.NumberOut
True=CheckGapL7550Speed75t.NumberOut True + 1:NEXT(880$);

2972$    ASSIGN:    CheckGapL7550Speed75t.NumberOut
False=CheckGapL7550Speed75t.NumberOut False + 1:NEXT(881$);

880$    TRANSPORT:  LAGVT,RStation7600;

881$    TRANSPORT:  LAGVT,LStation7600;

;
;
;    Model statements for module: Decide 1360
;
920$    BRANCH,    1:
        If,
            NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1)
&&NZ(RL14,2) &&NL(RL15) &&NZ(RL11,2) == 0,
            2973$,Yes:
            Else,2974$,Yes;
2973$    ASSIGN:    CheckGapL7550Speed55_75t.NumberOut
True=CheckGapL7550Speed55_75t.NumberOut True + 1:NEXT(883$);

2974$    ASSIGN:    CheckGapL7550Speed55_75t.NumberOut
False=CheckGapL7550Speed55_75t.NumberOut False + 1:NEXT(882$);

883$    TRANSPORT:  LAGVT,RStation7600;

882$    TRANSPORT:  LAGVT,LStation7600;

;
;
;    Model statements for module: Decide 1361
;
921$    BRANCH,    1:
        If,NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1)
&&NZ(RL14,2) &&NZ(RL11,2) == 0,
            2975$,Yes:
            Else,2976$,Yes;
2975$    ASSIGN:    CheckGapL7550Speed35_55t.NumberOut
True=CheckGapL7550Speed35_55t.NumberOut True + 1:NEXT(884$);

2976$    ASSIGN:    CheckGapL7550Speed35_55t.NumberOut
False=CheckGapL7550Speed35_55t.NumberOut False + 1:NEXT(885$);

```

```

884$    TRANSPORT:  LAGVT,RStation7600;

885$    TRANSPORT:  LAGVT,LStation7600;

;
;
;   Model statements for module: Decide 1362
;
922$    BRANCH,      1:
          If,NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1)
&&NZ(RL11,2) == 0,2977$,Yes:
          Else,2978$,Yes;
2977$    ASSIGN:      CheckGapL7550Speed10_35t.NumberOut
True=CheckGapL7550Speed10_35t.NumberOut True + 1:NEXT(886$);

2978$    ASSIGN:      CheckGapL7550Speed10_35t.NumberOut
False=CheckGapL7550Speed10_35t.NumberOut False + 1:NEXT(887$);

886$    TRANSPORT:  LAGVT,RStation7600;

887$    TRANSPORT:  LAGVT,LStation7600;

;
;
;   Model statements for module: Decide 1364
;
924$    BRANCH,      1:
          If,SpeedatRStation7500>75,925$,Yes:
          If,SpeedatRStation7500 > 55 && SpeedatRStation7500<= 75,926$,Yes:
          If,SpeedatRStation7500 >35 && SpeedatRStation7500 <=55,927$,Yes:
          If,SpeedatRStation7500 >10 && SpeedatRStation7500 <=35,928$,Yes:
          Else,929$,Yes;

;
;
;   Model statements for module: Decide 1369
;
929$    BRANCH,      1:
          If,NZ(RL10,1)&&NZ(RL10,2) &&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1) ==
0,2981$,Yes:
          Else,2982$,Yes;
2981$    ASSIGN:      CheckGapL7550Speed10lesst.NumberOut
True=CheckGapL7550Speed10lesst.NumberOut True + 1:NEXT(938$);

2982$    ASSIGN:      CheckGapL7550Speed10lesst.NumberOut
False=CheckGapL7550Speed10lesst.NumberOut False + 1:NEXT(939$);

938$    TRANSPORT:  LAGVT,RStation7600;

939$    TRANSPORT:  LAGVT,LStation7600;

```

```

;
;
;   Model statements for module: Decide 1365
;
925$   BRANCH,      1:
        If,
            NZ(RL9,1) &&NZ(RL9,2) && NZ(RL9,3)
&&NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2)
&&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1)&&NZ(RL12,2) == 0,
            2983$,Yes:
                Else,2984$,Yes;
2983$   ASSIGN:      CheckGapL7550Speed75lesst.NumberOut
True=CheckGapL7550Speed75lesst.NumberOut True + 1:NEXT(930$);

2984$   ASSIGN:      CheckGapL7550Speed75lesst.NumberOut
False=CheckGapL7550Speed75lesst.NumberOut False + 1:NEXT(931$);

930$   TRANSPORT:    LAGVT,RStation7600;

931$   TRANSPORT:    LAGVT,LStation7600;

;
;
;   Model statements for module: Decide 1366
;
926$   BRANCH,      1:
        If,
            NZ(RL9,2) && NZ(RL9,3) &&NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2)
&&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1) == 0,
            2985$,Yes:
                Else,2986$,Yes;
2985$   ASSIGN:      CheckGapL7550Speed55_75lesst.NumberOut
True=CheckGapL7550Speed55_75lesst.NumberOut True + 1
:NEXT(932$);

2986$   ASSIGN:      CheckGapL7550Speed55_75lesst.NumberOut
False=CheckGapL7550Speed55_75lesst.NumberOut False + 1
:NEXT(933$);

932$   TRANSPORT:    LAGVT,RStation7600;

933$   TRANSPORT:    LAGVT,LStation7600;

;
;
;   Model statements for module: Decide 1367
;
927$   BRANCH,      1:

```

```

                If,NZ(RL9,3) &&NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2)
&&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1) == 0,2987$,

```

```

    Yes:

```

```

    Else,2988$,Yes;

```

```

2987$    ASSIGN:    CheckGapL7550Speed35_55lesst.NumberOut
True=CheckGapL7550Speed35_55lesst.NumberOut True + 1
        :NEXT(934$);

```

```

2988$    ASSIGN:    CheckGapL7550Speed35_55lesst.NumberOut
False=CheckGapL7550Speed35_55lesst.NumberOut False + 1
        :NEXT(935$);

```

```

934$    TRANSPORT:  LAGVT,RStation7600;

```

```

935$    TRANSPORT:  LAGVT,LStation7600;

```

```

;
;
;
;

```

```

; Model statements for module: Decide 1368

```

```

928$    BRANCH,    1:
                If,NZ(RL9,4)&&NZ(RL10,1)&&NZ(RL10,2)
&&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1) == 0,2989$,Yes:
                Else,2990$,Yes;

```

```

2989$    ASSIGN:    CheckGapL7550Speed10_35lesst.NumberOut
True=CheckGapL7550Speed10_35lesst.NumberOut True + 1
        :NEXT(936$);

```

```

2990$    ASSIGN:    CheckGapL7550Speed10_35lesst.NumberOut
False=CheckGapL7550Speed10_35lesst.NumberOut False + 1
        :NEXT(937$);

```

```

936$    TRANSPORT:  LAGVT,RStation7600;

```

```

937$    TRANSPORT:  LAGVT,LStation7600;

```

```

;
;
;
;

```

```

; Model statements for module: Station 107

```

```

940$    STATION,    RStation7600;

```

```

2993$    DELAY:      0.0,,VA:NEXT(960$);

```

```

;
;
;
;

```

```

; Model statements for module: Decide 1376

```

```

960$    BRANCH,    1:
                If,Entity.Type==RCar,956$,Yes:

```



```

    If,Entity.Type==RTruck,957$,Yes:
    If,Entity.Type==LCar,958$,Yes:
    Else,959$,Yes;

;
;
;   Model statements for module: Assign 162
;
959$   ASSIGN:   SpeedatRStation7600=VTU(LAGVT,LTruck#):NEXT(943$);

;
;
;   Model statements for module: Decide 1370
;
943$   BRANCH,   1:
        If,Entity.Type==RCar,2996$,Yes:
        Else,2997$,Yes;
2996$   ASSIGN:   Decide 1370.NumberOut True=Decide 1370.NumberOut True + 1:NEXT(942$);
2997$   ASSIGN:   Decide 1370.NumberOut False=Decide 1370.NumberOut False +
1:NEXT(945$);
942$   TRANSPORT:  RAGV,RStation7650;

;
;
;   Model statements for module: Decide 1371
;
945$   BRANCH,   1:
        If,Entity.Type==RTruck,2998$,Yes:
        Else,2999$,Yes;
2998$   ASSIGN:   Decide 1371.NumberOut True=Decide 1371.NumberOut True + 1:NEXT(944$);
2999$   ASSIGN:   Decide 1371.NumberOut False=Decide 1371.NumberOut False +
1:NEXT(946$);
944$   TRANSPORT:  RAGVT,RStation7650;

;
;
;   Model statements for module: Decide 1372
;
946$   BRANCH,   1:
        If,Entity.Type==LCar,3000$,Yes:
        Else,3001$,Yes;
3000$   ASSIGN:   Decide 1372.NumberOut True=Decide 1372.NumberOut True + 1:NEXT(947$);
3001$   ASSIGN:   Decide 1372.NumberOut False=Decide 1372.NumberOut False +
1:NEXT(948$);

```

```

947$    TRANSPORT:   LAGV,RStation7650;

948$    TRANSPORT:   LAGVT,RStation7650;

;
;
;   Model statements for module: Assign 159
;
956$    ASSIGN:      SpeedatRStation7600=VTU(RAGV,RCar#):NEXT(943$);

;
;
;   Model statements for module: Assign 160
;
957$    ASSIGN:      SpeedatRStation7600=VTU(RAGVT,RTruck#):NEXT(943$);

;
;
;   Model statements for module: Assign 161
;
958$    ASSIGN:      SpeedatRStation7600=VTU(LAGV,LCar#):NEXT(943$);

;
;
;   Model statements for module: Station 108
;

941$    STATION,     RStation7650;
3004$   DELAY:       0.0,,VA:NEXT(965$);

;
;
;   Model statements for module: Decide 1377
;
965$    BRANCH,      1:
                If,Entity.Type==RCar,961$,Yes:
                If,Entity.Type==RTruck,962$,Yes:
                If,Entity.Type==LCar,963$,Yes:
                Else,964$,Yes;

;
;
;   Model statements for module: Assign 166
;
964$    ASSIGN:      SpeedatRStation7650=VTU(LAGVT,LTruck#):NEXT(950$);

;

```

```

;
;   Model statements for module: Decide 1373
;
950$   BRANCH,      1:
        If,Entity.Type==RCar,3007$,Yes:
        Else,3008$,Yes;
3007$   ASSIGN:      Decide 1373.NumberOut True=Decide 1373.NumberOut True + 1:NEXT(949$);

3008$   ASSIGN:      Decide 1373.NumberOut False=Decide 1373.NumberOut False +
1:NEXT(952$);

949$   TRANSPORT:    RAGV,RStation7700;

;
;
;   Model statements for module: Decide 1374
;
952$   BRANCH,      1:
        If,Entity.Type==RTruck,3009$,Yes:
        Else,3010$,Yes;
3009$   ASSIGN:      Decide 1374.NumberOut True=Decide 1374.NumberOut True + 1:NEXT(951$);

3010$   ASSIGN:      Decide 1374.NumberOut False=Decide 1374.NumberOut False +
1:NEXT(953$);

951$   TRANSPORT:    RAGVT,RStation7700;

;
;
;   Model statements for module: Decide 1375
;
953$   BRANCH,      1:
        If,Entity.Type==LCar,3011$,Yes:
        Else,3012$,Yes;
3011$   ASSIGN:      Decide 1375.NumberOut True=Decide 1375.NumberOut True + 1:NEXT(954$);

3012$   ASSIGN:      Decide 1375.NumberOut False=Decide 1375.NumberOut False +
1:NEXT(955$);

954$   TRANSPORT:    LAGV,RStation7700;

955$   TRANSPORT:    LAGVT,RStation7700;

;
;
;   Model statements for module: Assign 163
;
961$   ASSIGN:      SpeedatRStation7650=VTU(RAGV,RCar#):NEXT(950$);

```

```

;
;
; Model statements for module: Assign 164
;
962$    ASSIGN:    SpeedatRStation7650=VTU(RAGVT,RTruck#):NEXT(950$);

;
;
; Model statements for module: Assign 165
;
963$    ASSIGN:    SpeedatRStation7650=VTU(LAGV,LCar#):NEXT(950$);

;
;
; Model statements for module: Station 109
;
966$    STATION,    LStation7600;
3015$    DELAY:    0.0,,VA:NEXT(967$);

;
;
; Model statements for module: Decide 1378
;
967$    BRANCH,    1:
                If,Entity.Type==LCar,3016$,Yes:
                Else,3017$,Yes;
3016$    ASSIGN:    checkvehicletype7600.NumberOut True=checkvehicletype7600.NumberOut
True + 1:NEXT(968$);

3017$    ASSIGN:    checkvehicletype7600.NumberOut False=checkvehicletype7600.NumberOut
False + 1:NEXT(969$);

;
;
; Model statements for module: Decide 1379
;
968$    BRANCH,    1:
                If,SpeedatRStation7550 >= VTU(LAGV,LCar#),3018$,Yes:
                Else,3019$,Yes;
3018$    ASSIGN:    Decide 1379.NumberOut True=Decide 1379.NumberOut True + 1:NEXT(970$);

3019$    ASSIGN:    Decide 1379.NumberOut False=Decide 1379.NumberOut False +
1:NEXT(996$);

;
;
; Model statements for module: Decide 1381

```

```

;
970$    BRANCH,      1:
        If,SpeedatRStation7550>75,971$,Yes:
        If,SpeedatRStation7550 > 55 && SpeedatRStation7550 <= 75,984$,Yes:
        If,SpeedatRStation7550 >35 && SpeedatRStation7550<=55,987$,Yes:
        If,SpeedatRStation7550 >10 && SpeedatRStation7550 <=35,990$,Yes:
        Else,993$,Yes;

;
;
;   Model statements for module: Decide 1386
;
993$    BRANCH,      1:
        If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL12,2) == 0,3022$,Yes:
        Else,3023$,Yes;
3022$    ASSIGN:      CheckGapL7600Speed10.NumberOut
True=CheckGapL7600Speed10.NumberOut True + 1:NEXT(994$);

3023$    ASSIGN:      CheckGapL7600Speed10.NumberOut
False=CheckGapL7600Speed10.NumberOut False + 1:NEXT(995$);

994$    TRANSPORT:    LAGV,RStation7650;

995$    TRANSPORT:    LAGV,LStation7650;

;
;
;   Model statements for module: Decide 1382
;
971$    BRANCH,      1:
        If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL14,2)
&&NL(RL15)&&NL(RL16) && NZ(RL12,2) == 0,3024$,
        Yes:
        Else,3025$,Yes;
3024$    ASSIGN:      CheckGapL7600Speed75.NumberOut
True=CheckGapL7600Speed75.NumberOut True + 1:NEXT(972$);

3025$    ASSIGN:      CheckGapL7600Speed75.NumberOut
False=CheckGapL7600Speed75.NumberOut False + 1:NEXT(973$);

972$    TRANSPORT:    LAGV,RStation7650;

973$    TRANSPORT:    LAGV,LStation7650;

;
;
;   Model statements for module: Decide 1383
;
984$    BRANCH,      1:
        If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL14,2)
&&NL(RL15)&&NZ(RL12,2) == 0,3026$,Yes:

```

```

Else,3027$,Yes;
3026$    ASSIGN:    CheckGapL7600Speed55_75.NumberOut
True=CheckGapL7600Speed55_75.NumberOut True + 1:NEXT(985$);

3027$    ASSIGN:    CheckGapL7600Speed55_75.NumberOut
False=CheckGapL7600Speed55_75.NumberOut False + 1:NEXT(986$);

985$     TRANSPORT:  LAGV,RStation7650;

986$     TRANSPORT:  LAGV,LStation7650;

;
;
;   Model statements for module: Decide 1384
;
987$     BRANCH,     1:
                If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL14,2) &&NZ(RL12,2) ==
0,3028$,Yes:
                Else,3029$,Yes;
3028$     ASSIGN:    CheckGapL7600Speed35_55.NumberOut
True=CheckGapL7600Speed35_55.NumberOut True + 1:NEXT(988$);

3029$     ASSIGN:    CheckGapL7600Speed35_55.NumberOut
False=CheckGapL7600Speed35_55.NumberOut False + 1:NEXT(989$);

988$     TRANSPORT:  LAGV,RStation7650;

989$     TRANSPORT:  LAGV,LStation7650;

;
;
;   Model statements for module: Decide 1385
;
990$     BRANCH,     1:
                If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL12,2) == 0,3030$,Yes:
                Else,3031$,Yes;
3030$     ASSIGN:    CheckGapL7600Speed10_35.NumberOut
True=CheckGapL7600Speed10_35.NumberOut True + 1:NEXT(991$);

3031$     ASSIGN:    CheckGapL7600Speed10_35.NumberOut
False=CheckGapL7600Speed10_35.NumberOut False + 1:NEXT(992$);

991$     TRANSPORT:  LAGV,RStation7650;

992$     TRANSPORT:  LAGV,LStation7650;

;
;
;   Model statements for module: Decide 1387
;

```

```

996$      BRANCH,      1:
          If,SpeedatRStation7550>75,997$,Yes:
          If,SpeedatRStation7550 > 55 && SpeedatRStation7550 <= 75,1000$,Yes:
          If,SpeedatRStation7550>35 && SpeedatRStation7550 <=55,1003$,Yes:
          If,SpeedatRStation7550 >10 && SpeedatRStation7550 <=35,1006$,Yes:
          Else,1009$,Yes;

;
;
;   Model statements for module: Decide 1392
;
1009$      BRANCH,      1:
          If,NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)== 0,3034$,Yes:
          Else,3035$,Yes;
3034$      ASSIGN:      CheckGapL7600Speed10less.NumberOut
True=CheckGapL7600Speed10less.NumberOut True + 1:NEXT(1010$);

3035$      ASSIGN:      CheckGapL7600Speed10less.NumberOut
False=CheckGapL7600Speed10less.NumberOut False + 1:NEXT(1011$);

1010$      TRANSPORT:   LAGV,RStation7650;

1011$      TRANSPORT:   LAGV,LStation7650;

;
;
;   Model statements for module: Decide 1388
;
997$      BRANCH,      1:
          If,
          NZ(RL9,4) && NZ(RL10,1) &&NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2)
&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2) == 0,
          3036$,Yes:
          Else,3037$,Yes;
3036$      ASSIGN:      CheckGapL7600Speed75less.NumberOut
True=CheckGapL7600Speed75less.NumberOut True + 1:NEXT(998$);

3037$      ASSIGN:      CheckGapL7600Speed75less.NumberOut
False=CheckGapL7600Speed75less.NumberOut False + 1:NEXT(999$);

998$      TRANSPORT:   LAGV,RStation7650;

999$      TRANSPORT:   LAGV,LStation7650;

;
;
;   Model statements for module: Decide 1389
;
1000$      BRANCH,      1:
          If,NZ(RL10,1) &&NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2)
&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)== 0,3038$,

```

```

                Yes:
                Else,3039$,Yes;
3038$    ASSIGN:    CheckGapL7600Speed55_75less.NumberOut
True=CheckGapL7600Speed55_75less.NumberOut True + 1
                :NEXT(1001$);

3039$    ASSIGN:    CheckGapL7600Speed55_75less.NumberOut
False=CheckGapL7600Speed55_75less.NumberOut False + 1
                :NEXT(1002$);

1001$    TRANSPORT:    LAGV,RStation7650;

1002$    TRANSPORT:    LAGV,LStation7650;

;
;
;    Model statements for module: Decide 1390
;
1003$    BRANCH,    1:
                If,NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2)
&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)== 0,3040$,Yes:
                Else,3041$,Yes;
3040$    ASSIGN:    CheckGapL7600Speed35_55less.NumberOut
True=CheckGapL7600Speed35_55less.NumberOut True + 1
                :NEXT(1004$);

3041$    ASSIGN:    CheckGapL7600Speed35_55less.NumberOut
False=CheckGapL7600Speed35_55less.NumberOut False + 1
                :NEXT(1005$);

1004$    TRANSPORT:    LAGV,RStation7650;

1005$    TRANSPORT:    LAGV,LStation7650;

;
;
;    Model statements for module: Decide 1391
;
1006$    BRANCH,    1:
                If,NZ(RL11,2) &&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)== 0,3042$,Yes:
                Else,3043$,Yes;
3042$    ASSIGN:    CheckGapL7600Speed10_35less.NumberOut
True=CheckGapL7600Speed10_35less.NumberOut True + 1
                :NEXT(1007$);

3043$    ASSIGN:    CheckGapL7600Speed10_35less.NumberOut
False=CheckGapL7600Speed10_35less.NumberOut False + 1
                :NEXT(1008$);

1007$    TRANSPORT:    LAGV,RStation7650;

```



```

1008$    TRANSPORT:   LAGV,LStation7650;

;
;
;   Model statements for module: Decide 1380
;
969$    BRANCH,      1:
          If,SpeedatRStation7550 >= VTU(LAGVT,LTruck#),3044$,Yes:
          Else,3045$,Yes;
3044$    ASSIGN:      Decide 1380.NumberOut True=Decide 1380.NumberOut True +
1:NEXT(1012$);

3045$    ASSIGN:      Decide 1380.NumberOut False=Decide 1380.NumberOut False +
1:NEXT(1018$);

;
;
;   Model statements for module: Decide 1393
;
1012$    BRANCH,      1:
          If,SpeedatRStation7550>75,1013$,Yes:
          If,SpeedatRStation7550 > 55 && SpeedatRStation7550 <= 75,1014$,Yes:
          If,SpeedatRStation7550 >35 && SpeedatRStation7550 <=55,1015$,Yes:
          If,SpeedatRStation7550 >10 && SpeedatRStation7550 <=35,1016$,Yes:
          Else,1017$,Yes;

;
;
;   Model statements for module: Decide 1398
;
1017$    BRANCH,      1:
          If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL14,2) && NZ(RL12,2) ==
0,3048$,Yes:
          Else,3049$,Yes;
3048$    ASSIGN:      CheckGapL7600Speed10t.NumberOut
True=CheckGapL7600Speed10t.NumberOut True + 1:NEXT(982$);

3049$    ASSIGN:      CheckGapL7600Speed10t.NumberOut
False=CheckGapL7600Speed10t.NumberOut False + 1:NEXT(983$);

982$    TRANSPORT:   LAGVT,RStation7650;

983$    TRANSPORT:   LAGVT,LStation7650;

;
;
;   Model statements for module: Decide 1394
;
1013$    BRANCH,      1:
          If,

```

```

        NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL14,2)
&&NL(RL15)&&NL(RL16) &&NL(RL17)&& NZ(RL12,2) == 0,
        3050$,Yes;
        Else,3051$,Yes;
3050$    ASSIGN:    CheckGapL7600Speed75t.NumberOut
True=CheckGapL7600Speed75t.NumberOut True + 1:NEXT(974$);

3051$    ASSIGN:    CheckGapL7600Speed75t.NumberOut
False=CheckGapL7600Speed75t.NumberOut False + 1:NEXT(975$);

974$    TRANSPORT:  LAGVT,RStation7650;

975$    TRANSPORT:  LAGVT,LStation7650;

;
;
;   Model statements for module: Decide 1395
;
1014$    BRANCH,    1:
        If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL14,2)
&&NL(RL15)&&NL(RL16) && NZ(RL12,2) == 0,3052$,
        Yes:
        Else,3053$,Yes;
3052$    ASSIGN:    CheckGapL7600Speed55_75t.NumberOut
True=CheckGapL7600Speed55_75t.NumberOut True + 1:NEXT(977$);

3053$    ASSIGN:    CheckGapL7600Speed55_75t.NumberOut
False=CheckGapL7600Speed55_75t.NumberOut False + 1:NEXT(976$);

977$    TRANSPORT:  LAGVT,RStation7650;

976$    TRANSPORT:  LAGVT,LStation7650;

;
;
;   Model statements for module: Decide 1396
;
1015$    BRANCH,    1:
        If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&&
NZ(RL12,2) == 0,3054$,Yes:
        Else,3055$,Yes;
3054$    ASSIGN:    CheckGapL7600Speed35_55t.NumberOut
True=CheckGapL7600Speed35_55t.NumberOut True + 1:NEXT(978$);

3055$    ASSIGN:    CheckGapL7600Speed35_55t.NumberOut
False=CheckGapL7600Speed35_55t.NumberOut False + 1:NEXT(979$);

978$    TRANSPORT:  LAGVT,RStation7650;

979$    TRANSPORT:  LAGVT,LStation7650;

```

```

;
;
;   Model statements for module: Decide 1397
;
1016$   BRANCH,      1:
                If,NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) &&NZ(RL14,2) && NZ(RL12,2) ==
0,3056$,Yes:
                Else,3057$,Yes;
3056$   ASSIGN:      CheckGapL7600Speed10_35t.NumberOut
True=CheckGapL7600Speed10_35t.NumberOut True + 1:NEXT(980$);

3057$   ASSIGN:      CheckGapL7600Speed10_35t.NumberOut
False=CheckGapL7600Speed10_35t.NumberOut False + 1:NEXT(981$);

980$    TRANSPORT:   LAGVT,RStation7650;

981$    TRANSPORT:   LAGVT,LStation7650;

;
;
;   Model statements for module: Decide 1399
;
1018$   BRANCH,      1:
                If,SpeedatRStation7550>75,1019$,Yes:
                If,SpeedatRStation7550 > 55 && SpeedatRStation7550<= 75,1020$,Yes:
                If,SpeedatRStation7550 >35 && SpeedatRStation7550 <=55,1021$,Yes:
                If,SpeedatRStation7550 >10 && SpeedatRStation7550 <=35,1022$,Yes:
                Else,1023$,Yes;

;
;
;   Model statements for module: Decide 1404
;
1023$   BRANCH,      1:
                If,NZ(RL11,1)&&NZ(RL11,2) &&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1) ==
0,3060$,Yes:
                Else,3061$,Yes;
3060$   ASSIGN:      CheckGapL7600Speed10lesst.NumberOut
True=CheckGapL7600Speed10lesst.NumberOut True + 1:NEXT(1032$);

3061$   ASSIGN:      CheckGapL7600Speed10lesst.NumberOut
False=CheckGapL7600Speed10lesst.NumberOut False + 1:NEXT(1033$);

1032$   TRANSPORT:   LAGVT,RStation7650;

1033$   TRANSPORT:   LAGVT,LStation7650;

;
;
;   Model statements for module: Decide 1400

```

```

;
1019$    BRANCH,    1:
        If,
            NZ(RL9,2) && NZ(RL9,3) && NZ(RL9,4) && NZ(RL10,1)
&& NZ(RL10,2) && NZ(RL11,1) && NZ(RL11,2)
&& NZ(RL12,1) && NZ(RL12,2) && NZ(RL13,1) && NZ(RL13,2) == 0,
            3062$, Yes:
            Else, 3063$, Yes;
3062$    ASSIGN:    CheckGapL7600Speed75lesst.NumberOut
True=CheckGapL7600Speed75lesst.NumberOut True + 1:NEXT(1024$);

3063$    ASSIGN:    CheckGapL7600Speed75lesst.NumberOut
False=CheckGapL7600Speed75lesst.NumberOut False + 1:NEXT(1025$);

1024$    TRANSPORT:  LAGVT,RStation7650;

1025$    TRANSPORT:  LAGVT,LStation7650;

;
;
;    Model statements for module: Decide 1401
;
1020$    BRANCH,    1:
        If,
            NZ(RL9,3) && NZ(RL9,4) && NZ(RL10,1)
&& NZ(RL10,2) && NZ(RL11,1) && NZ(RL11,2) && NZ(RL12,1) && NZ(RL12,2) && NZ(RL13,1) == 0,
            3064$, Yes:
            Else, 3065$, Yes;
3064$    ASSIGN:    CheckGapL7600Speed55_75lesst.NumberOut
True=CheckGapL7600Speed55_75lesst.NumberOut True + 1
:NEXT(1026$);

3065$    ASSIGN:    CheckGapL7600Speed55_75lesst.NumberOut
False=CheckGapL7600Speed55_75lesst.NumberOut False + 1
:NEXT(1027$);

1026$    TRANSPORT:  LAGVT,RStation7650;

1027$    TRANSPORT:  LAGVT,LStation7650;

;
;
;    Model statements for module: Decide 1402
;
1021$    BRANCH,    1:
        If, NZ(RL10,1) && NZ(RL10,2) && NZ(RL11,1) && NZ(RL11,2)
&& NZ(RL12,1) && NZ(RL12,2) && NZ(RL13,1) == 0, 3066$,
            Yes:
            Else, 3067$, Yes;
3066$    ASSIGN:    CheckGapL7600Speed35_55lesst.NumberOut
True=CheckGapL7600Speed35_55lesst.NumberOut True + 1

```

```

: NEXT(1028$);

3067$    ASSIGN:    CheckGapL7600Speed35_55lesst.NumberOut
False=CheckGapL7600Speed35_55lesst.NumberOut False + 1
: NEXT(1029$);

1028$    TRANSPORT:  LAGVT,RStation7650;

1029$    TRANSPORT:  LAGVT,LStation7650;

;
;
;   Model statements for module: Decide 1403
;
1022$    BRANCH,    1:
            If,NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2)
&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1) == 0,3068$,Yes:
            Else,3069$,Yes;
3068$    ASSIGN:    CheckGapL7600Speed10_35lesst.NumberOut
True=CheckGapL7600Speed10_35lesst.NumberOut True + 1
: NEXT(1030$);

3069$    ASSIGN:    CheckGapL7600Speed10_35lesst.NumberOut
False=CheckGapL7600Speed10_35lesst.NumberOut False + 1
: NEXT(1031$);

1030$    TRANSPORT:  LAGVT,RStation7650;

1031$    TRANSPORT:  LAGVT,LStation7650;

;
;
;   Model statements for module: Station 110
;
1034$    STATION,    LStation7650;
3072$    DELAY:      0.0,,VA:NEXT(1035$);

;
;
;   Model statements for module: Decide 1405
;
1035$    BRANCH,    1:
            If,Entity.Type==LCar,3073$,Yes:
            Else,3074$,Yes;
3073$    ASSIGN:    checkvehicletype7650.NumberOut True=checkvehicletype7650.NumberOut
True + 1:NEXT(1036$);

3074$    ASSIGN:    checkvehicletype7650.NumberOut False=checkvehicletype7650.NumberOut
False + 1:NEXT(1037$);

```

```

;
;
;   Model statements for module: Decide 1406
;
1036$   BRANCH,      1:
        If,SpeedatRStation7600 >= VTU(LAGV,LCar#),3075$,Yes:
        Else,3076$,Yes;
3075$   ASSIGN:      Decide 1406.NumberOut True=Decide 1406.NumberOut True +
1:NEXT(1038$);

3076$   ASSIGN:      Decide 1406.NumberOut False=Decide 1406.NumberOut False +
1:NEXT(1064$);

;
;
;   Model statements for module: Decide 1408
;
1038$   BRANCH,      1:
        If,SpeedatRStation7600>75,1039$,Yes:
        If,SpeedatRStation7600 > 55 && SpeedatRStation7600 <= 75,1052$,Yes:
        If,SpeedatRStation7600 >35 && SpeedatRStation7600<=55,1055$,Yes:
        If,SpeedatRStation7600 >10 && SpeedatRStation7600 <=35,1058$,Yes:
        Else,1061$,Yes;

;
;
;   Model statements for module: Decide 1413
;
1061$   BRANCH,      1:
        If,NZ(RL14,1) &&NZ(RL14,2) &&NZ(RL13,2) == 0,3079$,Yes:
        Else,3080$,Yes;
3079$   ASSIGN:      CheckGapL7650Speed10.NumberOut
True=CheckGapL7650Speed10.NumberOut True + 1:NEXT(1062$);

3080$   ASSIGN:      CheckGapL7650Speed10.NumberOut
False=CheckGapL7650Speed10.NumberOut False + 1:NEXT(1063$);

1062$   TRANSPORT:   LAGV,RStation7700;

1063$   TRANSPORT:   LAGV,LStation7700;

;
;
;   Model statements for module: Decide 1409
;
1039$   BRANCH,      1:
        If,NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&&NL(RL16) &&NL(RL17)&&
NZ(RL13,2) == 0,3081$,Yes:
        Else,3082$,Yes;

```

```

3081$    ASSIGN:    CheckGapL7650Speed75.NumberOut
True=CheckGapL7650Speed75.NumberOut True + 1:NEXT(1040$);

3082$    ASSIGN:    CheckGapL7650Speed75.NumberOut
False=CheckGapL7650Speed75.NumberOut False + 1:NEXT(1041$);

1040$    TRANSPORT:  LAGV,RStation7700;

1041$    TRANSPORT:  LAGV,LStation7700;

;
;
;  Model statements for module: Decide 1410
;
1052$    BRANCH,    1:
                If,NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&&NL(RL16) &&NZ(RL13,2) ==
0,3083$,Yes:
                Else,3084$,Yes;
3083$    ASSIGN:    CheckGapL7650Speed55_75.NumberOut
True=CheckGapL7650Speed55_75.NumberOut True + 1:NEXT(1053$);

3084$    ASSIGN:    CheckGapL7650Speed55_75.NumberOut
False=CheckGapL7650Speed55_75.NumberOut False + 1:NEXT(1054$);

1053$    TRANSPORT:  LAGV,RStation7700;

1054$    TRANSPORT:  LAGV,LStation7700;

;
;
;  Model statements for module: Decide 1411
;
1055$    BRANCH,    1:
                If,NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&&NZ(RL13,2) == 0,3085$,Yes:
                Else,3086$,Yes;
3085$    ASSIGN:    CheckGapL7650Speed35_55.NumberOut
True=CheckGapL7650Speed35_55.NumberOut True + 1:NEXT(1056$);

3086$    ASSIGN:    CheckGapL7650Speed35_55.NumberOut
False=CheckGapL7650Speed35_55.NumberOut False + 1:NEXT(1057$);

1056$    TRANSPORT:  LAGV,RStation7700;

1057$    TRANSPORT:  LAGV,LStation7700;

;
;
;  Model statements for module: Decide 1412
;
1058$    BRANCH,    1:

```

```

                If,NZ(RL14,1) &&NZ(RL14,2)
&&NZ(RL15,1)&&NZ(RL15,2)&&NZ(RL15,3)&&NZ(RL15,4)&&NZ(RL13,2) == 0,3087$,
                Yes:
                Else,3088$,Yes;
3087$    ASSIGN:    CheckGapL7650Speed10_35.NumberOut
True=CheckGapL7650Speed10_35.NumberOut True + 1:NEXT(1059$);

3088$    ASSIGN:    CheckGapL7650Speed10_35.NumberOut
False=CheckGapL7650Speed10_35.NumberOut False + 1:NEXT(1060$);

1059$    TRANSPORT:    LAGV,RStation7700;

1060$    TRANSPORT:    LAGV,LStation7700;

;
;
;    Model statements for module: Decide 1414
;
1064$    BRANCH,    1:
                If,SpeedatRStation7600>75,1065$,Yes:
                If,SpeedatRStation7600 > 55 && SpeedatRStation7600 <= 75,1068$,Yes:
                If,SpeedatRStation7600>35 && SpeedatRStation7600 <=55,1071$,Yes:
                If,SpeedatRStation7600>10 && SpeedatRStation7600 <=35,1074$,Yes:
                Else,1077$,Yes;

;
;
;    Model statements for module: Decide 1419
;
1077$    BRANCH,    1:
                If,NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1) == 0,3091$,Yes:
                Else,3092$,Yes;
3091$    ASSIGN:    CheckGapL7650Speed10less.NumberOut
True=CheckGapL7650Speed10less.NumberOut True + 1:NEXT(1078$);

3092$    ASSIGN:    CheckGapL7650Speed10less.NumberOut
False=CheckGapL7650Speed10less.NumberOut False + 1:NEXT(1079$);

1078$    TRANSPORT:    LAGV,RStation7700;

1079$    TRANSPORT:    LAGV,LStation7700;

;
;
;    Model statements for module: Decide 1415
;
1065$    BRANCH,    1:
                If,
                NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2)
&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2) == 0,
                3093$,Yes:

```



```

Else,3094$,Yes;
3093$    ASSIGN:    CheckGapL7650Speed75less.NumberOut
True=CheckGapL7650Speed75less.NumberOut True + 1:NEXT(1066$);

3094$    ASSIGN:    CheckGapL7650Speed75less.NumberOut
False=CheckGapL7650Speed75less.NumberOut False + 1:NEXT(1067$);

1066$    TRANSPORT:  LAGV,RStation7700;

1067$    TRANSPORT:  LAGV,LStation7700;

;
;
;   Model statements for module: Decide 1416
;
1068$    BRANCH,    1:
                If,NZ(RL11,1)&&NZ(RL11,2)
&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1) == 0,3095$,
                Yes:
                Else,3096$,Yes;
3095$    ASSIGN:    CheckGapL7650Speed55_75less.NumberOut
True=CheckGapL7650Speed55_75less.NumberOut True + 1
                :NEXT(1069$);

3096$    ASSIGN:    CheckGapL7650Speed55_75less.NumberOut
False=CheckGapL7650Speed55_75less.NumberOut False + 1
                :NEXT(1070$);

1069$    TRANSPORT:  LAGV,RStation7700;

1070$    TRANSPORT:  LAGV,LStation7700;

;
;
;   Model statements for module: Decide 1417
;
1071$    BRANCH,    1:
                If,NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1) ==
0,3097$,Yes:
                Else,3098$,Yes;
3097$    ASSIGN:    CheckGapL7650Speed35_55less.NumberOut
True=CheckGapL7650Speed35_55less.NumberOut True + 1
                :NEXT(1072$);

3098$    ASSIGN:    CheckGapL7650Speed35_55less.NumberOut
False=CheckGapL7650Speed35_55less.NumberOut False + 1
                :NEXT(1073$);

1072$    TRANSPORT:  LAGV,RStation7700;

1073$    TRANSPORT:  LAGV,LStation7700;

```

```

;
;
;   Model statements for module: Decide 1418
;
1074$   BRANCH,      1:
          If,NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1) == 0,3099$,Yes:
          Else,3100$,Yes;
3099$   ASSIGN:      CheckGapL7650Speed10_35less.NumberOut
True=CheckGapL7650Speed10_35less.NumberOut True + 1
          :NEXT(1075$);

3100$   ASSIGN:      CheckGapL7650Speed10_35less.NumberOut
False=CheckGapL7650Speed10_35less.NumberOut False + 1
          :NEXT(1076$);

1075$   TRANSPORT:   LAGV,RStation7700;

1076$   TRANSPORT:   LAGV,LStation7700;

;
;
;   Model statements for module: Decide 1407
;
1037$   BRANCH,      1:
          If,SpeedatRStation7600 >= VTU(LAGVT,LTruck#),3101$,Yes:
          Else,3102$,Yes;
3101$   ASSIGN:      Decide 1407.NumberOut True=Decide 1407.NumberOut True +
1:NEXT(1080$);

3102$   ASSIGN:      Decide 1407.NumberOut False=Decide 1407.NumberOut False +
1:NEXT(1086$);

;
;
;   Model statements for module: Decide 1420
;
1080$   BRANCH,      1:
          If,SpeedatRStation7600>75,1081$,Yes:
          If,SpeedatRStation7600 > 55 && SpeedatRStation7600 <= 75,1082$,Yes:
          If,SpeedatRStation7600 >35 && SpeedatRStation7600 <=55,1083$,Yes:
          If,SpeedatRStation7600 >10 && SpeedatRStation7600 <=35,1084$,Yes:
          Else,1085$,Yes;

;
;
;   Model statements for module: Decide 1425
;
1085$   BRANCH,      1:
          If,NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&& NZ(RL13,2) == 0,3105$,Yes:

```

```

        Else,3106$,Yes;
3105$    ASSIGN:    CheckGapL7650Speed10t.NumberOut
True=CheckGapL7650Speed10t.NumberOut True + 1:NEXT(1050$);

3106$    ASSIGN:    CheckGapL7650Speed10t.NumberOut
False=CheckGapL7650Speed10t.NumberOut False + 1:NEXT(1051$);

1050$    TRANSPORT:  LAGVT,RStation7700;

1051$    TRANSPORT:  LAGVT,LStation7700;

;
;
;   Model statements for module: Decide 1421
;
1081$    BRANCH,    1:
                If,NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&&NL(RL16)
&&NL(RL17)&&NL(RL18)&& NZ(RL13,2) == 0,3107$,Yes:
                Else,3108$,Yes;
3107$    ASSIGN:    CheckGapL7650Speed75t.NumberOut
True=CheckGapL7650Speed75t.NumberOut True + 1:NEXT(1042$);

3108$    ASSIGN:    CheckGapL7650Speed75t.NumberOut
False=CheckGapL7650Speed75t.NumberOut False + 1:NEXT(1043$);

1042$    TRANSPORT:  LAGVT,RStation7700;

1043$    TRANSPORT:  LAGVT,LStation7700;

;
;
;   Model statements for module: Decide 1422
;
1082$    BRANCH,    1:
                If,NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&&NL(RL16) &&NL(RL17)&&
NZ(RL13,2) == 0,3109$,Yes:
                Else,3110$,Yes;
3109$    ASSIGN:    CheckGapL7650Speed55_75t.NumberOut
True=CheckGapL7650Speed55_75t.NumberOut True + 1:NEXT(1045$);

3110$    ASSIGN:    CheckGapL7650Speed55_75t.NumberOut
False=CheckGapL7650Speed55_75t.NumberOut False + 1:NEXT(1044$);

1045$    TRANSPORT:  LAGVT,RStation7700;

1044$    TRANSPORT:  LAGVT,LStation7700;

;
;
;   Model statements for module: Decide 1423

```

```

;
1083$    BRANCH,    1:
           If,NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&&NL(RL16) && NZ(RL13,2) ==
0,3111$,Yes:
           Else,3112$,Yes;
3111$    ASSIGN:    CheckGapL7650Speed35_55t.NumberOut
True=CheckGapL7650Speed35_55t.NumberOut True + 1:NEXT(1046$);

3112$    ASSIGN:    CheckGapL7650Speed35_55t.NumberOut
False=CheckGapL7650Speed35_55t.NumberOut False + 1:NEXT(1047$);

1046$    TRANSPORT:  LAGVT,RStation7700;

1047$    TRANSPORT:  LAGVT,LStation7700;

;
;
;    Model statements for module: Decide 1424
;
1084$    BRANCH,    1:
           If,NZ(RL14,1) &&NZ(RL14,2) &&NL(RL15)&& NZ(RL13,2) == 0,3113$,Yes:
           Else,3114$,Yes;
3113$    ASSIGN:    CheckGapL7650Speed10_35t.NumberOut
True=CheckGapL7650Speed10_35t.NumberOut True + 1:NEXT(1048$);

3114$    ASSIGN:    CheckGapL7650Speed10_35t.NumberOut
False=CheckGapL7650Speed10_35t.NumberOut False + 1:NEXT(1049$);

1048$    TRANSPORT:  LAGVT,RStation7700;

1049$    TRANSPORT:  LAGVT,LStation7700;

;
;
;    Model statements for module: Decide 1426
;
1086$    BRANCH,    1:
           If,SpeedatRStation7600>75,1087$,Yes:
           If,SpeedatRStation7600 > 55 && SpeedatRStation7600<= 75,1088$,Yes:
           If,SpeedatRStation7600 >35 && SpeedatRStation7600 <=55,1089$,Yes:
           If,SpeedatRStation7600 >10 && SpeedatRStation7600 <=35,1090$,Yes:
           Else,1091$,Yes;

;
;
;    Model statements for module: Decide 1431
;
1091$    BRANCH,    1:
           If,NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1) ==
0,3117$,Yes:
           Else,3118$,Yes;

```

3117\$ ASSIGN: CheckGapL7650Speed10lesst.NumberOut
 True=CheckGapL7650Speed10lesst.NumberOut True + 1:NEXT(1100\$);

3118\$ ASSIGN: CheckGapL7650Speed10lesst.NumberOut
 False=CheckGapL7650Speed10lesst.NumberOut False + 1:NEXT(1101\$);

1100\$ TRANSPORT: LAGVT,RStation7700;

1101\$ TRANSPORT: LAGVT,LStation7700;

;
 ;

; Model statements for module: Decide 1427

;

1087\$ BRANCH, 1:
 If,
 NZ(RL9,4) && NZ(RL10,1) &&NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2)
 &&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2) == 0,
 3119\$,Yes:
 Else,3120\$,Yes;

3119\$ ASSIGN: CheckGapL7650Speed75lesst.NumberOut
 True=CheckGapL7650Speed75lesst.NumberOut True + 1:NEXT(1092\$);

3120\$ ASSIGN: CheckGapL7650Speed75lesst.NumberOut
 False=CheckGapL7650Speed75lesst.NumberOut False + 1:NEXT(1093\$);

1092\$ TRANSPORT: LAGVT,RStation7700;

1093\$ TRANSPORT: LAGVT,LStation7700;

;
 ;

; Model statements for module: Decide 1428

;

1088\$ BRANCH, 1:
 If,
 NZ(RL10,1) &&NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2)
 &&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1) == 0,
 3121\$,Yes:
 Else,3122\$,Yes;

3121\$ ASSIGN: CheckGapL7650Speed55_75lesst.NumberOut
 True=CheckGapL7650Speed55_75lesst.NumberOut True + 1
 :NEXT(1094\$);

3122\$ ASSIGN: CheckGapL7650Speed55_75lesst.NumberOut
 False=CheckGapL7650Speed55_75lesst.NumberOut False + 1
 :NEXT(1095\$);

1094\$ TRANSPORT: LAGVT,RStation7700;

1095\$ TRANSPORT: LAGVT,LStation7700;

```

;
;
;   Model statements for module: Decide 1429
;
1089$   BRANCH,    1:
        If,NZ(RL11,1)&&NZ(RL11,2)
&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1) == 0,3123$,
        Yes:
        Else,3124$,Yes;
3123$   ASSIGN:    CheckGapL7650Speed35_55lesst.NumberOut
True=CheckGapL7650Speed35_55lesst.NumberOut True + 1
        :NEXT(1096$);

3124$   ASSIGN:    CheckGapL7650Speed35_55lesst.NumberOut
False=CheckGapL7650Speed35_55lesst.NumberOut False + 1
        :NEXT(1097$);

1096$   TRANSPORT:  LAGVT,RStation7700;

1097$   TRANSPORT:  LAGVT,LStation7700;

;
;
;   Model statements for module: Decide 1430
;
1090$   BRANCH,    1:
        If,NZ(RL11,2)
&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1) == 0,3125$,Yes:
        Else,3126$,Yes;
3125$   ASSIGN:    CheckGapL7650Speed10_35lesst.NumberOut
True=CheckGapL7650Speed10_35lesst.NumberOut True + 1
        :NEXT(1098$);

3126$   ASSIGN:    CheckGapL7650Speed10_35lesst.NumberOut
False=CheckGapL7650Speed10_35lesst.NumberOut False + 1
        :NEXT(1099$);

1098$   TRANSPORT:  LAGVT,RStation7700;

1099$   TRANSPORT:  LAGVT,LStation7700;

;
;
;   Model statements for module: Station 111
;
1102$   STATION,    RStation7700;
3129$   DELAY:      0.0,,VA:NEXT(1114$);

```

```

;
;
;   Model statements for module: Decide 1435
;
1114$   BRANCH,      1:
        If,Entity.Type==RCar,1110$,Yes:
        If,Entity.Type==RTruck,1111$,Yes:
        If,Entity.Type==LCar,1112$,Yes:
        Else,1113$,Yes;

;
;
;   Model statements for module: Assign 170
;
1113$   ASSIGN:      SpeedatRStation7700=VTU(LAGVT,LTruck#):NEXT(1104$);

;
;
;   Model statements for module: Decide 1432
;
1104$   BRANCH,      1:
        If,Entity.Type==RCar,3132$,Yes:
        Else,3133$,Yes;
3132$   ASSIGN:      Decide 1432.NumberOut True=Decide 1432.NumberOut True +
1:NEXT(1103$);

3133$   ASSIGN:      Decide 1432.NumberOut False=Decide 1432.NumberOut False +
1:NEXT(1106$);

1103$   TRANSPORT:   RAGV,RStation7750;

;
;
;   Model statements for module: Decide 1433
;
1106$   BRANCH,      1:
        If,Entity.Type==RTruck,3134$,Yes:
        Else,3135$,Yes;
3134$   ASSIGN:      Decide 1433.NumberOut True=Decide 1433.NumberOut True +
1:NEXT(1105$);

3135$   ASSIGN:      Decide 1433.NumberOut False=Decide 1433.NumberOut False +
1:NEXT(1107$);

1105$   TRANSPORT:   RAGVT,RStation7750;

;
;
;   Model statements for module: Decide 1434

```

```

;
1107$    BRANCH,    1:
          If,Entity.Type==LCar,3136$,Yes:
          Else,3137$,Yes;
3136$    ASSIGN:    Decide 1434.NumberOut True=Decide 1434.NumberOut True +
1:NEXT(1108$);

3137$    ASSIGN:    Decide 1434.NumberOut False=Decide 1434.NumberOut False +
1:NEXT(1109$);

1108$    TRANSPORT:  LAGV,RStation7750;

1109$    TRANSPORT:  LAGVT,RStation7750;

;
;
;    Model statements for module: Assign 167
;
1110$    ASSIGN:    SpeedatRStation7700=VTU(RAGV,RCar#):NEXT(1104$);

;
;
;    Model statements for module: Assign 168
;
1111$    ASSIGN:    SpeedatRStation7700=VTU(RAGVT,RTruck#):NEXT(1104$);

;
;
;    Model statements for module: Assign 169
;
1112$    ASSIGN:    SpeedatRStation7700=VT(LAGV):NEXT(1104$);

;
;
;    Model statements for module: Station 112
;

1115$    STATION,    LStation7700;
3140$    DELAY:    0.0,,VA:NEXT(1116$);

;
;
;    Model statements for module: Decide 1436
;
1116$    BRANCH,    1:
          If,Entity.Type==LCar,3141$,Yes:
          Else,3142$,Yes;

```



```

3141$    ASSIGN:    checkvehicletype7700.NumberOut True=checkvehicletype7700.NumberOut
True + 1:NEXT(1117$);

```

```

3142$    ASSIGN:    checkvehicletype7700.NumberOut False=checkvehicletype7700.NumberOut
False + 1:NEXT(1118$);

```

```

;
;
; Model statements for module: Decide 1437
;

```

```

1117$    BRANCH,    1:
                If,SpeedatRStation7650 >= VTU(LAGV,LCar#),3143$,Yes:
                Else,3144$,Yes;
3143$    ASSIGN:    Decide 1437.NumberOut True=Decide 1437.NumberOut True +
1:NEXT(1119$);

```

```

3144$    ASSIGN:    Decide 1437.NumberOut False=Decide 1437.NumberOut False +
1:NEXT(1145$);

```

```

;
;
; Model statements for module: Decide 1439
;

```

```

1119$    BRANCH,    1:
                If,SpeedatRStation7650>75,1120$,Yes:
                If,SpeedatRStation7650 > 55 && SpeedatRStation7650 <= 75,1133$,Yes:
                If,SpeedatRStation7650 >35 && SpeedatRStation7650<=55,1136$, Yes:
                If,SpeedatRStation7650 >10 && SpeedatRStation7650 <=35,1139$,Yes:
                Else,1142$,Yes;

```

```

;
;
; Model statements for module: Decide 1444
;

```

```

1142$    BRANCH,    1:
                If,NL(RL15)&&NZ(RL14,2) == 0,3147$,Yes:
                Else,3148$,Yes;
3147$    ASSIGN:    CheckGapL7700Speed10.NumberOut
True=CheckGapL7700Speed10.NumberOut True + 1:NEXT(1143$);

```

```

3148$    ASSIGN:    CheckGapL7700Speed10.NumberOut
False=CheckGapL7700Speed10.NumberOut False + 1:NEXT(1144$);

```

```

1143$    TRANSPORT:    LAGV,RStation7750;

```

```

1144$    TRANSPORT:    LAGV,LStation7750;

```

```

;
;
; Model statements for module: Decide 1440

```

```

;
1120$    BRANCH,      1:
          If,NL(RL15)&&NL(RL16) &&NL(RL17)&&NL(RL18)&& NZ(RL14,2) ==
0,3149$,Yes:
          Else,3150$,Yes;
3149$    ASSIGN:      CheckGapL7700Speed75.NumberOut
True=CheckGapL7700Speed75.NumberOut True + 1:NEXT(1121$);

3150$    ASSIGN:      CheckGapL7700Speed75.NumberOut
False=CheckGapL7700Speed75.NumberOut False + 1:NEXT(1122$);

1121$    TRANSPORT:   LAGV,RStation7750;

1122$    TRANSPORT:   LAGV,LStation7750;

;
;
;    Model statements for module: Decide 1441
;
1133$    BRANCH,      1:
          If,NL(RL15)&&NL(RL16) &&NL(RL17)&&NZ(RL14,2) == 0,3151$,Yes:
          Else,3152$,Yes;
3151$    ASSIGN:      CheckGapL7700Speed55_75.NumberOut
True=CheckGapL7700Speed55_75.NumberOut True + 1:NEXT(1134$);

3152$    ASSIGN:      CheckGapL7700Speed55_75.NumberOut
False=CheckGapL7700Speed55_75.NumberOut False + 1:NEXT(1135$);

1134$    TRANSPORT:   LAGV,RStation7750;

1135$    TRANSPORT:   LAGV,LStation7750;

;
;
;    Model statements for module: Decide 1442
;
1136$    BRANCH,      1:
          If,NL(RL15)&&NL(RL16) &&NZ(RL14,2) == 0,3153$,Yes:
          Else,3154$,Yes;
3153$    ASSIGN:      CheckGapL7700Speed35_55.NumberOut
True=CheckGapL7700Speed35_55.NumberOut True + 1:NEXT(1137$);

3154$    ASSIGN:      CheckGapL7700Speed35_55.NumberOut
False=CheckGapL7700Speed35_55.NumberOut False + 1:NEXT(1138$);

1137$    TRANSPORT:   LAGV,RStation7750;

1138$    TRANSPORT:   LAGV,LStation7750;

;

```

```

;
;   Model statements for module: Decide 1443
;
1139$   BRANCH,      1:
          If,NL(RL15)&&NZ(RL16,1) &&NZ(RL16,2) &&NZ(RL16,3) &&NZ(RL16,4)
&&NZ(RL14,2) == 0,3155$,Yes:
          Else,3156$,Yes;
3155$   ASSIGN:      CheckGapL7700Speed10_35.NumberOut
True=CheckGapL7700Speed10_35.NumberOut True + 1:NEXT(1140$);

3156$   ASSIGN:      CheckGapL7700Speed10_35.NumberOut
False=CheckGapL7700Speed10_35.NumberOut False + 1:NEXT(1141$);

1140$   TRANSPORT:   LAGV,RStation7750;

1141$   TRANSPORT:   LAGV,LStation7750;

;
;
;   Model statements for module: Decide 1445
;
1145$   BRANCH,      1:
          If,SpeedatRStation7650>75,1146$,Yes:
          If,SpeedatRStation7650 > 55 && SpeedatRStation7650 <= 75,1149$,Yes:
          If,SpeedatRStation7650>35 && SpeedatRStation7650 <=55,1152$,Yes:
          If,SpeedatRStation7650>10 && SpeedatRStation7650 <=35,1155$,Yes:
          Else,1158$,Yes;

;
;
;   Model statements for module: Decide 1450
;
1158$   BRANCH,      1:
          If,NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2) == 0,3159$,Yes:
          Else,3160$,Yes;
3159$   ASSIGN:      CheckGapL7700Speed10less.NumberOut
True=CheckGapL7700Speed10less.NumberOut True + 1:NEXT(1159$);

3160$   ASSIGN:      CheckGapL7700Speed10less.NumberOut
False=CheckGapL7700Speed10less.NumberOut False + 1:NEXT(1160$);

1159$   TRANSPORT:   LAGV,RStation7750;

1160$   TRANSPORT:   LAGV,LStation7750;

;
;
;   Model statements for module: Decide 1446
;
1146$   BRANCH,      1:
          If,

```

```

      NZ(RL11,2)&&NZ(RL12,1)&&NZ(RL12,2)
&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)&&NZ(RL15,5)== 0,

```

```

      3161$,Yes:

```

```

      Else,3162$,Yes;

```

```

3161$    ASSIGN:    CheckGapL7700Speed75less.NumberOut
True=CheckGapL7700Speed75less.NumberOut True + 1:NEXT(1147$);

```

```

3162$    ASSIGN:    CheckGapL7700Speed75less.NumberOut
False=CheckGapL7700Speed75less.NumberOut False + 1:NEXT(1148$);

```

```

1147$    TRANSPORT:  LAGV,RStation7750;

```

```

1148$    TRANSPORT:  LAGV,LStation7750;

```

```

;

```

```

;

```

```

;   Model statements for module: Decide 1447

```

```

;

```

```

1149$    BRANCH,      1:

```

```

      If,

```

```

      NZ(RL12,1) &&NZ(RL12,2)

```

```

&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)== 0,

```

```

      3163$,Yes:

```

```

      Else,3164$,Yes;

```

```

3163$    ASSIGN:    CheckGapL7700Speed55_75less.NumberOut
True=CheckGapL7700Speed55_75less.NumberOut True + 1
      :NEXT(1150$);

```

```

3164$    ASSIGN:    CheckGapL7700Speed55_75less.NumberOut
False=CheckGapL7700Speed55_75less.NumberOut False + 1
      :NEXT(1151$);

```

```

1150$    TRANSPORT:  LAGV,RStation7750;

```

```

1151$    TRANSPORT:  LAGV,LStation7750;

```

```

;

```

```

;

```

```

;   Model statements for module: Decide 1448

```

```

;

```

```

1152$    BRANCH,      1:

```

```

      If,

```

```

      NZ(RL12,2)

```

```

&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)== 0,

```

```

      3165$,Yes:

```

```

      Else,3166$,Yes;

```

```

3165$    ASSIGN:    CheckGapL7700Speed35_55less.NumberOut
True=CheckGapL7700Speed35_55less.NumberOut True + 1

```

```

: NEXT(1153$);

3166$    ASSIGN:    CheckGapL7700Speed35_55less.NumberOut
False=CheckGapL7700Speed35_55less.NumberOut False + 1
: NEXT(1154$);

1153$    TRANSPORT:  LAGV,RStation7750;

1154$    TRANSPORT:  LAGV,LStation7750;

;
;
; Model statements for module: Decide 1449
;
1155$    BRANCH,    1:
            If,NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)== 0,3167$,
            Yes:
            Else,3168$,Yes;
3167$    ASSIGN:    CheckGapL7700Speed10_35less.NumberOut
True=CheckGapL7700Speed10_35less.NumberOut True + 1
: NEXT(1156$);

3168$    ASSIGN:    CheckGapL7700Speed10_35less.NumberOut
False=CheckGapL7700Speed10_35less.NumberOut False + 1
: NEXT(1157$);

1156$    TRANSPORT:  LAGV,RStation7750;

1157$    TRANSPORT:  LAGV,LStation7750;

;
;
; Model statements for module: Decide 1438
;
1118$    BRANCH,    1:
            If,SpeedatRStation7650 >= VTU(LAGVT,LTruck#),3169$,Yes:
            Else,3170$,Yes;
3169$    ASSIGN:    Decide 1438.NumberOut True=Decide 1438.NumberOut True +
1:NEXT(1161$);

3170$    ASSIGN:    Decide 1438.NumberOut False=Decide 1438.NumberOut False +
1:NEXT(1167$);

;
;
; Model statements for module: Decide 1451
;
1161$    BRANCH,    1:
            If,SpeedatRStation7650>75,1162$,Yes:

```

```

If,SpeedatRStation7650 > 55 && SpeedatRStation7650 <= 75,1163$,Yes:
If,SpeedatRStation7650 >35 && SpeedatRStation7650 <=55,1164$,Yes:
If,SpeedatRStation7650 >10 && SpeedatRStation7650 <=35,1165$,Yes:
Else,1166$,Yes;

;
;
;   Model statements for module: Decide 1456
;
1166$   BRANCH,      1:
        If,NL(RL15)&&NL(RL16) &&NZ(RL14,2) == 0,3173$,Yes:
        Else,3174$,Yes;
3173$   ASSIGN:      CheckGapL7700Speed10t.NumberOut
True=CheckGapL7700Speed10t.NumberOut True + 1:NEXT(1131$);

3174$   ASSIGN:      CheckGapL7700Speed10t.NumberOut
False=CheckGapL7700Speed10t.NumberOut False + 1:NEXT(1132$);

1131$   TRANSPORT:   LAGVT,RStation7750;

1132$   TRANSPORT:   LAGVT,LStation7750;

;
;
;   Model statements for module: Decide 1452
;
1162$   BRANCH,      1:
        If,NL(RL15)&&NL(RL16) &&NL(RL17)&&NL(RL18)&&NL(RL19)&& NZ(RL14,2)
== 0,3175$,Yes:
        Else,3176$,Yes;
3175$   ASSIGN:      CheckGapL7700Speed75t.NumberOut
True=CheckGapL7700Speed75t.NumberOut True + 1:NEXT(1123$);

3176$   ASSIGN:      CheckGapL7700Speed75t.NumberOut
False=CheckGapL7700Speed75t.NumberOut False + 1:NEXT(1124$);

1123$   TRANSPORT:   LAGVT,RStation7750;

1124$   TRANSPORT:   LAGVT,LStation7750;

;
;
;   Model statements for module: Decide 1453
;
1163$   BRANCH,      1:
        If,NL(RL15)&&NL(RL16) &&NL(RL17)&&NL(RL18)&&NZ(RL14,2) ==
0,3177$,Yes:
        Else,3178$,Yes;
3177$   ASSIGN:      CheckGapL7700Speed55_75t.NumberOut
True=CheckGapL7700Speed55_75t.NumberOut True + 1:NEXT(1126$);

```

3178\$ ASSIGN: CheckGapL7700Speed55_75t.NumberOut
False=CheckGapL7700Speed55_75t.NumberOut False + 1:NEXT(1125\$);

1126\$ TRANSPORT: LAGVT,RStation7750;

1125\$ TRANSPORT: LAGVT,LStation7750;

;
;

; Model statements for module: Decide 1454

;

1164\$ BRANCH, 1:
 If,NL(RL15)&&NL(RL16) &&NL(RL17)&&NZ(RL14,2) == 0,3179\$,Yes:
 Else,3180\$,Yes;

3179\$ ASSIGN: CheckGapL7700Speed35_55t.NumberOut
True=CheckGapL7700Speed35_55t.NumberOut True + 1:NEXT(1127\$);

3180\$ ASSIGN: CheckGapL7700Speed35_55t.NumberOut
False=CheckGapL7700Speed35_55t.NumberOut False + 1:NEXT(1128\$);

1127\$ TRANSPORT: LAGVT,RStation7750;

1128\$ TRANSPORT: LAGVT,LStation7750;

;
;

; Model statements for module: Decide 1455

;

1165\$ BRANCH, 1:
 If,NL(RL15)&&NL(RL16) &&NZ(RL14,2) == 0,3181\$,Yes:
 Else,3182\$,Yes;

3181\$ ASSIGN: CheckGapL7700Speed10_35t.NumberOut
True=CheckGapL7700Speed10_35t.NumberOut True + 1:NEXT(1129\$);

3182\$ ASSIGN: CheckGapL7700Speed10_35t.NumberOut
False=CheckGapL7700Speed10_35t.NumberOut False + 1:NEXT(1130\$);

1129\$ TRANSPORT: LAGVT,RStation7750;

1130\$ TRANSPORT: LAGVT,LStation7750;

;
;

; Model statements for module: Decide 1457

;

1167\$ BRANCH, 1:
 If,SpeedatRStation7650>75,1168\$,Yes:
 If,SpeedatRStation7650 > 55 && SpeedatRStation7650<= 75,1169\$,Yes:
 If,SpeedatRStation7650 >35 && SpeedatRStation7650 <=55,1170\$,Yes:
 If,SpeedatRStation7650 >10 && SpeedatRStation7650 <=35,1171\$,Yes:

```

Else,1172$,Yes;

;
;
;   Model statements for module: Decide 1462
;
1172$   BRANCH,      1:
        If,

NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)&&NZ(RL15,5)== 0,
        3185$,Yes:
        Else,3186$,Yes;
3185$   ASSIGN:      CheckGapL7700Speed10lesst.NumberOut
True=CheckGapL7700Speed10lesst.NumberOut True + 1:NEXT(1181$);

3186$   ASSIGN:      CheckGapL7700Speed10lesst.NumberOut
False=CheckGapL7700Speed10lesst.NumberOut False + 1:NEXT(1182$);

1181$   TRANSPORT:   LAGVT,RStation7750;

1182$   TRANSPORT:   LAGVT,LStation7750;

;
;
;   Model statements for module: Decide 1458
;
1168$   BRANCH,      1:
        If,
        NZ(RL10,2)&&NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1)&&NZ(RL12,2)
&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)&&NZ(RL15,5)== 0,
        3187$,Yes:
        Else,3188$,Yes;
3187$   ASSIGN:      CheckGapL7700Speed75lesst.NumberOut
True=CheckGapL7700Speed75lesst.NumberOut True + 1:NEXT(1173$);

3188$   ASSIGN:      CheckGapL7700Speed75lesst.NumberOut
False=CheckGapL7700Speed75lesst.NumberOut False + 1:NEXT(1174$);

1173$   TRANSPORT:   LAGVT,RStation7750;

1174$   TRANSPORT:   LAGVT,LStation7750;

;
;
;   Model statements for module: Decide 1459
;
1169$   BRANCH,      1:
        If,

```



```

NZ(RL11,2)&&NZ(RL12,1)&&NZ(RL12,2)
&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)&&NZ(RL15,5)== 0,

```

```

3189$,Yes:

```

```

Else,3190$,Yes;

```

```

3189$    ASSIGN:    CheckGapL7700Speed55_75lesst.NumberOut
True=CheckGapL7700Speed55_75lesst.NumberOut True + 1
:NEXT(1175$);

```

```

3190$    ASSIGN:    CheckGapL7700Speed55_75lesst.NumberOut
False=CheckGapL7700Speed55_75lesst.NumberOut False + 1
:NEXT(1176$);

```

```

1175$    TRANSPORT:  LAGVT,RStation7750;

```

```

1176$    TRANSPORT:  LAGVT,LStation7750;

```

```

;

```

```

;

```

```

;    Model statements for module: Decide 1460

```

```

;

```

```

1170$    BRANCH,      1:

```

```

If,

```

```

NZ(RL12,1)&&NZ(RL12,2)

```

```

&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)&&NZ(RL15,5)== 0,

```

```

3191$,Yes:

```

```

Else,3192$,Yes;

```

```

3191$    ASSIGN:    CheckGapL7700Speed35_55lesst.NumberOut
True=CheckGapL7700Speed35_55lesst.NumberOut True + 1
:NEXT(1177$);

```

```

3192$    ASSIGN:    CheckGapL7700Speed35_55lesst.NumberOut
False=CheckGapL7700Speed35_55lesst.NumberOut False + 1
:NEXT(1178$);

```

```

1177$    TRANSPORT:  LAGVT,RStation7750;

```

```

1178$    TRANSPORT:  LAGVT,LStation7750;

```

```

;

```

```

;

```

```

;    Model statements for module: Decide 1461

```

```

;

```

```

1171$    BRANCH,      1:

```

```

If,

```

```

NZ(RL12,2)

```

```

&&NZ(RL13,1)&&NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NZ(RL15,1)&&NZ(RL15,2)
&&NZ(RL15,3)&&NZ(RL15,4)&&NZ(RL15,5)== 0,

```

```

3193$,Yes:

```

```

Else,3194$,Yes;

```

```

3193$    ASSIGN:    CheckGapL7700Speed10_35lesst.NumberOut
True=CheckGapL7700Speed10_35lesst.NumberOut True + 1
          :NEXT(1179$);

```

```

3194$    ASSIGN:    CheckGapL7700Speed10_35lesst.NumberOut
False=CheckGapL7700Speed10_35lesst.NumberOut False + 1
          :NEXT(1180$);

```

```

1179$    TRANSPORT:  LAGVT,RStation7750;

```

```

1180$    TRANSPORT:  LAGVT,LStation7750;

```

```

;
;
;  Model statements for module: Station 113
;

```

```

1183$    STATION,    RStation7750;
3197$    DELAY:      0.0,,VA:NEXT(1216$);

```

```

;
;
;  Model statements for module: Decide 1475
;

```

```

1216$    BRANCH,    1:
          If,Entity.Type==RCar,1212$,Yes:
          If,Entity.Type==RTruck,1213$,Yes:
          If,Entity.Type==LCar,1214$,Yes:
          Else,1215$,Yes;

```

```

;
;
;  Model statements for module: Assign 174
;

```

```

1215$    ASSIGN:    SpeedatRStation7750=VTU(LAGVT,LTruck#):NEXT(1188$);

```

```

;
;
;  Model statements for module: Decide 1463
;

```

```

1188$    BRANCH,    1:
          If,Entity.Type==RCar,3200$,Yes:
          Else,3201$,Yes;

```

```

3200$    ASSIGN:    Decide 1463.NumberOut True=Decide 1463.NumberOut True +
1:NEXT(1187$);

```

```

3201$    ASSIGN:    Decide 1463.NumberOut False=Decide 1463.NumberOut False +
1:NEXT(1190$);

```

```

1187$    TRANSPORT:  RAGV,RStation7800;

```

```

;
;
;   Model statements for module: Decide 1464
;
1190$   BRANCH,      1:
          If,Entity.Type==RTruck,3202$,Yes:
          Else,3203$,Yes;
3202$   ASSIGN:      Decide 1464.NumberOut True=Decide 1464.NumberOut True +
1:NEXT(1189$);

3203$   ASSIGN:      Decide 1464.NumberOut False=Decide 1464.NumberOut False +
1:NEXT(1191$);

1189$   TRANSPORT:   RAGVT,RStation7800;

;
;
;   Model statements for module: Decide 1465
;
1191$   BRANCH,      1:
          If,Entity.Type==LCar,3204$,Yes:
          Else,3205$,Yes;
3204$   ASSIGN:      Decide 1465.NumberOut True=Decide 1465.NumberOut True +
1:NEXT(1192$);

3205$   ASSIGN:      Decide 1465.NumberOut False=Decide 1465.NumberOut False +
1:NEXT(1193$);

1192$   TRANSPORT:   LAGV,RStation7800;

1193$   TRANSPORT:   LAGVT,RStation7800;

;
;
;   Model statements for module: Assign 171
;
1212$   ASSIGN:      SpeedatRStation7750=VTU(RAGV,RCar#):NEXT(1188$);

;
;
;   Model statements for module: Assign 172
;
1213$   ASSIGN:      SpeedatRStation7750=VTU(RAGVT,RTruck#):NEXT(1188$);

;
;
;   Model statements for module: Assign 173

```

```

;
1214$    ASSIGN:    SpeedatRStation7750=VT(LAGV):NEXT(1188$);

;
;
;    Model statements for module: Station 114
;

1184$    STATION,    RStation7800;
3208$    DELAY:      0.0,,VA:NEXT(1221$);

;
;
;    Model statements for module: Decide 1476
;
1221$    BRANCH,      1:
                If,Entity.Type==RCar,1217$,Yes:
                If,Entity.Type==RTruck,1218$,Yes:
                If,Entity.Type==LCar,1219$,Yes:
                Else,1220$,Yes;

;
;
;    Model statements for module: Assign 178
;
1220$    ASSIGN:      SpeedatRStation7800=VTU(LAGVT,LTruck#):NEXT(1195$);

;
;
;    Model statements for module: Decide 1466
;
1195$    BRANCH,      1:
                If,Entity.Type==RCar,3211$,Yes:
                Else,3212$,Yes;
3211$    ASSIGN:      Decide 1466.NumberOut True=Decide 1466.NumberOut True +
1:NEXT(1194$);

3212$    ASSIGN:      Decide 1466.NumberOut False=Decide 1466.NumberOut False +
1:NEXT(1197$);

1194$    TRANSPORT:    RAGV,RStation7850;

;
;
;    Model statements for module: Decide 1467
;
1197$    BRANCH,      1:
                If,Entity.Type==RTruck,3213$,Yes:
                Else,3214$,Yes;

```

```
3213$    ASSIGN:    Decide 1467.NumberOut True=Decide 1467.NumberOut True +
1:NEXT(1196$);
```

```
3214$    ASSIGN:    Decide 1467.NumberOut False=Decide 1467.NumberOut False +
1:NEXT(1198$);
```

```
1196$    TRANSPORT:  RAGVT,RStation7850;
```

```
;
;
;    Model statements for module: Decide 1468
;
```

```
1198$    BRANCH,    1:
                If,Entity.Type==LCar,3215$,Yes:
                Else,3216$,Yes;
```

```
3215$    ASSIGN:    Decide 1468.NumberOut True=Decide 1468.NumberOut True +
1:NEXT(1199$);
```

```
3216$    ASSIGN:    Decide 1468.NumberOut False=Decide 1468.NumberOut False +
1:NEXT(1200$);
```

```
1199$    TRANSPORT:  LAGV,RStation7850;
```

```
1200$    TRANSPORT:  LAGVT,RStation7850;
```

```
;
;
;    Model statements for module: Assign 175
;
1217$    ASSIGN:    SpeedatRStation7800=VTU(RAGV,RCar#):NEXT(1195$);
```

```
;
;
;    Model statements for module: Assign 176
;
1218$    ASSIGN:    SpeedatRStation7800=VTU(RAGVT,RTruck#):NEXT(1195$);
```

```
;
;
;    Model statements for module: Assign 177
;
1219$    ASSIGN:    SpeedatRStation7800=VT(LAGV):NEXT(1195$);
```

```
;
;
;    Model statements for module: Station 115
;
```

```

1185$    STATION,    RStation7850;
3219$    DELAY:      0.0,,VA:NEXT(1226$);

;
;
; Model statements for module: Decide 1477
;
1226$    BRANCH,      1:
          If,Entity.Type==RCar,1222$,Yes:
          If,Entity.Type==RTruck,1223$,Yes:
          If,Entity.Type==LCar,1224$,Yes:
          Else,1225$,Yes;

;
;
; Model statements for module: Assign 182
;
1225$    ASSIGN:      SpeedatRStation7850=VTU(LAGVT,LTruck#):NEXT(1202$);

;
;
; Model statements for module: Decide 1469
;
1202$    BRANCH,      1:
          If,Entity.Type==RCar,3222$,Yes:
          Else,3223$,Yes;
3222$    ASSIGN:      Decide 1469.NumberOut True=Decide 1469.NumberOut True +
1:NEXT(1201$);

3223$    ASSIGN:      Decide 1469.NumberOut False=Decide 1469.NumberOut False +
1:NEXT(1204$);

1201$    TRANSPORT:   RAGV,RStation7900;

;
;
; Model statements for module: Decide 1470
;
1204$    BRANCH,      1:
          If,Entity.Type==RTruck,3224$,Yes:
          Else,3225$,Yes;
3224$    ASSIGN:      Decide 1470.NumberOut True=Decide 1470.NumberOut True +
1:NEXT(1203$);

3225$    ASSIGN:      Decide 1470.NumberOut False=Decide 1470.NumberOut False +
1:NEXT(1205$);

1203$    TRANSPORT:   RAGVT,RStation7900;

```

```

;
;
;   Model statements for module: Decide 1471
;
1205$   BRANCH,      1:
          If,Entity.Type==LCar,3226$,Yes:
          Else,3227$,Yes;
3226$   ASSIGN:      Decide 1471.NumberOut True=Decide 1471.NumberOut True +
1:NEXT(1206$);

3227$   ASSIGN:      Decide 1471.NumberOut False=Decide 1471.NumberOut False +
1:NEXT(1207$);

1206$   TRANSPORT:   LAGV,RStation7900;

1207$   TRANSPORT:   LAGVT,RStation7900;

;
;
;   Model statements for module: Assign 179
;
1222$   ASSIGN:      SpeedatRStation7850=VTU(RAGV,RCar#):NEXT(1202$);

;
;
;   Model statements for module: Assign 180
;
1223$   ASSIGN:      SpeedatRStation7850=VTU(RAGVT,RTruck#):NEXT(1202$);

;
;
;   Model statements for module: Assign 181
;
1224$   ASSIGN:      SpeedatRStation7850=VTU(LAGV,LCar#):NEXT(1202$);

;
;
;   Model statements for module: Station 116
;
1186$   STATION,      RStation7900;
3230$   DELAY:        0.0,,VA:NEXT(1231$);

;
;
;   Model statements for module: Decide 1478
;
1231$   BRANCH,      1:

```

```

    If,Entity.Type==RCar,1227$,Yes:
    If,Entity.Type==RTruck,1228$,Yes:
    If,Entity.Type==LCar,1229$,Yes:
    Else,1230$,Yes;

;
;
;   Model statements for module: Assign 186
;
1230$   ASSIGN:   SpeedatRStation7900=VTU(LAGVT,LTruck#):NEXT(1211$);

1211$   TRANSPORT:  LAGVT,RStation7950;

;
;
;   Model statements for module: Assign 183
;
1227$   ASSIGN:   SpeedatRStation7900=VTU(RAGV,RCar#):NEXT(1208$);

1208$   TRANSPORT:  RAGV,RStation7950;

;
;
;   Model statements for module: Assign 184
;
1228$   ASSIGN:   SpeedatRStation7900=VTU(RAGVT,RTruck#):NEXT(1209$);

1209$   TRANSPORT:  RAGVT,RStation7950;

;
;
;   Model statements for module: Assign 185
;
1229$   ASSIGN:   SpeedatRStation7900=VTU(LAGV,LCar#):NEXT(1210$);

1210$   TRANSPORT:  LAGV,RStation7950;

;
;
;   Model statements for module: Station 117
;

1232$   STATION,   LStation7750;
3235$   DELAY:     0.0,,VA:NEXT(1233$);

;
;
;   Model statements for module: Decide 1479

```



```

;
1233$    BRANCH,    1:
            If,Entity.Type==LCar,3236$,Yes:
            Else,3237$,Yes;
3236$    ASSIGN:    checkvehicletype7750.NumberOut True=checkvehicletype7750.NumberOut
True + 1:NEXT(1234$);

3237$    ASSIGN:    checkvehicletype7750.NumberOut False=checkvehicletype7750.NumberOut
False + 1:NEXT(1235$);

```

```

;
;
; Model statements for module: Decide 1480
;
1234$    BRANCH,    1:
            If,SpeedatRStation7700 >= VTU(LAGV,LCar#),3238$,Yes:
            Else,3239$,Yes;
3238$    ASSIGN:    Decide 1480.NumberOut True=Decide 1480.NumberOut True +
1:NEXT(1236$);

3239$    ASSIGN:    Decide 1480.NumberOut False=Decide 1480.NumberOut False +
1:NEXT(1262$);

```

```

;
;
; Model statements for module: Decide 1482
;
1236$    BRANCH,    1:
            If,SpeedatRStation7700>75,1237$,Yes:
            If,SpeedatRStation7700 > 55 && SpeedatRStation7700 <= 75,1250$,Yes:
            If,SpeedatRStation7700 >35 && SpeedatRStation7700<=55,1253$,Yes:
            If,SpeedatRStation7700 >10 && SpeedatRStation7700 <=35,1256$,Yes:
            Else,1259$,Yes;

```

```

;
;
; Model statements for module: Decide 1487
;
1259$    BRANCH,    1:
            If,NL(RL16)&&NZ(RL17,1) &&NZ(RL15,10) == 0,3242$,Yes:
            Else,3243$,Yes;
3242$    ASSIGN:    CheckGapL7750Speed10.NumberOut
True=CheckGapL7750Speed10.NumberOut True + 1:NEXT(1260$);

3243$    ASSIGN:    CheckGapL7750Speed10.NumberOut
False=CheckGapL7750Speed10.NumberOut False + 1:NEXT(1261$);

```

```

1260$    TRANSPORT:    LAGV,RStation7800;

```

```

1261$    TRANSPORT:    LAGV,LStation7800;

```

```

;
;
;   Model statements for module: Decide 1483
;
1237$   BRANCH,      1:
        If,
            NL(RL16) &&NL(RL17)&&NL(RL18)&&NL(RL19)&& NZ(RL15,6) && NZ(RL15,7)
&& NZ(RL15,8) && NZ(RL15,9) && NZ(RL15,10) == 0,
            3244$,Yes:
            Else,3245$,Yes;
3244$   ASSIGN:      CheckGapL7750Speed75.NumberOut
True=CheckGapL7750Speed75.NumberOut True + 1:NEXT(1238$);

3245$   ASSIGN:      CheckGapL7750Speed75.NumberOut
False=CheckGapL7750Speed75.NumberOut False + 1:NEXT(1239$);

1238$   TRANSPORT:   LAGV,RStation7800;

1239$   TRANSPORT:   LAGV,LStation7800;

;
;
;   Model statements for module: Decide 1484
;
1250$   BRANCH,      1:
        If,NL(RL16)&&NL(RL17)&&NL(RL18)&&NZ(RL15,7) && NZ(RL15,8) &&
NZ(RL15,9) && NZ(RL15,10) == 0,3246$,
        Yes:
        Else,3247$,Yes;
3246$   ASSIGN:      CheckGapL7750Speed55_75.NumberOut
True=CheckGapL7750Speed55_75.NumberOut True + 1:NEXT(1251$);

3247$   ASSIGN:      CheckGapL7750Speed55_75.NumberOut
False=CheckGapL7750Speed55_75.NumberOut False + 1:NEXT(1252$);

1251$   TRANSPORT:   LAGV,RStation7800;

1252$   TRANSPORT:   LAGV,LStation7800;

;
;
;   Model statements for module: Decide 1485
;
1253$   BRANCH,      1:
        If,NL(RL16)&&NL(RL17)&&NZ(RL15,7) && NZ(RL15,8) && NZ(RL15,9) &&
NZ(RL15,10) == 0,3248$,Yes:
        Else,3249$,Yes;
3248$   ASSIGN:      CheckGapL7750Speed35_55.NumberOut
True=CheckGapL7750Speed35_55.NumberOut True + 1:NEXT(1254$);

```

```
3249$    ASSIGN:    CheckGapL7750Speed35_55.NumberOut
False=CheckGapL7750Speed35_55.NumberOut False + 1:NEXT(1255$);
```

```
1254$    TRANSPORT:  LAGV,RStation7800;
```

```
1255$    TRANSPORT:  LAGV,LStation7800;
```

```
;
;
; Model statements for module: Decide 1486
;
```

```
1256$    BRANCH,      1:
          If,
            NL(RL16)&&NZ(RL17,1) &&NZ(RL17,2) &&NZ(RL17,3) &&NZ(RL17,4)
&&NZ(RL15,7) &&NZ(RL15,8) &&NZ(RL15,9) &&NZ(RL15,10) == 0,
            3250$,Yes;
          Else,3251$,Yes;
```

```
3250$    ASSIGN:    CheckGapL7750Speed10_35.NumberOut
True=CheckGapL7750Speed10_35.NumberOut True + 1:NEXT(1257$);
```

```
3251$    ASSIGN:    CheckGapL7750Speed10_35.NumberOut
False=CheckGapL7750Speed10_35.NumberOut False + 1:NEXT(1258$);
```

```
1257$    TRANSPORT:  LAGV,RStation7800;
```

```
1258$    TRANSPORT:  LAGV,LStation7800;
```

```
;
;
; Model statements for module: Decide 1488
;
```

```
1262$    BRANCH,      1:
          If,SpeedatRStation7700>75,1263$,Yes:
          If,SpeedatRStation7700 > 55 && SpeedatRStation7700 <= 75,1266$,Yes:
          If,SpeedatRStation7700>35 && SpeedatRStation7700 <=55,1269$,Yes:
          If,SpeedatRStation7700>10 && SpeedatRStation7700 <=35,1272$,Yes:
          Else,1275$,Yes;
```

```
;
;
; Model statements for module: Decide 1493
;
```

```
1275$    BRANCH,      1:
          If,NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL16,3)== 0,3254$,Yes:
          Else,3255$,Yes;
```

```
3254$    ASSIGN:    CheckGapL7750Speed10less.NumberOut
True=CheckGapL7750Speed10less.NumberOut True + 1:NEXT(1276$);
```

```
3255$    ASSIGN:    CheckGapL7750Speed10less.NumberOut
False=CheckGapL7750Speed10less.NumberOut False + 1:NEXT(1277$);
```

```

1276$    TRANSPORT:   LAGV,RStation7800;

1277$    TRANSPORT:   LAGV,LStation7800;

;
;
;   Model statements for module: Decide 1489
;
1263$    BRANCH,      1:
          If,
          NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)
&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL16,3)== 0,
          3256$,Yes:
          Else,3257$,Yes;
3256$    ASSIGN:      CheckGapL7750Speed75less.NumberOut
True=CheckGapL7750Speed75less.NumberOut True + 1:NEXT(1264$);

3257$    ASSIGN:      CheckGapL7750Speed75less.NumberOut
False=CheckGapL7750Speed75less.NumberOut False + 1:NEXT(1265$);

1264$    TRANSPORT:   LAGV,RStation7800;

1265$    TRANSPORT:   LAGV,LStation7800;

;
;
;   Model statements for module: Decide 1490
;
1266$    BRANCH,      1:
          If,
          NZ(RL13,1)&&NZ(RL13,2)
&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL16,3)== 0,
          3258$,Yes:
          Else,3259$,Yes;
3258$    ASSIGN:      CheckGapL7750Speed55_75less.NumberOut
True=CheckGapL7750Speed55_75less.NumberOut True + 1
          :NEXT(1267$);

3259$    ASSIGN:      CheckGapL7750Speed55_75less.NumberOut
False=CheckGapL7750Speed55_75less.NumberOut False + 1
          :NEXT(1268$);

1267$    TRANSPORT:   LAGV,RStation7800;

1268$    TRANSPORT:   LAGV,LStation7800;

;
;
;   Model statements for module: Decide 1491
;

```

```

1269$    BRANCH,    1:

If,NZ(RL13,2)&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL1
6,3)== 0,3260$,Yes:
    Else,3261$,Yes;
3260$    ASSIGN:    CheckGapL7750Speed35_55less.NumberOut
True=CheckGapL7750Speed35_55less.NumberOut True + 1
    :NEXT(1270$);

3261$    ASSIGN:    CheckGapL7750Speed35_55less.NumberOut
False=CheckGapL7750Speed35_55less.NumberOut False + 1
    :NEXT(1271$);

1270$    TRANSPORT:  LAGV,RStation7800;

1271$    TRANSPORT:  LAGV,LStation7800;

;
;
;    Model statements for module: Decide 1492
;
1272$    BRANCH,    1:
    If,NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL16,3)==
0,3262$,Yes:
    Else,3263$,Yes;
3262$    ASSIGN:    CheckGapL7750Speed10_35less.NumberOut
True=CheckGapL7750Speed10_35less.NumberOut True + 1
    :NEXT(1273$);

3263$    ASSIGN:    CheckGapL7750Speed10_35less.NumberOut
False=CheckGapL7750Speed10_35less.NumberOut False + 1
    :NEXT(1274$);

1273$    TRANSPORT:  LAGV,RStation7800;

1274$    TRANSPORT:  LAGV,LStation7800;

;
;
;    Model statements for module: Decide 1481
;
1235$    BRANCH,    1:
    If,SpeedatRStation7700 >= VTU(LAGVT,LTruck#),3264$,Yes:
    Else,3265$,Yes;
3264$    ASSIGN:    Decide 1481.NumberOut True=Decide 1481.NumberOut True +
1:NEXT(1278$);

3265$    ASSIGN:    Decide 1481.NumberOut False=Decide 1481.NumberOut False +
1:NEXT(1284$);

```

```

;
;
;   Model statements for module: Decide 1494
;
1278$   BRANCH,      1:
        If,SpeedatRStation7700>75,1279$,Yes:
        If,SpeedatRStation7700 > 55 && SpeedatRStation7700 <= 75,1280$,Yes:
        If,SpeedatRStation7700 >35 && SpeedatRStation7700 <=55,1281$,Yes:
        If,SpeedatRStation7700 >10 && SpeedatRStation7700 <=35,1282$,Yes:
        Else,1283$,Yes;

;
;
;   Model statements for module: Decide 1499
;
1283$   BRANCH,      1:
        If,NL(RL18)&&NL(RL19)&& NZ(RL15,7) && NZ(RL15,8) && NZ(RL15,9) &&
NZ(RL15,10) == 0,3268$,Yes:
        Else,3269$,Yes;
3268$   ASSIGN:      CheckGapL7750Speed10t.NumberOut
True=CheckGapL7750Speed10t.NumberOut True + 1:NEXT(1248$);

3269$   ASSIGN:      CheckGapL7750Speed10t.NumberOut
False=CheckGapL7750Speed10t.NumberOut False + 1:NEXT(1249$);

1248$   TRANSPORT:   LAGVT,RStation7800;

1249$   TRANSPORT:   LAGVT,LStation7800;

;
;
;   Model statements for module: Decide 1495
;
1279$   BRANCH,      1:
        If,
        NL(RL16) &&NL(RL17)&&NL(RL18)&&NL(RL19)&& NZ(RL20,1)&&
NZ(RL20,2)&& NZ(RL20,3)&& NZ(RL20,4)&& NZ(RL20,5) &&NZ(RL15,6) && NZ(RL15,7) &&
NZ(RL15,8) && NZ(RL15,9) && NZ(RL15,10) == 0,
        3270$,Yes:
        Else,3271$,Yes;
3270$   ASSIGN:      CheckGapL7750Speed75t.NumberOut
True=CheckGapL7750Speed75t.NumberOut True + 1:NEXT(1240$);

3271$   ASSIGN:      CheckGapL7750Speed75t.NumberOut
False=CheckGapL7750Speed75t.NumberOut False + 1:NEXT(1241$);

1240$   TRANSPORT:   LAGVT,RStation7800;

1241$   TRANSPORT:   LAGVT,LStation7800;

;

```

```

;
;   Model statements for module: Decide 1496
;
1280$   BRANCH,      1:
        If,
        NL(RL16) &&NL(RL17)&&NL(RL18)&&NL(RL19)&& NZ(RL15,6) && NZ(RL15,7)
&& NZ(RL15,8) && NZ(RL15,9) && NZ(RL15,10) == 0,
        3272$,Yes:
        Else,3273$,Yes;
3272$   ASSIGN:      CheckGapL7750Speed55_75t.NumberOut
True=CheckGapL7750Speed55_75t.NumberOut True + 1:NEXT(1243$);

3273$   ASSIGN:      CheckGapL7750Speed55_75t.NumberOut
False=CheckGapL7750Speed55_75t.NumberOut False + 1:NEXT(1242$);

1243$   TRANSPORT:   LAGVT,RStation7800;

1242$   TRANSPORT:   LAGVT,LStation7800;

;
;
;   Model statements for module: Decide 1497
;
1281$   BRANCH,      1:
        If,NL(RL17)&&NL(RL18)&&NL(RL19)&& NZ(RL15,7) && NZ(RL15,8) &&
NZ(RL15,9) && NZ(RL15,10) == 0,3274$,
        Yes:
        Else,3275$,Yes;
3274$   ASSIGN:      CheckGapL7750Speed35_55t.NumberOut
True=CheckGapL7750Speed35_55t.NumberOut True + 1:NEXT(1244$);

3275$   ASSIGN:      CheckGapL7750Speed35_55t.NumberOut
False=CheckGapL7750Speed35_55t.NumberOut False + 1:NEXT(1245$);

1244$   TRANSPORT:   LAGVT,RStation7800;

1245$   TRANSPORT:   LAGVT,LStation7800;

;
;
;   Model statements for module: Decide 1498
;
1282$   BRANCH,      1:
        If,NL(RL18)&&NL(RL19)&& NZ(RL15,7) && NZ(RL15,8) && NZ(RL15,9) &&
NZ(RL15,10) == 0,3276$,Yes:
        Else,3277$,Yes;
3276$   ASSIGN:      CheckGapL7750Speed10_35t.NumberOut
True=CheckGapL7750Speed10_35t.NumberOut True + 1:NEXT(1246$);

3277$   ASSIGN:      CheckGapL7750Speed10_35t.NumberOut
False=CheckGapL7750Speed10_35t.NumberOut False + 1:NEXT(1247$);

```

```

1246$    TRANSPORT:   LAGVT,RStation7800;

1247$    TRANSPORT:   LAGVT,LStation7800;

;
;
;   Model statements for module: Decide 1500
;
1284$    BRANCH,      1:
                If,SpeedatRStation7700>75,1285$,Yes:
                If,SpeedatRStation7700 > 55 && SpeedatRStation7700<= 75,1286$,Yes:
                If,SpeedatRStation7700 >35 && SpeedatRStation7700 <=55,1287$,Yes:
                If,SpeedatRStation7700 >10 && SpeedatRStation7700 <=35,1288$,Yes:
                Else,1289$,Yes;

;
;
;   Model statements for module: Decide 1505
;
1289$    BRANCH,      1:
                If,NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)==
0,3280$,Yes:
                Else,3281$,Yes;
3280$    ASSIGN:      CheckGapL7750Speed10lesst.NumberOut
True=CheckGapL7750Speed10lesst.NumberOut True + 1:NEXT(1298$);

3281$    ASSIGN:      CheckGapL7750Speed10lesst.NumberOut
False=CheckGapL7750Speed10lesst.NumberOut False + 1:NEXT(1299$);

1298$    TRANSPORT:   LAGVT,RStation7800;

1299$    TRANSPORT:   LAGVT,LStation7800;

;
;
;   Model statements for module: Decide 1501
;
1285$    BRANCH,      1:
                If,

NZ(RL11,1)&&NZ(RL11,2)&&NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)
&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL16,3)== 0,
                3282$,Yes:
                Else,3283$,Yes;
3282$    ASSIGN:      CheckGapL7750Speed75lesst.NumberOut
True=CheckGapL7750Speed75lesst.NumberOut True + 1:NEXT(1290$);

3283$    ASSIGN:      CheckGapL7750Speed75lesst.NumberOut
False=CheckGapL7750Speed75lesst.NumberOut False + 1:NEXT(1291$);

```



```

1290$    TRANSPORT:   LAGVT,RStation7800;

1291$    TRANSPORT:   LAGVT,LStation7800;

;
;
;   Model statements for module: Decide 1502
;
1286$    BRANCH,      1:
          If,
          NZ(RL12,1)&&NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)
&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL16,3)== 0,
          3284$,Yes:
          Else,3285$,Yes;
3284$    ASSIGN:      CheckGapL7750Speed55_75lesst.NumberOut
True=CheckGapL7750Speed55_75lesst.NumberOut True + 1
          :NEXT(1292$);

3285$    ASSIGN:      CheckGapL7750Speed55_75lesst.NumberOut
False=CheckGapL7750Speed55_75lesst.NumberOut False + 1
          :NEXT(1293$);

1292$    TRANSPORT:   LAGVT,RStation7800;

1293$    TRANSPORT:   LAGVT,LStation7800;

;
;
;   Model statements for module: Decide 1503
;
1287$    BRANCH,      1:
          If,
          NZ(RL12,2)&&NZ(RL13,1)&&NZ(RL13,2)
&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL16,3)== 0,
          3286$,Yes:
          Else,3287$,Yes;
3286$    ASSIGN:      CheckGapL7750Speed35_55lesst.NumberOut
True=CheckGapL7750Speed35_55lesst.NumberOut True + 1
          :NEXT(1294$);

3287$    ASSIGN:      CheckGapL7750Speed35_55lesst.NumberOut
False=CheckGapL7750Speed35_55lesst.NumberOut False + 1
          :NEXT(1295$);

1294$    TRANSPORT:   LAGVT,RStation7800;

1295$    TRANSPORT:   LAGVT,LStation7800;

;
;

```

```

;   Model statements for module: Decide 1504
;
1288$   BRANCH,      1:
          If,NZ(RL13,2)
&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NZ(RL16,1)&&NZ(RL16,2)&&NZ(RL16,3)==
0,3288$,Yes:
          Else,3289$,Yes;
3288$   ASSIGN:      CheckGapL7750Speed10_35lesst.NumberOut
True=CheckGapL7750Speed10_35lesst.NumberOut True + 1
          :NEXT(1296$);

3289$   ASSIGN:      CheckGapL7750Speed10_35lesst.NumberOut
False=CheckGapL7750Speed10_35lesst.NumberOut False + 1
          :NEXT(1297$);

1296$   TRANSPORT:   LAGVT,RStation7800;

1297$   TRANSPORT:   LAGVT,LStation7800;

;
;
;   Model statements for module: Station 118
;

1300$   STATION,      LStation7800;
3292$   DELAY:        0.0,,VA:NEXT(1301$);

;
;
;   Model statements for module: Decide 1506
;
1301$   BRANCH,      1:
          If,Entity.Type==LCar,3293$,Yes:
          Else,3294$,Yes;
3293$   ASSIGN:      checkvehicletype7800.NumberOut True=checkvehicletype7800.NumberOut
True + 1:NEXT(1302$);

3294$   ASSIGN:      checkvehicletype7800.NumberOut False=checkvehicletype7800.NumberOut
False + 1:NEXT(1303$);

;
;
;   Model statements for module: Decide 1507
;
1302$   BRANCH,      1:
          If,SpeedatRStation7750 >= VTU(LAGV,LCar#),3295$,Yes:
          Else,3296$,Yes;
3295$   ASSIGN:      Decide 1507.NumberOut True=Decide 1507.NumberOut True +
1:NEXT(1304$);

```

```

3296$    ASSIGN:    Decide 1507.NumberOut False=Decide 1507.NumberOut False +
1:NEXT(1330$);

;
;
;    Model statements for module: Decide 1509
;
1304$    BRANCH,    1:
                If,SpeedatRStation7750>75,1305$,Yes:
                If,SpeedatRStation7750 > 55 && SpeedatRStation7750<= 75,1318$,Yes:
                If,SpeedatRStation7750 >35 && SpeedatRStation7750<=55,1321$,Yes:
                If,SpeedatRStation7750>10 && SpeedatRStation7750 <=35,1324$,Yes:
                Else,1327$,Yes;

;
;
;    Model statements for module: Decide 1514
;
1327$    BRANCH,    1:
                If,NL(RL17)&&NZ(RL18,1) &&NZ(RL16,10) == 0,3299$,Yes:
                Else,3300$,Yes;
3299$    ASSIGN:    CheckGapL7800Speed10.NumberOut
True=CheckGapL7800Speed10.NumberOut True + 1:NEXT(1328$);

3300$    ASSIGN:    CheckGapL7800Speed10.NumberOut
False=CheckGapL7800Speed10.NumberOut False + 1:NEXT(1329$);

1328$    TRANSPORT:  LAGV,RStation7850;

1329$    TRANSPORT:  LAGV,LStation7850;

;
;
;    Model statements for module: Decide 1510
;
1305$    BRANCH,    1:
                If,
                NL(RL17)&&NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL20,2)
&&NZ(RL20,3) &&NZ(RL20,4)&& NZ(RL16,6) && NZ(RL16,7) && NZ(RL16,8) && NZ(RL16,9)
&& NZ(RL16,10) == 0,
                3301$,Yes:
                Else,3302$,Yes;
3301$    ASSIGN:    CheckGapL7800Speed75.NumberOut
True=CheckGapL7800Speed75.NumberOut True + 1:NEXT(1306$);

3302$    ASSIGN:    CheckGapL7800Speed75.NumberOut
False=CheckGapL7800Speed75.NumberOut False + 1:NEXT(1307$);

1306$    TRANSPORT:  LAGV,RStation7850;

1307$    TRANSPORT:  LAGV,LStation7850;

```

```

;
;
;   Model statements for module: Decide 1511
;
1318$   BRANCH,      1:
          If,NL(RL17)&&NL(RL18)&&NL(RL19)&&NZ(RL16,7) && NZ(RL16,8) &&
NZ(RL16,9) && NZ(RL16,10) == 0,3303$,
          Yes:
          Else,3304$,Yes;
3303$   ASSIGN:      CheckGapL7800Speed55_75.NumberOut
True=CheckGapL7800Speed55_75.NumberOut True + 1:NEXT(1319$);

3304$   ASSIGN:      CheckGapL7800Speed55_75.NumberOut
False=CheckGapL7800Speed55_75.NumberOut False + 1:NEXT(1320$);

1319$   TRANSPORT:   LAGV,RStation7850;

1320$   TRANSPORT:   LAGV,LStation7850;

;
;
;   Model statements for module: Decide 1512
;
1321$   BRANCH,      1:
          If,NL(RL18)&&NL(RL19)&&NZ(RL16,7) && NZ(RL16,8) && NZ(RL16,9) &&
NZ(RL16,10) == 0,3305$,Yes:
          Else,3306$,Yes;
3305$   ASSIGN:      CheckGapL7800Speed35_55.NumberOut
True=CheckGapL7800Speed35_55.NumberOut True + 1:NEXT(1322$);

3306$   ASSIGN:      CheckGapL7800Speed35_55.NumberOut
False=CheckGapL7800Speed35_55.NumberOut False + 1:NEXT(1323$);

1322$   TRANSPORT:   LAGV,RStation7850;

1323$   TRANSPORT:   LAGV,LStation7850;

;
;
;   Model statements for module: Decide 1513
;
1324$   BRANCH,      1:
          If,
          NL(RL17)&&NZ(RL18,1) &&NZ(RL18,2) &&NZ(RL18,3) &&NZ(RL18,4)
&&NZ(RL16,7) &&NZ(RL16,8) &&NZ(RL16,9) &&NZ(RL16,10) == 0,
          3307$,Yes:
          Else,3308$,Yes;
3307$   ASSIGN:      CheckGapL7800Speed10_35.NumberOut
True=CheckGapL7800Speed10_35.NumberOut True + 1:NEXT(1325$);

```

3308\$ ASSIGN: CheckGapL7800Speed10_35.NumberOut
False=CheckGapL7800Speed10_35.NumberOut False + 1:NEXT(1326\$);

1325\$ TRANSPORT: LAGV,RStation7850;

1326\$ TRANSPORT: LAGV,LStation7850;

;
;
; Model statements for module: Decide 1515
;

1330\$ BRANCH, 1:
 If,SpeedatRStation7750>75,1331\$,Yes:
 If,SpeedatRStation7750 > 55 && SpeedatRStation7750 <= 75,1334\$,Yes:
 If,SpeedatRStation7750>35 && SpeedatRStation7750 <=55,1337\$,Yes:
 If,SpeedatRStation7750>10 && SpeedatRStation7750 <=35,1340\$,Yes:
 Else,1343\$,Yes;

;
;
; Model statements for module: Decide 1520
;

1343\$ BRANCH, 1:
 If,NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)== 0,3311\$,Yes:
 Else,3312\$,Yes;

3311\$ ASSIGN: CheckGapL7800Speed10less.NumberOut
True=CheckGapL7800Speed10less.NumberOut True + 1:NEXT(1344\$);

3312\$ ASSIGN: CheckGapL7800Speed10less.NumberOut
False=CheckGapL7800Speed10less.NumberOut False + 1:NEXT(1345\$);

1344\$ TRANSPORT: LAGV,RStation7850;

1345\$ TRANSPORT: LAGV,LStation7850;

;
;
; Model statements for module: Decide 1516
;

1331\$ BRANCH, 1:
 If,
 NZ(RL13,1)&&NZ(RL13,2)
 &&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)== 0,
 3313\$,Yes:
 Else,3314\$,Yes;

3313\$ ASSIGN: CheckGapL7800Speed75less.NumberOut
True=CheckGapL7800Speed75less.NumberOut True + 1:NEXT(1332\$);

```
3314$    ASSIGN:    CheckGapL7800Speed75less.NumberOut
False=CheckGapL7800Speed75less.NumberOut False + 1:NEXT(1333$);
```

```
1332$    TRANSPORT:  LAGV,RStation7850;
```

```
1333$    TRANSPORT:  LAGV,LStation7850;
```

```
;  
;
```

```
;    Model statements for module: Decide 1517
```

```
;
```

```
1334$    BRANCH,    1:
```

```
If,NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)== 0,3315$,Yes:
```

```
    Else,3316$,Yes;
```

```
3315$    ASSIGN:    CheckGapL7800Speed55_75less.NumberOut
True=CheckGapL7800Speed55_75less.NumberOut True + 1
:NEXT(1335$);
```

```
3316$    ASSIGN:    CheckGapL7800Speed55_75less.NumberOut
False=CheckGapL7800Speed55_75less.NumberOut False + 1
:NEXT(1336$);
```

```
1335$    TRANSPORT:  LAGV,RStation7850;
```

```
1336$    TRANSPORT:  LAGV,LStation7850;
```

```
;  
;
```

```
;    Model statements for module: Decide 1518
```

```
;
```

```
1337$    BRANCH,    1:
```

```
If,NZ(RL14,2)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)== 0,3317$,Yes:
```

```
    Else,3318$,Yes;
```

```
3317$    ASSIGN:    CheckGapL7800Speed35_55less.NumberOut
True=CheckGapL7800Speed35_55less.NumberOut True + 1
:NEXT(1338$);
```

```
3318$    ASSIGN:    CheckGapL7800Speed35_55less.NumberOut
False=CheckGapL7800Speed35_55less.NumberOut False + 1
:NEXT(1339$);
```

```
1338$    TRANSPORT:  LAGV,RStation7850;
```

```
1339$    TRANSPORT:  LAGV,LStation7850;
```

```
;
```

```

;
;   Model statements for module: Decide 1519
;
1340$   BRANCH,      1:
           If,NZ(RL15,2)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)==
0,3319$,Yes:
           Else,3320$,Yes;
3319$   ASSIGN:      CheckGapL7800Speed10_35less.NumberOut
True=CheckGapL7800Speed10_35less.NumberOut True + 1
           :NEXT(1341$);

3320$   ASSIGN:      CheckGapL7800Speed10_35less.NumberOut
False=CheckGapL7800Speed10_35less.NumberOut False + 1
           :NEXT(1342$);

1341$   TRANSPORT:   LAGV,RStation7850;

1342$   TRANSPORT:   LAGV,LStation7850;

;
;
;   Model statements for module: Decide 1508
;
1303$   BRANCH,      1:
           If,SpeedatRStation7750 >= VTU(LAGVT,LTruck#),3321$,Yes:
           Else,3322$,Yes;
3321$   ASSIGN:      Decide 1508.NumberOut True=Decide 1508.NumberOut True +
1:NEXT(1346$);

3322$   ASSIGN:      Decide 1508.NumberOut False=Decide 1508.NumberOut False +
1:NEXT(1352$);

;
;
;   Model statements for module: Decide 1521
;
1346$   BRANCH,      1:
           If,SpeedatRStation7750>75,1347$,Yes:
           If,SpeedatRStation7750 > 55 && SpeedatRStation7750 <= 75,1348$,Yes:
           If,SpeedatRStation7750 >35 && SpeedatRStation7750 <=55,1349$,Yes:
           If,SpeedatRStation7750 >10 && SpeedatRStation7750 <=35,1350$,Yes:
           Else,1351$,Yes;

;
;
;   Model statements for module: Decide 1526
;
1351$   BRANCH,      1:
           If,NL(RL17)&&NL(RL18)&&NZ(RL16,8) && NZ(RL16,9) && NZ(RL16,10) ==
0,3325$,Yes:
           Else,3326$,Yes;

```

3325\$ ASSIGN: CheckGapL7800Speed10t.NumberOut
True=CheckGapL7800Speed10t.NumberOut True + 1:NEXT(1316\$);

3326\$ ASSIGN: CheckGapL7800Speed10t.NumberOut
False=CheckGapL7800Speed10t.NumberOut False + 1:NEXT(1317\$);

1316\$ TRANSPORT: LAGVT,RStation7850;

1317\$ TRANSPORT: LAGVT,LStation7850;

;
;

; Model statements for module: Decide 1522

;

1347\$ BRANCH, 1:

 If,

 NL(RL17)&&NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL20,2)

&&NZ(RL20,3)

&&NZ(RL20,4)&&NZ(RL20,5)&&NZ(RL20,6)&&NZ(RL20,7)&&NZ(RL20,8)&&NZ(RL20,9)&&NZ(RL20,10)&& NZ(RL16,6) && NZ(RL16,7) && NZ(RL16,8) && NZ(RL16,9) && NZ(RL16,10) == 0,

 3327\$,Yes;

 Else,3328\$,Yes;

3327\$ ASSIGN: CheckGapL7800Speed75t.NumberOut
True=CheckGapL7800Speed75t.NumberOut True + 1:NEXT(1308\$);

3328\$ ASSIGN: CheckGapL7800Speed75t.NumberOut
False=CheckGapL7800Speed75t.NumberOut False + 1:NEXT(1309\$);

1308\$ TRANSPORT: LAGVT,RStation7850;

1309\$ TRANSPORT: LAGVT,LStation7850;

;
;

; Model statements for module: Decide 1523

;

1348\$ BRANCH, 1:

 If,

 NL(RL17)&&NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL20,2)

&&NZ(RL20,3) &&NZ(RL20,4)&&NZ(RL20,5)&& NZ(RL16,8) && NZ(RL16,9) && NZ(RL16,10) == 0,

 3329\$,Yes;

 Else,3330\$,Yes;

3329\$ ASSIGN: CheckGapL7800Speed55_75t.NumberOut
True=CheckGapL7800Speed55_75t.NumberOut True + 1:NEXT(1311\$);

3330\$ ASSIGN: CheckGapL7800Speed55_75t.NumberOut
False=CheckGapL7800Speed55_75t.NumberOut False + 1:NEXT(1310\$);

1311\$ TRANSPORT: LAGVT,RStation7850;


```

1310$    TRANSPORT:   LAGVT,LStation7850;

;
;
;   Model statements for module: Decide 1524
;
1349$    BRANCH,      1:
                If,NL(RL17)&&NL(RL18)&&NL(RL19)&&NZ(RL20,1)&& NZ(RL16,8) &&
NZ(RL16,9) && NZ(RL16,10) == 0,3331$,
                Yes:
                Else,3332$,Yes;
3331$    ASSIGN:      CheckGapL7800Speed35_55t.NumberOut
True=CheckGapL7800Speed35_55t.NumberOut True + 1:NEXT(1312$);

3332$    ASSIGN:      CheckGapL7800Speed35_55t.NumberOut
False=CheckGapL7800Speed35_55t.NumberOut False + 1:NEXT(1313$);

1312$    TRANSPORT:   LAGVT,RStation7850;

1313$    TRANSPORT:   LAGVT,LStation7850;

;
;
;   Model statements for module: Decide 1525
;
1350$    BRANCH,      1:
                If,NL(RL17)&&NL(RL18)&&NZ(RL16,8) && NZ(RL16,9) && NZ(RL16,10) ==
0,3333$,Yes:
                Else,3334$,Yes;
3333$    ASSIGN:      CheckGapL7800Speed10_35t.NumberOut
True=CheckGapL7800Speed10_35t.NumberOut True + 1:NEXT(1314$);

3334$    ASSIGN:      CheckGapL7800Speed10_35t.NumberOut
False=CheckGapL7800Speed10_35t.NumberOut False + 1:NEXT(1315$);

1314$    TRANSPORT:   LAGVT,RStation7850;

1315$    TRANSPORT:   LAGVT,LStation7850;

;
;
;   Model statements for module: Decide 1527
;
1352$    BRANCH,      1:
                If,SpeedatRStation7750>75,1353$,Yes:
                If,SpeedatRStation7750 > 55 && SpeedatRStation7750<= 75,1354$,Yes:
                If,SpeedatRStation7750 >35 && SpeedatRStation7750 <=55,1355$,Yes:
                If,SpeedatRStation7750 >10 && SpeedatRStation7750 <=35,1356$,Yes:
                Else,1357$,Yes;

```

```

;
;
;   Model statements for module: Decide 1532
;
1357$   BRANCH,      1:
           If,NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)==
0,3337$,Yes:
           Else,3338$,Yes;
3337$   ASSIGN:      CheckGapL7800Speed10lesst.NumberOut
True=CheckGapL7800Speed10lesst.NumberOut True + 1:NEXT(1366$);

3338$   ASSIGN:      CheckGapL7800Speed10lesst.NumberOut
False=CheckGapL7800Speed10lesst.NumberOut False + 1:NEXT(1367$);

1366$   TRANSPORT:   LAGVT,RStation7850;

1367$   TRANSPORT:   LAGVT,LStation7850;

;
;
;   Model statements for module: Decide 1528
;
1353$   BRANCH,      1:
           If,
           NZ(RL12,1)&&NZ(RL12,2) &&NZ(RL13,1)&&NZ(RL13,2)
&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL1
7,3)== 0,
           3339$,Yes:
           Else,3340$,Yes;
3339$   ASSIGN:      CheckGapL7800Speed75lesst.NumberOut
True=CheckGapL7800Speed75lesst.NumberOut True + 1:NEXT(1358$);

3340$   ASSIGN:      CheckGapL7800Speed75lesst.NumberOut
False=CheckGapL7800Speed75lesst.NumberOut False + 1:NEXT(1359$);

1358$   TRANSPORT:   LAGVT,RStation7850;

1359$   TRANSPORT:   LAGVT,LStation7850;

;
;
;   Model statements for module: Decide 1529
;
1354$   BRANCH,      1:
           If,
           NZ(RL13,1)&&NZ(RL13,2)
&&NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL1
7,3)== 0,
           3341$,Yes:
           Else,3342$,Yes;

```

```

3341$    ASSIGN:    CheckGapL7800Speed55_75lesst.NumberOut
True=CheckGapL7800Speed55_75lesst.NumberOut True + 1
          :NEXT(1360$);

```

```

3342$    ASSIGN:    CheckGapL7800Speed55_75lesst.NumberOut
False=CheckGapL7800Speed55_75lesst.NumberOut False + 1
          :NEXT(1361$);

```

```

1360$    TRANSPORT:  LAGVT,RStation7850;

```

```

1361$    TRANSPORT:  LAGVT,LStation7850;

```

```

;
;
;   Model statements for module: Decide 1530
;
1355$    BRANCH,      1:

```

```

If,NZ(RL14,1)&&NZ(RL14,2)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,
3)== 0,3343$,Yes:

```

```

    Else,3344$,Yes;

```

```

3343$    ASSIGN:    CheckGapL7800Speed35_55lesst.NumberOut
True=CheckGapL7800Speed35_55lesst.NumberOut True + 1
          :NEXT(1362$);

```

```

3344$    ASSIGN:    CheckGapL7800Speed35_55lesst.NumberOut
False=CheckGapL7800Speed35_55lesst.NumberOut False + 1
          :NEXT(1363$);

```

```

1362$    TRANSPORT:  LAGVT,RStation7850;

```

```

1363$    TRANSPORT:  LAGVT,LStation7850;

```

```

;
;
;   Model statements for module: Decide 1531
;
1356$    BRANCH,      1:

```

```

If,NZ(RL14,2)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)==
0,3345$,Yes:

```

```

    Else,3346$,Yes;

```

```

3345$    ASSIGN:    CheckGapL7800Speed10_35lesst.NumberOut
True=CheckGapL7800Speed10_35lesst.NumberOut True + 1
          :NEXT(1364$);

```

```

3346$    ASSIGN:    CheckGapL7800Speed10_35lesst.NumberOut
False=CheckGapL7800Speed10_35lesst.NumberOut False + 1
          :NEXT(1365$);

```

```

1364$    TRANSPORT:  LAGVT,RStation7850;

```

1365\$ TRANSPORT: LAGVT,LStation7850;

;
;
;
; Model statements for module: Station 119
;

1368\$ STATION, LStation7850;
3349\$ DELAY: 0.0,,VA:NEXT(1369\$);

;
;
;
; Model statements for module: Decide 1533
;

1369\$ BRANCH, 1:
 If,Entity.Type==LCar,3350\$,Yes:
 Else,3351\$,Yes;
3350\$ ASSIGN: checkvehicletype7850.NumberOut True=checkvehicletype7850.NumberOut
True + 1:NEXT(1370\$);

3351\$ ASSIGN: checkvehicletype7850.NumberOut False=checkvehicletype7850.NumberOut
False + 1:NEXT(1371\$);

;
;
;
; Model statements for module: Decide 1534
;

1370\$ BRANCH, 1:
 If,SpeedatRStation7800 >= VTU(LAGV,LCar#),3352\$,Yes:
 Else,3353\$,Yes;
3352\$ ASSIGN: Decide 1534.NumberOut True=Decide 1534.NumberOut True +
1:NEXT(1372\$);

3353\$ ASSIGN: Decide 1534.NumberOut False=Decide 1534.NumberOut False +
1:NEXT(1398\$);

;
;
;
; Model statements for module: Decide 1536
;

1372\$ BRANCH, 1:
 If,SpeedatRStation7800>75,1373\$,Yes:
 If,SpeedatRStation7800 > 55 && SpeedatRStation7800<= 75,1386\$,Yes:
 If,SpeedatRStation7800 >35 && SpeedatRStation7800<=55,1389\$,Yes:
 If,SpeedatRStation7800>10 && SpeedatRStation7800 <=35,1392\$,Yes:
 Else,1395\$,Yes;

;

```

;
;   Model statements for module: Decide 1541
;
1395$   BRANCH,      1:
           If,NL(RL18) && NZ(RL17,9) && NZ(RL17,10) == 0,3356$,Yes:
           Else,3357$,Yes;
3356$   ASSIGN:      CheckGapL7850Speed10.NumberOut
True=CheckGapL7850Speed10.NumberOut True + 1:NEXT(1396$);

3357$   ASSIGN:      CheckGapL7850Speed10.NumberOut
False=CheckGapL7850Speed10.NumberOut False + 1:NEXT(1397$);

1396$   TRANSPORT:   LAGV,RStation7900;

1397$   TRANSPORT:   LAGV,LStation7900;

;
;
;   Model statements for module: Decide 1537
;
1373$   BRANCH,      1:
           If,
           NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL20,2) &&NZ(RL20,3)
&&NZ(RL20,4)&&NZ(RL20,5)&&NZ(RL20,6)&&NZ(RL20,7)&&NZ(RL20,8)&& NZ(RL17,6) &&
NZ(RL17,7) && NZ(RL17,8) && NZ(RL17,9) && NZ(RL17,10) == 0,
           3358$,Yes:
           Else,3359$,Yes;
3358$   ASSIGN:      CheckGapL7850Speed75.NumberOut
True=CheckGapL7850Speed75.NumberOut True + 1:NEXT(1374$);

3359$   ASSIGN:      CheckGapL7850Speed75.NumberOut
False=CheckGapL7850Speed75.NumberOut False + 1:NEXT(1375$);

1374$   TRANSPORT:   LAGV,RStation7900;

1375$   TRANSPORT:   LAGV,LStation7900;

;
;
;   Model statements for module: Decide 1538
;
1386$   BRANCH,      1:
           If,
           NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL20,2) &&NZ(RL20,3)
&&NZ(RL20,4)&&NZ(RL17,6) && NZ(RL17,7) && NZ(RL17,8) && NZ(RL17,9) && NZ(RL17,10)
== 0,
           3360$,Yes:
           Else,3361$,Yes;
3360$   ASSIGN:      CheckGapL7850Speed55_75.NumberOut
True=CheckGapL7850Speed55_75.NumberOut True + 1:NEXT(1387$);

```

```

3361$    ASSIGN:    CheckGapL7850Speed55_75.NumberOut
False=CheckGapL7850Speed55_75.NumberOut False + 1:NEXT(1388$);

1387$    TRANSPORT:  LAGV,RStation7900;

1388$    TRANSPORT:  LAGV,LStation7900;

;
;
;   Model statements for module: Decide 1539
;
1389$    BRANCH,      1:
          If,
              NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL17,6) && NZ(RL17,7) &&
NZ(RL17,8) && NZ(RL17,9) && NZ(RL17,10) == 0,
              3362$,Yes:
              Else,3363$,Yes;
3362$    ASSIGN:    CheckGapL7850Speed35_55.NumberOut
True=CheckGapL7850Speed35_55.NumberOut True + 1:NEXT(1390$);

3363$    ASSIGN:    CheckGapL7850Speed35_55.NumberOut
False=CheckGapL7850Speed35_55.NumberOut False + 1:NEXT(1391$);

1390$    TRANSPORT:  LAGV,RStation7900;

1391$    TRANSPORT:  LAGV,LStation7900;

;
;
;   Model statements for module: Decide 1540
;
1392$    BRANCH,      1:
          If,
              NL(RL18)&&NZ(RL19,1) &&NZ(RL19,2) &&NZ(RL19,3) &&NZ(RL19,4)
&&NZ(RL19,5) &&NZ(RL19,6) && NZ(RL17,9) && NZ(RL17,10) == 0,
              3364$,Yes:
              Else,3365$,Yes;
3364$    ASSIGN:    CheckGapL7850Speed10_35.NumberOut
True=CheckGapL7850Speed10_35.NumberOut True + 1:NEXT(1393$);

3365$    ASSIGN:    CheckGapL7850Speed10_35.NumberOut
False=CheckGapL7850Speed10_35.NumberOut False + 1:NEXT(1394$);

1393$    TRANSPORT:  LAGV,RStation7900;

1394$    TRANSPORT:  LAGV,LStation7900;

;
;
;   Model statements for module: Decide 1542

```

```

;
1398$    BRANCH,    1:
          If,SpeedatRStation7800>75,1399$,Yes:
          If,SpeedatRStation7800 > 55 && SpeedatRStation7800 <= 75,1402$,Yes:
          If,SpeedatRStation7800>35 && SpeedatRStation7800 <=55,1405$,Yes:
          If,SpeedatRStation7800>10 && SpeedatRStation7800 <=35,1408$,Yes:
          Else,1411$,Yes;

;
;
;    Model statements for module: Decide 1547
;
1411$    BRANCH,    1:
          If,NL(RL17)&&NZ(RL18,1)&&NZ(RL18,2)&&NZ(RL18,3)== 0,3368$,Yes:
          Else,3369$,Yes;
3368$    ASSIGN:    CheckGapL7850Speed10less.NumberOut
True=CheckGapL7850Speed10less.NumberOut True + 1:NEXT(1412$);

3369$    ASSIGN:    CheckGapL7850Speed10less.NumberOut
False=CheckGapL7850Speed10less.NumberOut False + 1:NEXT(1413$);

1412$    TRANSPORT:  LAGV,RStation7900;

1413$    TRANSPORT:  LAGV,LStation7900;

;
;
;    Model statements for module: Decide 1543
;
1399$    BRANCH,    1:
          If,NZ(RL14,1)&&NZ(RL14,2)
&&NL(RL15)&&NL(RL16)&&NL(RL17)&&NZ(RL18,1)&&NZ(RL18,2)&&NZ(RL18,3)== 0,
          3370$,Yes:
          Else,3371$,Yes;
3370$    ASSIGN:    CheckGapL7850Speed75less.NumberOut
True=CheckGapL7850Speed75less.NumberOut True + 1:NEXT(1400$);

3371$    ASSIGN:    CheckGapL7850Speed75less.NumberOut
False=CheckGapL7850Speed75less.NumberOut False + 1:NEXT(1401$);

1400$    TRANSPORT:  LAGV,RStation7900;

1401$    TRANSPORT:  LAGV,LStation7900;

;
;
;    Model statements for module: Decide 1544
;
1402$    BRANCH,    1:

```

```
If,NL(RL15)&&NL(RL16)&&NL(RL17)&&NZ(RL18,1)&&NZ(RL18,2)&&NZ(RL18,3)==
0,3372$,Yes;
```

```
Else,3373$,Yes;
```

```
3372$    ASSIGN:    CheckGapL7850Speed55_75less.NumberOut
True=CheckGapL7850Speed55_75less.NumberOut True + 1
: NEXT(1403$);
```

```
3373$    ASSIGN:    CheckGapL7850Speed55_75less.NumberOut
False=CheckGapL7850Speed55_75less.NumberOut False + 1
: NEXT(1404$);
```

```
1403$    TRANSPORT:  LAGV,RStation7900;
```

```
1404$    TRANSPORT:  LAGV,LStation7900;
```

```
;
;
;
;
```

```
Model statements for module: Decide 1545
```

```
1405$    BRANCH,      1:
If,
```

```
NZ(RL15,6)&&NZ(RL15,7)&&NZ(RL15,8)&&NZ(RL15,9)&&NZ(RL15,10)&&NL(RL16)&&NL(RL17)
)&&NZ(RL18,1)&&NZ(RL18,2)&&NZ(RL18,3)== 0,
```

```
3374$,Yes;
```

```
Else,3375$,Yes;
```

```
3374$    ASSIGN:    CheckGapL7850Speed35_55less.NumberOut
True=CheckGapL7850Speed35_55less.NumberOut True + 1
: NEXT(1406$);
```

```
3375$    ASSIGN:    CheckGapL7850Speed35_55less.NumberOut
False=CheckGapL7850Speed35_55less.NumberOut False + 1
: NEXT(1407$);
```

```
1406$    TRANSPORT:  LAGV,RStation7900;
```

```
1407$    TRANSPORT:  LAGV,LStation7900;
```

```
;
;
;
;
```

```
Model statements for module: Decide 1546
```

```
1408$    BRANCH,      1:
If,
```

```
NZ(RL16,6)&&NZ(RL16,7)&&NZ(RL16,8)&&NZ(RL16,9)&&NZ(RL16,10)&&NL(RL17)&&NZ(RL18
,1)&&NZ(RL18,2)&&NZ(RL18,3)== 0,
```

```
3376$,Yes;
```

```
Else,3377$,Yes;
```



```

3376$    ASSIGN:    CheckGapL7850Speed10_35less.NumberOut
True=CheckGapL7850Speed10_35less.NumberOut True + 1
          :NEXT(1409$);

```

```

3377$    ASSIGN:    CheckGapL7850Speed10_35less.NumberOut
False=CheckGapL7850Speed10_35less.NumberOut False + 1
          :NEXT(1410$);

```

```

1409$    TRANSPORT:  LAGV,RStation7900;

```

```

1410$    TRANSPORT:  LAGV,LStation7900;

```

```

;
;
;   Model statements for module: Decide 1535
;
1371$    BRANCH,      1:
          If,SpeedatRStation7800 >= VTU(LAGVT,LTruck#),3378$,Yes:
          Else,3379$,Yes;
3378$    ASSIGN:      Decide 1535.NumberOut True=Decide 1535.NumberOut True +
1:NEXT(1414$);
3379$    ASSIGN:      Decide 1535.NumberOut False=Decide 1535.NumberOut False +
1:NEXT(1420$);

```

```

;
;
;   Model statements for module: Decide 1548
;
1414$    BRANCH,      1:
          If,SpeedatRStation7800>75,1415$,Yes:
          If,SpeedatRStation7800 > 55 && SpeedatRStation7800 <= 75,1416$,Yes:
          If,SpeedatRStation7800 >35 && SpeedatRStation7800 <=55,1417$,Yes:
          If,SpeedatRStation7800 >10 && SpeedatRStation7800 <=35,1418$,Yes:
          Else,1419$,Yes;

```

```

;
;
;   Model statements for module: Decide 1553
;
1419$    BRANCH,      1:
          If,NL(RL18)&&NL(RL19)&&NZ(RL17,9) && NZ(RL17,10) == 0,3382$,Yes:
          Else,3383$,Yes;
3382$    ASSIGN:      CheckGapL7850Speed10t.NumberOut
True=CheckGapL7850Speed10t.NumberOut True + 1:NEXT(1384$);
3383$    ASSIGN:      CheckGapL7850Speed10t.NumberOut
False=CheckGapL7850Speed10t.NumberOut False + 1:NEXT(1385$);
1384$    TRANSPORT:  LAGVT,RStation7900;

```

```

1385$    TRANSPORT:   LAGVT,LStation7900;

;
;
;   Model statements for module: Decide 1549
;
1415$    BRANCH,      1:
          If,
              NL(RL18)&&NL(RL19)&&NL(RL20) && NZ(RL17,6) && NZ(RL17,7) &&
NZ(RL17,8) && NZ(RL17,9) && NZ(RL17,10) == 0,
              3384$,Yes:
              Else,3385$,Yes;
3384$    ASSIGN:      CheckGapL7850Speed75t.NumberOut
True=CheckGapL7850Speed75t.NumberOut True + 1:NEXT(1376$);

3385$    ASSIGN:      CheckGapL7850Speed75t.NumberOut
False=CheckGapL7850Speed75t.NumberOut False + 1:NEXT(1377$);

1376$    TRANSPORT:   LAGVT,RStation7900;

1377$    TRANSPORT:   LAGVT,LStation7900;

;
;
;   Model statements for module: Decide 1550
;
1416$    BRANCH,      1:
          If,
              NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL20,2) &&NZ(RL20,3)
&&NZ(RL20,4)&&NZ(RL20,5)&&NZ(RL20,6)&&NZ(RL20,7)&&NZ(RL20,8)&&NZ(RL20,9)&&
NZ(RL17,6) && NZ(RL17,7) && NZ(RL17,8) && NZ(RL17,9) && NZ(RL17,10) == 0,
              3386$,Yes:
              Else,3387$,Yes;
3386$    ASSIGN:      CheckGapL7850Speed55_75t.NumberOut
True=CheckGapL7850Speed55_75t.NumberOut True + 1:NEXT(1379$);

3387$    ASSIGN:      CheckGapL7850Speed55_75t.NumberOut
False=CheckGapL7850Speed55_75t.NumberOut False + 1:NEXT(1378$);

1379$    TRANSPORT:   LAGVT,RStation7900;

1378$    TRANSPORT:   LAGVT,LStation7900;

;
;
;   Model statements for module: Decide 1551
;
1417$    BRANCH,      1:
          If,

```

```

NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL20,2) &&NZ(RL20,3)
&&NZ(RL20,4)&&NZ(RL20,5)&&NZ(RL20,6)&& NZ(RL17,7) && NZ(RL17,8) && NZ(RL17,9) &&
NZ(RL17,10) == 0,

```

```

3388$,Yes:

```

```

Else,3389$,Yes;

```

```

3388$    ASSIGN:    CheckGapL7850Speed35_55t.NumberOut
True=CheckGapL7850Speed35_55t.NumberOut True + 1:NEXT(1380$);

```

```

3389$    ASSIGN:    CheckGapL7850Speed35_55t.NumberOut
False=CheckGapL7850Speed35_55t.NumberOut False + 1:NEXT(1381$);

```

```

1380$    TRANSPORT:  LAGVT,RStation7900;

```

```

1381$    TRANSPORT:  LAGVT,LStation7900;

```

```

;

```

```

;

```

```

;    Model statements for module: Decide 1552

```

```

;

```

```

1418$    BRANCH,      1:

```

```

    If,

```

```

        NL(RL18)&&NL(RL19)&&NZ(RL20,1) &&NZ(RL20,2) && NZ(RL17,7) &&
NZ(RL17,8) && NZ(RL17,9) && NZ(RL17,10) == 0,

```

```

3390$,Yes:

```

```

Else,3391$,Yes;

```

```

3390$    ASSIGN:    CheckGapL7850Speed10_35t.NumberOut
True=CheckGapL7850Speed10_35t.NumberOut True + 1:NEXT(1382$);

```

```

3391$    ASSIGN:    CheckGapL7850Speed10_35t.NumberOut
False=CheckGapL7850Speed10_35t.NumberOut False + 1:NEXT(1383$);

```

```

1382$    TRANSPORT:  LAGVT,RStation7900;

```

```

1383$    TRANSPORT:  LAGVT,LStation7900;

```

```

;

```

```

;

```

```

;    Model statements for module: Decide 1554

```

```

;

```

```

1420$    BRANCH,      1:

```

```

    If,SpeedatRStation7800>75,1421$,Yes:

```

```

    If,SpeedatRStation7800 > 55 && SpeedatRStation7800<= 75,1422$,Yes:

```

```

    If,SpeedatRStation7800 >35 && SpeedatRStation7800 <=55,1423$,Yes:

```

```

    If,SpeedatRStation7800 >10 && SpeedatRStation7800 <=35,1424$,Yes:

```

```

    Else,1425$,Yes;

```

```

;

```

```

;

```

```

;    Model statements for module: Decide 1559

```

```

;

```

```

1425$    BRANCH,      1:

```

```

                If,NL(RL16)&&NL(RL17)&&NZ(RL18,1)&&NZ(RL18,2)== 0,3394$,Yes:
                Else,3395$,Yes;
3394$    ASSIGN:    CheckGapL7850Speed10lesst.NumberOut
True=CheckGapL7850Speed10lesst.NumberOut True + 1:NEXT(1434$);

3395$    ASSIGN:    CheckGapL7850Speed10lesst.NumberOut
False=CheckGapL7850Speed10lesst.NumberOut False + 1:NEXT(1435$);

1434$    TRANSPORT:  LAGVT,RStation7900;

1435$    TRANSPORT:  LAGVT,LStation7900;

;
;
;   Model statements for module: Decide 1555
;
1421$    BRANCH,    1:
                If,
                NZ(RL13,1)&&NZ(RL13,2) &&NZ(RL14,1)&&NZ(RL14,2)
&&NL(RL15)&&NL(RL16)&&NL(RL17)&&NZ(RL18,1)&&NZ(RL18,2)&&NZ(RL18,3)== 0,
                3396$,Yes:
                Else,3397$,Yes;
3396$    ASSIGN:    CheckGapL7850Speed75lesst.NumberOut
True=CheckGapL7850Speed75lesst.NumberOut True + 1:NEXT(1426$);

3397$    ASSIGN:    CheckGapL7850Speed75lesst.NumberOut
False=CheckGapL7850Speed75lesst.NumberOut False + 1:NEXT(1427$);

1426$    TRANSPORT:  LAGVT,RStation7900;

1427$    TRANSPORT:  LAGVT,LStation7900;

;
;
;   Model statements for module: Decide 1556
;
1422$    BRANCH,    1:

If,NL(RL14)&&NL(RL15)&&NL(RL16)&&NL(RL17)&&NZ(RL18,1)&&NZ(RL18,2)&&NZ(RL18,3)=
= 0,3398$,Yes:
                Else,3399$,Yes;
3398$    ASSIGN:    CheckGapL7850Speed55_75lesst.NumberOut
True=CheckGapL7850Speed55_75lesst.NumberOut True + 1
:NEXT(1428$);

3399$    ASSIGN:    CheckGapL7850Speed55_75lesst.NumberOut
False=CheckGapL7850Speed55_75lesst.NumberOut False + 1
:NEXT(1429$);

1428$    TRANSPORT:  LAGVT,RStation7900;

```

```

1429$    TRANSPORT:  LAGVT,LStation7900;

;
;
;  Model statements for module: Decide 1557
;
1423$    BRANCH,      1:

If,NL(RL15)&&NL(RL16)&&NL(RL17)&&NZ(RL18,1)&&NZ(RL18,2)&&NZ(RL18,3)==
0,3400$,Yes:
        Else,3401$,Yes;
3400$    ASSIGN:      CheckGapL7850Speed35_55lesst.NumberOut
True=CheckGapL7850Speed35_55lesst.NumberOut True + 1
        :NEXT(1430$);

3401$    ASSIGN:      CheckGapL7850Speed35_55lesst.NumberOut
False=CheckGapL7850Speed35_55lesst.NumberOut False + 1
        :NEXT(1431$);

1430$    TRANSPORT:  LAGVT,RStation7900;

1431$    TRANSPORT:  LAGVT,LStation7900;

;
;
;  Model statements for module: Decide 1558
;
1424$    BRANCH,      1:
        If,

NZ(RL15,6)&&NZ(RL15,7)&&NZ(RL15,8)&&NZ(RL15,9)&&NZ(RL15,10)&&NL(RL16)&&NL(RL17
)&&NZ(RL18,1)&&NZ(RL18,2)&&NZ(RL18,3)== 0,
        3402$,Yes:
        Else,3403$,Yes;
3402$    ASSIGN:      CheckGapL7850Speed10_35lesst.NumberOut
True=CheckGapL7850Speed10_35lesst.NumberOut True + 1
        :NEXT(1432$);

3403$    ASSIGN:      CheckGapL7850Speed10_35lesst.NumberOut
False=CheckGapL7850Speed10_35lesst.NumberOut False + 1
        :NEXT(1433$);

1432$    TRANSPORT:  LAGVT,RStation7900;

1433$    TRANSPORT:  LAGVT,LStation7900;

;
;
;  Model statements for module: Station 120
;

```

```

1436$    STATION,    LStation7900;
3406$    DELAY:      0.0,,VA:NEXT(1437$);

;
;
;    Model statements for module: Decide 1560
;
1437$    BRANCH,      1:
                If,Entity.Type==LCar,3407$,Yes:
                Else,3408$,Yes;
3407$    ASSIGN:      checkvehicletype7900.NumberOut True=checkvehicletype7900.NumberOut
True + 1:NEXT(1438$);

3408$    ASSIGN:      checkvehicletype7900.NumberOut False=checkvehicletype7900.NumberOut
False + 1:NEXT(1439$);

;
;
;    Model statements for module: Decide 1561
;
1438$    BRANCH,      1:
                If,SpeedatRStation7850 >= VTU(LAGV,LCar#),3409$,Yes:
                Else,3410$,Yes;
3409$    ASSIGN:      Decide 1561.NumberOut True=Decide 1561.NumberOut True +
1:NEXT(1440$);

3410$    ASSIGN:      Decide 1561.NumberOut False=Decide 1561.NumberOut False +
1:NEXT(1466$);

;
;
;    Model statements for module: Decide 1563
;
1440$    BRANCH,      1:
                If,SpeedatRStation7850>75,1441$,Yes:
                If,SpeedatRStation7850 > 55 && SpeedatRStation7850<= 75,1454$,Yes:
                If,SpeedatRStation7850 >35 && SpeedatRStation7850<=55,1457$,Yes:
                If,SpeedatRStation7850>10 && SpeedatRStation7850 <=35,1460$,Yes:
                Else,1463$,Yes;

;
;
;    Model statements for module: Decide 1568
;
1463$    BRANCH,      1:
                If,NL(RL19)&&NZ(RL18,9) && NZ(RL18,10) == 0,3413$,Yes:
                Else,3414$,Yes;
3413$    ASSIGN:      CheckGapL7900Speed10.NumberOut
True=CheckGapL7900Speed10.NumberOut True + 1:NEXT(1464$);

```

```

3414$    ASSIGN:    CheckGapL7900Speed10.NumberOut
False=CheckGapL7900Speed10.NumberOut False + 1:NEXT(1465$);

1464$    TRANSPORT:  LAGV,RStation7950;

1465$    TRANSPORT:  LAGV,LStation7950;

;
;
;   Model statements for module: Decide 1564
;
1441$    BRANCH,    1:
                If,NL(RL19)&&NL(RL20)&&NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) ==
0,3415$,Yes:
                Else,3416$,Yes;
3415$    ASSIGN:    CheckGapL7900Speed75.NumberOut
True=CheckGapL7900Speed75.NumberOut True + 1:NEXT(1442$);

3416$    ASSIGN:    CheckGapL7900Speed75.NumberOut
False=CheckGapL7900Speed75.NumberOut False + 1:NEXT(1443$);

1442$    TRANSPORT:  LAGV,RStation7950;

1443$    TRANSPORT:  LAGV,LStation7950;

;
;
;   Model statements for module: Decide 1565
;
1454$    BRANCH,    1:
                If,
                NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2) &&NZ(RL20,3) &&NZ(RL20,4)
&&NZ(RL20,5) &&NZ(RL20,6) &&NZ(RL20,7) &&NZ(RL20,8) &&NZ(RL20,9) &&NZ(RL20,10)
&&NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) == 0,
                3417$,Yes:
                Else,3418$,Yes;
3417$    ASSIGN:    CheckGapL7900Speed55_75.NumberOut
True=CheckGapL7900Speed55_75.NumberOut True + 1:NEXT(1455$);

3418$    ASSIGN:    CheckGapL7900Speed55_75.NumberOut
False=CheckGapL7900Speed55_75.NumberOut False + 1:NEXT(1456$);

1455$    TRANSPORT:  LAGV,RStation7950;

1456$    TRANSPORT:  LAGV,LStation7950;

;
;
;   Model statements for module: Decide 1566

```

```

;
1457$    BRANCH,    1:
            If,
                NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2) &&NZ(RL20,3) &&NZ(RL20,4)
&&NZ(RL20,5) &&NZ(RL20,6) &&NZ(RL20,7) &&NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10)
== 0,
                3419$,Yes:
                Else,3420$,Yes;
3419$    ASSIGN:    CheckGapL7900Speed35_55.NumberOut
True=CheckGapL7900Speed35_55.NumberOut True + 1:NEXT(1458$);

3420$    ASSIGN:    CheckGapL7900Speed35_55.NumberOut
False=CheckGapL7900Speed35_55.NumberOut False + 1:NEXT(1459$);

1458$    TRANSPORT:  LAGV,RStation7950;

1459$    TRANSPORT:  LAGV,LStation7950;

;
;
;    Model statements for module: Decide 1567
;
1460$    BRANCH,    1:
            If,NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2) &&NZ(RL20,3)&&NZ(RL18,9) &&
NZ(RL18,10) == 0,3421$,Yes:
            Else,3422$,Yes;
3421$    ASSIGN:    CheckGapL7900Speed10_35.NumberOut
True=CheckGapL7900Speed10_35.NumberOut True + 1:NEXT(1461$);

3422$    ASSIGN:    CheckGapL7900Speed10_35.NumberOut
False=CheckGapL7900Speed10_35.NumberOut False + 1:NEXT(1462$);

1461$    TRANSPORT:  LAGV,RStation7950;

1462$    TRANSPORT:  LAGV,LStation7950;

;
;
;    Model statements for module: Decide 1569
;
1466$    BRANCH,    1:
            If,SpeedatRStation7850>75,1467$,Yes:
            If,SpeedatRStation7850 > 55 && SpeedatRStation7850 <= 75,1470$,Yes:
            If,SpeedatRStation7850>35 && SpeedatRStation7850 <=55,1473$,Yes:
            If,SpeedatRStation7850>10 && SpeedatRStation7850 <=35,1476$,Yes:
            Else,1479$,Yes;

;
;
;    Model statements for module: Decide 1574
;

```



```

1479$    BRANCH,    1:
          If,NL(RL18)&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)== 0,3425$,Yes:
          Else,3426$,Yes;
3425$    ASSIGN:    CheckGapL7900Speed10less.NumberOut
True=CheckGapL7900Speed10less.NumberOut True + 1:NEXT(1480$);

3426$    ASSIGN:    CheckGapL7900Speed10less.NumberOut
False=CheckGapL7900Speed10less.NumberOut False + 1:NEXT(1481$);

1480$    TRANSPORT:  LAGV,RStation7950;

1481$    TRANSPORT:  LAGV,LStation7950;

;
;
;   Model statements for module: Decide 1570
;
1467$    BRANCH,    1:

If,NL(RL15)&&NL(RL16)&&NL(RL17)&&NL(RL18)&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)=
= 0,3427$,Yes:
          Else,3428$,Yes;
3427$    ASSIGN:    CheckGapL7900Speed75less.NumberOut
True=CheckGapL7900Speed75less.NumberOut True + 1:NEXT(1468$);

3428$    ASSIGN:    CheckGapL7900Speed75less.NumberOut
False=CheckGapL7900Speed75less.NumberOut False + 1:NEXT(1469$);

1468$    TRANSPORT:  LAGV,RStation7950;

1469$    TRANSPORT:  LAGV,LStation7950;

;
;
;   Model statements for module: Decide 1571
;
1470$    BRANCH,    1:

If,NL(RL16)&&NL(RL17)&&NL(RL18)&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)==
0,3429$,Yes:
          Else,3430$,Yes;
3429$    ASSIGN:    CheckGapL7900Speed55_75less.NumberOut
True=CheckGapL7900Speed55_75less.NumberOut True + 1
          :NEXT(1471$);

3430$    ASSIGN:    CheckGapL7900Speed55_75less.NumberOut
False=CheckGapL7900Speed55_75less.NumberOut False + 1
          :NEXT(1472$);

1471$    TRANSPORT:  LAGV,RStation7950;

```

1472\$ TRANSPORT: LAGV,LStation7950;

;
;
;
;
;

Model statements for module: Decide 1572

1473\$ BRANCH, 1:
 If,

NZ(RL16,6)&&NZ(RL16,7)&&NZ(RL16,8)&&NZ(RL16,9)&&NZ(RL16,10)&&NL(RL17)&&NL(RL18)
&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)== 0,

3431\$,Yes:

Else,3432\$,Yes;

3431\$ ASSIGN: CheckGapL7900Speed35_55less.NumberOut
True=CheckGapL7900Speed35_55less.NumberOut True + 1
:NEXT(1474\$);

3432\$ ASSIGN: CheckGapL7900Speed35_55less.NumberOut
False=CheckGapL7900Speed35_55less.NumberOut False + 1
:NEXT(1475\$);

1474\$ TRANSPORT: LAGV,RStation7950;

1475\$ TRANSPORT: LAGV,LStation7950;

;
;
;
;
;

Model statements for module: Decide 1573

1476\$ BRANCH, 1:
 If,

NZ(RL17,6)&&NZ(RL17,7)&&NZ(RL17,8)&&NZ(RL17,9)&&NZ(RL17,10)&&NL(RL18)&&NZ(RL19
,1)&&NZ(RL19,2)&&NZ(RL19,3)== 0,

3433\$,Yes:

Else,3434\$,Yes;

3433\$ ASSIGN: CheckGapL7900Speed10_35less.NumberOut
True=CheckGapL7900Speed10_35less.NumberOut True + 1
:NEXT(1477\$);

3434\$ ASSIGN: CheckGapL7900Speed10_35less.NumberOut
False=CheckGapL7900Speed10_35less.NumberOut False + 1
:NEXT(1478\$);

1477\$ TRANSPORT: LAGV,RStation7950;

1478\$ TRANSPORT: LAGV,LStation7950;

;
;

```

;   Model statements for module: Decide 1562
;
1439$   BRANCH,      1:
           If,SpeedatRStation7850 >= VTU(LAGVT,LTruck#),3435$,Yes:
           Else,3436$,Yes;
3435$   ASSIGN:      Decide 1562.NumberOut True=Decide 1562.NumberOut True +
1:NEXT(1482$);

3436$   ASSIGN:      Decide 1562.NumberOut False=Decide 1562.NumberOut False +
1:NEXT(1488$);

;
;
;   Model statements for module: Decide 1575
;
1482$   BRANCH,      1:
           If,SpeedatRStation7850>75,1483$,Yes:
           If,SpeedatRStation7850 > 55 && SpeedatRStation7850 <= 75,1484$,Yes:
           If,SpeedatRStation7850 >35 && SpeedatRStation7850 <=55,1485$,Yes:
           If,SpeedatRStation7850 >10 && SpeedatRStation7850 <=35,1486$,Yes:
           Else,1487$,Yes;

;
;
;   Model statements for module: Decide 1580
;
1487$   BRANCH,      1:
           If,
           NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2) &&NZ(RL20,3) &&NZ(RL20,4)
&&NZ(RL20,5)&&NZ(RL18,9) && NZ(RL18,10) == 0,
           3439$,Yes:
           Else,3440$,Yes;
3439$   ASSIGN:      CheckGapL7900Speed10t.NumberOut
True=CheckGapL7900Speed10t.NumberOut True + 1:NEXT(1452$);

3440$   ASSIGN:      CheckGapL7900Speed10t.NumberOut
False=CheckGapL7900Speed10t.NumberOut False + 1:NEXT(1453$);

1452$   TRANSPORT:   LAGVT,RStation7950;

1453$   TRANSPORT:   LAGVT,LStation7950;

;
;
;   Model statements for module: Decide 1576
;
1483$   BRANCH,      1:
           If,NL(RL19)&&NL(RL20)&&NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) ==
0,3441$,Yes:
           Else,3442$,Yes;

```

```

3441$    ASSIGN:    CheckGapL7900Speed75t.NumberOut
True=CheckGapL7900Speed75t.NumberOut True + 1:NEXT(1444$);

3442$    ASSIGN:    CheckGapL7900Speed75t.NumberOut
False=CheckGapL7900Speed75t.NumberOut False + 1:NEXT(1445$);

1444$    TRANSPORT:  LAGVT,RStation7950;

1445$    TRANSPORT:  LAGVT,LStation7950;

;
;
;  Model statements for module: Decide 1577
;
1484$    BRANCH,    1:
                If,NL(RL19)&&NL(RL20)&&NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) ==
0,3443$,Yes:
                Else,3444$,Yes;
3443$    ASSIGN:    CheckGapL7900Speed55_75t.NumberOut
True=CheckGapL7900Speed55_75t.NumberOut True + 1:NEXT(1447$);

3444$    ASSIGN:    CheckGapL7900Speed55_75t.NumberOut
False=CheckGapL7900Speed55_75t.NumberOut False + 1:NEXT(1446$);

1447$    TRANSPORT:  LAGVT,RStation7950;

1446$    TRANSPORT:  LAGVT,LStation7950;

;
;
;  Model statements for module: Decide 1578
;
1485$    BRANCH,    1:
                If,
                NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2) &&NZ(RL20,3) &&NZ(RL20,4)
&&NZ(RL20,5) &&NZ(RL20,6) &&NZ(RL20,7) &&NZ(RL20,8) &&NZ(RL20,9) &&NZ(RL20,10)
&&NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) == 0,
                3445$,Yes:
                Else,3446$,Yes;
3445$    ASSIGN:    CheckGapL7900Speed35_55t.NumberOut
True=CheckGapL7900Speed35_55t.NumberOut True + 1:NEXT(1448$);

3446$    ASSIGN:    CheckGapL7900Speed35_55t.NumberOut
False=CheckGapL7900Speed35_55t.NumberOut False + 1:NEXT(1449$);

1448$    TRANSPORT:  LAGVT,RStation7950;

1449$    TRANSPORT:  LAGVT,LStation7950;

;

```

```

;
;   Model statements for module: Decide 1579
;
1486$   BRANCH,      1:
        If,
            NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2) &&NZ(RL20,3) &&NZ(RL20,4)
&&NZ(RL20,5) &&NZ(RL20,6) &&NZ(RL20,7) &&NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10)
== 0,
            3447$,Yes:
            Else,3448$,Yes;
3447$   ASSIGN:      CheckGapL7900Speed10_35t.NumberOut
True=CheckGapL7900Speed10_35t.NumberOut True + 1:NEXT(1450$);

3448$   ASSIGN:      CheckGapL7900Speed10_35t.NumberOut
False=CheckGapL7900Speed10_35t.NumberOut False + 1:NEXT(1451$);

1450$   TRANSPORT:   LAGVT,RStation7950;

1451$   TRANSPORT:   LAGVT,LStation7950;

;
;
;   Model statements for module: Decide 1581
;
1488$   BRANCH,      1:
        If,SpeedatRStation7850>75,1489$,Yes:
        If,SpeedatRStation7850 > 55 && SpeedatRStation7850<= 75,1490$,Yes:
        If,SpeedatRStation7850 >35 && SpeedatRStation7850 <=55,1491$,Yes:
        If,SpeedatRStation7850 >10 && SpeedatRStation7850 <=35,1492$,Yes:
        Else,1493$,Yes;

;
;
;   Model statements for module: Decide 1586
;
1493$   BRANCH,      1:
        If,NL(RL17)&&NL(RL18)&&NZ(RL19,1)&&NZ(RL19,2)== 0,3451$,Yes:
        Else,3452$,Yes;
3451$   ASSIGN:      CheckGapL7900Speed10lesst.NumberOut
True=CheckGapL7900Speed10lesst.NumberOut True + 1:NEXT(1502$);

3452$   ASSIGN:      CheckGapL7900Speed10lesst.NumberOut
False=CheckGapL7900Speed10lesst.NumberOut False + 1:NEXT(1503$);

1502$   TRANSPORT:   LAGVT,RStation7950;

1503$   TRANSPORT:   LAGVT,LStation7950;

;
;
;   Model statements for module: Decide 1582

```

```

;
1489$    BRANCH,    1:
        If,
        NZ(RL14,1)&&NZ(RL14,2)
&&NL(RL15)&&NL(RL16)&&NL(RL17)&&NL(RL18)&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)
== 0,
        3453$,Yes:
        Else,3454$,Yes;
3453$    ASSIGN:    CheckGapL7900Speed75lesst.NumberOut
True=CheckGapL7900Speed75lesst.NumberOut True + 1:NEXT(1494$);

3454$    ASSIGN:    CheckGapL7900Speed75lesst.NumberOut
False=CheckGapL7900Speed75lesst.NumberOut False + 1:NEXT(1495$);

1494$    TRANSPORT:  LAGVT,RStation7950;

1495$    TRANSPORT:  LAGVT,LStation7950;

;
;
;    Model statements for module: Decide 1583
;
1490$    BRANCH,    1:

If,NL(RL15)&&NL(RL16)&&NL(RL17)&&NL(RL18)&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)=
= 0,3455$,Yes:
        Else,3456$,Yes;
3455$    ASSIGN:    CheckGapL7900Speed55_75lesst.NumberOut
True=CheckGapL7900Speed55_75lesst.NumberOut True + 1
        :NEXT(1496$);

3456$    ASSIGN:    CheckGapL7900Speed55_75lesst.NumberOut
False=CheckGapL7900Speed55_75lesst.NumberOut False + 1
        :NEXT(1497$);

1496$    TRANSPORT:  LAGVT,RStation7950;

1497$    TRANSPORT:  LAGVT,LStation7950;

;
;
;    Model statements for module: Decide 1584
;
1491$    BRANCH,    1:

If,NL(RL16)&&NL(RL17)&&NL(RL18)&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)==
0,3457$,Yes:
        Else,3458$,Yes;
3457$    ASSIGN:    CheckGapL7900Speed35_55lesst.NumberOut
True=CheckGapL7900Speed35_55lesst.NumberOut True + 1
        :NEXT(1498$);

```

```

3458$    ASSIGN:    CheckGapL7900Speed35_55lesst.NumberOut
False=CheckGapL7900Speed35_55lesst.NumberOut False + 1
          :NEXT(1499$);

```

```

1498$    TRANSPORT:  LAGVT,RStation7950;

```

```

1499$    TRANSPORT:  LAGVT,LStation7950;

```

```

;
;
;   Model statements for module: Decide 1585
;

```

```

1492$    BRANCH,      1:
          If,

```

```

NZ(RL16,6)&&NZ(RL16,7)&&NZ(RL16,8)&&NZ(RL16,9)&&NZ(RL16,10)&&NL(RL17)&&NL(RL18)
)&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)== 0,
          3459$,Yes:
          Else,3460$,Yes;

```

```

3459$    ASSIGN:    CheckGapL7900Speed10_35lesst.NumberOut
True=CheckGapL7900Speed10_35lesst.NumberOut True + 1
          :NEXT(1500$);

```

```

3460$    ASSIGN:    CheckGapL7900Speed10_35lesst.NumberOut
False=CheckGapL7900Speed10_35lesst.NumberOut False + 1
          :NEXT(1501$);

```

```

1500$    TRANSPORT:  LAGVT,RStation7950;

```

```

1501$    TRANSPORT:  LAGVT,LStation7950;

```

```

;
;
;   Model statements for module: Station 121
;

```

```

1504$    STATION,      LStation7950;
3463$    DELAY:        0.0,,VA:NEXT(1505$);

```

```

;
;
;   Model statements for module: Decide 1587
;

```

```

1505$    BRANCH,      1:
          If,Entity.Type==LCar,3464$,Yes:
          Else,3465$,Yes;

```

```

3464$    ASSIGN:    checkvehicletype7950.NumberOut True=checkvehicletype7950.NumberOut
True + 1:NEXT(1512$);

```

```

3465$    ASSIGN:    checkvehicletype7950.NumberOut False=checkvehicletype7950.NumberOut
False + 1:NEXT(1513$);

```

```

;
;
;    Model statements for module: Decide 1615
;
1512$    BRANCH,    1:
            If,NQ(QueueatTaperT) + NQ(QueueatTaperC)<= 3,1506$,Yes:
            If,NQ(QueueatTaperT) + NQ(QueueatTaperC) > 3 && NQ(QueueatTaperT) +
NQ(QueueatTaperC) <= 5,1519$,
            Yes:
            If,NQ(QueueatTaperT) + NQ(QueueatTaperC) > 5 && NQ(QueueatTaperT) +
NQ(QueueatTaperC) <= 10,1530$,
            Yes:
            Else,1541$,Yes;

```

```

;
;
;    Model statements for module: Decide 1641
;
1541$    BRANCH,    1:
            If,SpeedatRStation7900>75,1547$,Yes:
            If,SpeedatRStation7900 > 55 && SpeedatRStation7900<= 75,1548$,Yes:
            If,SpeedatRStation7900 >35 && SpeedatRStation7900<=55,1549$,Yes:
            If,SpeedatRStation7900>10 && SpeedatRStation7900 <=35,1550$,Yes:
            If,SpeedatRStation7900 <=10,1551$,Yes:
            Else,1600$,Yes;

```

```

;
;
;    Model statements for module: Hold 47
;
1600$    QUEUE,    QueueatTaperC;
SCAN:    NZ(RL20,1)== 0:NEXT(1598$);

```

```

1598$    TRANSPORT:    LAGV,RStation8100;

```

```

;
;
;    Model statements for module: Decide 1642
;
1547$    BRANCH,    1:
            If,NL(RL18) && NL(RL19)&&NZ(RL20,1)== 0,3470$,Yes:
            Else,3471$,Yes;
3470$    ASSIGN:    Decide 1642.NumberOut True=Decide 1642.NumberOut True +
1:NEXT(1542$);

3471$    ASSIGN:    Decide 1642.NumberOut False=Decide 1642.NumberOut False +
1:NEXT(1600$);

```


1542\$ TRANSPORT: LAGV,RStation8100;

;
;
;
;
;

Model statements for module: Decide 1643

1548\$ BRANCH, 1:
 If,NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) && NL(RL19)&&NZ(RL20,1)==
0,3472\$,Yes:

 Else,3473\$,Yes;

3472\$ ASSIGN: Decide 1643.NumberOut True=Decide 1643.NumberOut True +
1:NEXT(1543\$);

3473\$ ASSIGN: Decide 1643.NumberOut False=Decide 1643.NumberOut False +
1:NEXT(1600\$);

1543\$ TRANSPORT: LAGV,RStation8100;

;
;
;
;
;

Model statements for module: Decide 1644

1549\$ BRANCH, 1:
 If,NZ(RL18,10) && NL(RL19)&&NZ(RL20,1)== 0,3474\$,Yes:
 Else,3475\$,Yes;

3474\$ ASSIGN: Decide 1644.NumberOut True=Decide 1644.NumberOut True +
1:NEXT(1544\$);

3475\$ ASSIGN: Decide 1644.NumberOut False=Decide 1644.NumberOut False +
1:NEXT(1600\$);

1544\$ TRANSPORT: LAGV,RStation8100;

;
;
;
;
;

Model statements for module: Decide 1645

1550\$ BRANCH, 1:
 If,

NZ(RL19,4)&&NZ(RL19,5)&&NZ(RL19,6)&&NZ(RL19,7)&&NZ(RL19,8)&&NZ(RL19,9)&&NZ(RL1
9,10)&&NZ(RL20,1)== 0,

 3476\$,Yes:

 Else,3477\$,Yes;

3476\$ ASSIGN: Decide 1645.NumberOut True=Decide 1645.NumberOut True +
1:NEXT(1545\$);

3477\$ ASSIGN: Decide 1645.NumberOut False=Decide 1645.NumberOut False +
1:NEXT(1600\$);

```

1545$    TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1646
;
1551$    BRANCH,      1:
                If,NZ(RL19,7) &&NZ(RL19,8) &&NZ(RL19,9) &&NZ(RL19,10) &&NZ(RL20,1)==
0,3478$,Yes:
                Else,3479$,Yes;
3478$    ASSIGN:      Decide 1646.NumberOut True=Decide 1646.NumberOut True +
1:NEXT(1546$);

3479$    ASSIGN:      Decide 1646.NumberOut False=Decide 1646.NumberOut False +
1:NEXT(1600$);

1546$    TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1590
;
1506$    BRANCH,      1:
                If,SpeedatRStation7900>75,1514$,Yes:
                If,SpeedatRStation7900 > 55 && SpeedatRStation7900<= 75,1515$,Yes:
                If,SpeedatRStation7900 >35 && SpeedatRStation7900<=55,1516$,Yes:
                If,SpeedatRStation7900>10 && SpeedatRStation7900 <=35,1517$,Yes:
                If,SpeedatRStation7900 <=10,1518$,Yes:
                Else,1600$,Yes;

;
;
;   Model statements for module: Decide 1624
;
1514$    BRANCH,      1:
                If,
                NZ(RL17,6) &&NZ(RL17,7) &&NZ(RL17,8) &&NZ(RL17,9) &&NZ(RL17,10) &&
NL(RL18) && NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,
                3482$,Yes:
                Else,3483$,Yes;
3482$    ASSIGN:      Decide 1624.NumberOut True=Decide 1624.NumberOut True +
1:NEXT(1507$);

3483$    ASSIGN:      Decide 1624.NumberOut False=Decide 1624.NumberOut False +
1:NEXT(1600$);

1507$    TRANSPORT:   LAGV,RStation8100;

;
;

```

```

;   Model statements for module: Decide 1625
;
1515$   BRANCH,      1:
        If,
        NZ(RL18,6) &&NZ(RL18,7) &&NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) &&
NL(RL19) &&NZ(RL20,1)&&NZ(RL20,2)== 0,
        3484$,Yes:
        Else,3485$,Yes;
3484$   ASSIGN:      Decide 1625.NumberOut True=Decide 1625.NumberOut True +
1:NEXT(1508$);

3485$   ASSIGN:      Decide 1625.NumberOut False=Decide 1625.NumberOut False +
1:NEXT(1600$);

1508$   TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1626
;
1516$   BRANCH,      1:
        If,NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) && NL(RL19)
&&NZ(RL20,1)&&NZ(RL20,2)== 0,3486$,Yes:
        Else,3487$,Yes;
3486$   ASSIGN:      Decide 1626.NumberOut True=Decide 1626.NumberOut True +
1:NEXT(1509$);

3487$   ASSIGN:      Decide 1626.NumberOut False=Decide 1626.NumberOut False +
1:NEXT(1600$);

1509$   TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1627
;
1517$   BRANCH,      1:
        If,NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) && NL(RL19) &&NZ(RL20,1)==
0,3488$,Yes:
        Else,3489$,Yes;
3488$   ASSIGN:      Decide 1627.NumberOut True=Decide 1627.NumberOut True +
1:NEXT(1510$);

3489$   ASSIGN:      Decide 1627.NumberOut False=Decide 1627.NumberOut False +
1:NEXT(1600$);

1510$   TRANSPORT:   LAGV,RStation8100;

;
;

```

```

;   Model statements for module: Decide 1628
;
1518$   BRANCH,      1:
          If,NL(RL19) &&NZ(RL20,1)== 0,3490$,Yes:
          Else,3491$,Yes;
3490$   ASSIGN:      Decide 1628.NumberOut True=Decide 1628.NumberOut True +
1:NEXT(1511$);

3491$   ASSIGN:      Decide 1628.NumberOut False=Decide 1628.NumberOut False +
1:NEXT(1600$);

1511$   TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1629
;
1519$   BRANCH,      1:
          If,SpeedatRStation7900>75,1525$,Yes:
          If,SpeedatRStation7900 > 55 && SpeedatRStation7900<= 75,1526$,Yes:
          If,SpeedatRStation7900 >35 && SpeedatRStation7900<=55,1527$,Yes:
          If,SpeedatRStation7900>10 && SpeedatRStation7900 <=35,1528$,Yes:
          If,SpeedatRStation7900 <=10,1529$,Yes:
          Else,1600$,Yes;

;
;
;   Model statements for module: Decide 1630
;
1525$   BRANCH,      1:
          If,NL(RL18) && NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)&&NZ(RL20,3)==
0,3494$,Yes:
          Else,3495$,Yes;
3494$   ASSIGN:      Decide 1630.NumberOut True=Decide 1630.NumberOut True +
1:NEXT(1520$);

3495$   ASSIGN:      Decide 1630.NumberOut False=Decide 1630.NumberOut False +
1:NEXT(1600$);

1520$   TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1631
;
1526$   BRANCH,      1:
          If,
          NZ(RL18,6) &&NZ(RL18,7) &&NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) &&
NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,
          3496$,Yes:
          Else,3497$,Yes;

```

3496\$ ASSIGN: Decide 1631.NumberOut True=Decide 1631.NumberOut True +
1:NEXT(1521\$);

3497\$ ASSIGN: Decide 1631.NumberOut False=Decide 1631.NumberOut False +
1:NEXT(1600\$);

1521\$ TRANSPORT: LAGV,RStation8100;

;
;
; Model statements for module: Decide 1632
;

1527\$ BRANCH, 1:
 If,NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) &&
NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,3498\$,Yes:
 Else,3499\$,Yes;

3498\$ ASSIGN: Decide 1632.NumberOut True=Decide 1632.NumberOut True +
1:NEXT(1522\$);

3499\$ ASSIGN: Decide 1632.NumberOut False=Decide 1632.NumberOut False +
1:NEXT(1600\$);

1522\$ TRANSPORT: LAGV,RStation8100;

;
;
; Model statements for module: Decide 1633
;

1528\$ BRANCH, 1:
 If,NZ(RL18,5) && NL(RL19)&&NZ(RL20,1)== 0,3500\$,Yes:
 Else,3501\$,Yes;

3500\$ ASSIGN: Decide 1633.NumberOut True=Decide 1633.NumberOut True +
1:NEXT(1523\$);

3501\$ ASSIGN: Decide 1633.NumberOut False=Decide 1633.NumberOut False +
1:NEXT(1600\$);

1523\$ TRANSPORT: LAGV,RStation8100;

;
;
; Model statements for module: Decide 1634
;

1529\$ BRANCH, 1:
 If,
 NZ(RL19,4) &&NZ(RL19,5) &&NZ(RL19,6) &&NZ(RL19,7) &&NZ(RL19,8)
&&NZ(RL19,9) &&NZ(RL19,10) &&NZ(RL20,1)== 0,
 3502\$,Yes:
 Else,3503\$,Yes;

3502\$ ASSIGN: Decide 1634.NumberOut True=Decide 1634.NumberOut True +
1:NEXT(1524\$);

3503\$ ASSIGN: Decide 1634.NumberOut False=Decide 1634.NumberOut False +
1:NEXT(1600\$);

1524\$ TRANSPORT: LAGV,RStation8100;

;
;
; Model statements for module: Decide 1635
;

1530\$ BRANCH, 1:
 If,SpeedatRStation7900>75,1536\$,Yes:
 If,SpeedatRStation7900 > 55 && SpeedatRStation7900<= 75,1537\$,Yes:
 If,SpeedatRStation7900 >35 && SpeedatRStation7900<=55,1538\$,Yes:
 If,SpeedatRStation7900>10 && SpeedatRStation7900 <=35,1539\$,Yes:
 If,SpeedatRStation7900 <=10,1540\$,Yes:
 Else,1600\$,Yes;

;
;
; Model statements for module: Decide 1636
;

1536\$ BRANCH, 1:
 If,NL(RL18) && NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,3506\$,Yes:
 Else,3507\$,Yes;

3506\$ ASSIGN: Decide 1636.NumberOut True=Decide 1636.NumberOut True +
1:NEXT(1531\$);

3507\$ ASSIGN: Decide 1636.NumberOut False=Decide 1636.NumberOut False +
1:NEXT(1600\$);

1531\$ TRANSPORT: LAGV,RStation8100;

;
;
; Model statements for module: Decide 1637
;

1537\$ BRANCH, 1:
 If,NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) &&
NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,3508\$,Yes:
 Else,3509\$,Yes;

3508\$ ASSIGN: Decide 1637.NumberOut True=Decide 1637.NumberOut True +
1:NEXT(1532\$);

3509\$ ASSIGN: Decide 1637.NumberOut False=Decide 1637.NumberOut False +
1:NEXT(1600\$);

1532\$ TRANSPORT: LAGV,RStation8100;

```

;
;
;   Model statements for module: Decide 1638
;
1538$   BRANCH,      1:
          If,NZ(RL18,10) && NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,3510$,Yes:
          Else,3511$,Yes;
3510$   ASSIGN:      Decide 1638.NumberOut True=Decide 1638.NumberOut True +
1:NEXT(1533$);

3511$   ASSIGN:      Decide 1638.NumberOut False=Decide 1638.NumberOut False +
1:NEXT(1600$);

1533$   TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1639
;
1539$   BRANCH,      1:
          If,NL(RL19)&&NZ(RL20,1)== 0,3512$,Yes:
          Else,3513$,Yes;
3512$   ASSIGN:      Decide 1639.NumberOut True=Decide 1639.NumberOut True +
1:NEXT(1534$);

3513$   ASSIGN:      Decide 1639.NumberOut False=Decide 1639.NumberOut False +
1:NEXT(1600$);

1534$   TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1640
;
1540$   BRANCH,      1:
          If,NZ(RL19,6) &&NZ(RL19,7) &&NZ(RL19,8) &&NZ(RL19,9) &&NZ(RL19,10)
&&NZ(RL20,1)== 0,3514$,Yes:
          Else,3515$,Yes;
3514$   ASSIGN:      Decide 1640.NumberOut True=Decide 1640.NumberOut True +
1:NEXT(1535$);

3515$   ASSIGN:      Decide 1640.NumberOut False=Decide 1640.NumberOut False +
1:NEXT(1600$);

1535$   TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Decide 1620

```

```

;
1513$    BRANCH,      1:
          If,NQ(QueueatTaperT) + NQ(QueueatTaperC) <= 3,1552$,Yes:
          If,NQ(QueueatTaperT) + NQ(QueueatTaperC) > 3 && NQ(QueueatTaperT) +
NQ(QueueatTaperC) <= 5,1574$,
          Yes:
          If,NQ(QueueatTaperT) + NQ(QueueatTaperC) > 5 && NQ(QueueatTaperT) +
NQ(QueueatTaperC) <= 10,1585$,
          Yes:
          Else,1563$,Yes;

;
;
; Model statements for module: Decide 1653
;
1563$    BRANCH,      1:
          If,SpeedatRStation7900>75,1569$,Yes:
          If,SpeedatRStation7900 > 55 && SpeedatRStation7900<= 75,1570$,Yes:
          If,SpeedatRStation7900 >35 && SpeedatRStation7900<=55,1571$,Yes:
          If,SpeedatRStation7900>10 && SpeedatRStation7900 <=35,1572$,Yes:
          If,SpeedatRStation7900 <=10,1573$,Yes:
          Else,1596$,Yes;

;
;
; Model statements for module: Hold 46
;
1596$    QUEUE,      QueueatTaperT;
          SCAN:      NZ(RL20,1)== 0:NEXT(1599$);

1599$    TRANSPORT:   LAGVT,RStation8100;

;
;
; Model statements for module: Decide 1654
;
1569$    BRANCH,      1:
          If,NL(RL17) &&NL(RL18) && NL(RL19)&&NZ(RL20,1)== 0,3520$,Yes:
          Else,3521$,Yes;
3520$    ASSIGN:      Decide 1654.NumberOut True=Decide 1654.NumberOut True +
1:NEXT(1564$);

3521$    ASSIGN:      Decide 1654.NumberOut False=Decide 1654.NumberOut False +
1:NEXT(1596$);

1564$    TRANSPORT:   LAGVT,RStation8100;

;
;
; Model statements for module: Decide 1655
;

```



```

1570$    BRANCH,    1:
           If,NZ(RL17,9) &&NZ(RL17,10) && NL(RL18)&& NL(RL19)&&NZ(RL20,1)==
0,3522$,Yes:

```

```

           Else,3523$,Yes;

```

```

3522$    ASSIGN:    Decide 1655.NumberOut True=Decide 1655.NumberOut True +
1:NEXT(1565$);

```

```

3523$    ASSIGN:    Decide 1655.NumberOut False=Decide 1655.NumberOut False +
1:NEXT(1596$);

```

```

1565$    TRANSPORT:  LAGVT,RStation8100;

```

```

;
;
;  Model statements for module: Decide 1656
;

```

```

1571$    BRANCH,    1:
           If,NL(RL18) && NL(RL19)&&NZ(RL20,1)== 0,3524$,Yes:
           Else,3525$,Yes;

```

```

3524$    ASSIGN:    Decide 1656.NumberOut True=Decide 1656.NumberOut True +
1:NEXT(1566$);

```

```

3525$    ASSIGN:    Decide 1656.NumberOut False=Decide 1656.NumberOut False +
1:NEXT(1596$);

```

```

1566$    TRANSPORT:  LAGVT,RStation8100;

```

```

;
;
;  Model statements for module: Decide 1657
;

```

```

1572$    BRANCH,    1:
           If,

```

```

NZ(RL18,5)&&NZ(RL18,6)&&NZ(RL18,7)&&NZ(RL18,8)&&NZ(RL18,9)&&NZ(RL18,10)&&NL(RL
19)&&NZ(RL20,1)== 0,

```

```

           3526$,Yes:

```

```

           Else,3527$,Yes;

```

```

3526$    ASSIGN:    Decide 1657.NumberOut True=Decide 1657.NumberOut True +
1:NEXT(1567$);

```

```

3527$    ASSIGN:    Decide 1657.NumberOut False=Decide 1657.NumberOut False +
1:NEXT(1596$);

```

```

1567$    TRANSPORT:  LAGVT,RStation8100;

```

```

;
;
;  Model statements for module: Decide 1658
;

```

```

1573$    BRANCH,    1:
                If,NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) &&NL(RL19)&&NZ(RL20,1)==
0,3528$,Yes:
                Else,3529$,Yes;
3528$    ASSIGN:    Decide 1658.NumberOut True=Decide 1658.NumberOut True +
1:NEXT(1568$);

3529$    ASSIGN:    Decide 1658.NumberOut False=Decide 1658.NumberOut False +
1:NEXT(1596$);

1568$    TRANSPORT:  LAGVT,RStation8100;

;
;
;   Model statements for module: Decide 1647
;
1552$    BRANCH,    1:
                If,SpeedatRStation7900>75,1558$,Yes:
                If,SpeedatRStation7900 > 55 && SpeedatRStation7900<= 75,1559$,Yes:
                If,SpeedatRStation7900 >35 && SpeedatRStation7900<=55,1560$,Yes:
                If,SpeedatRStation7900>10 && SpeedatRStation7900 <=35,1561$,Yes:
                If,SpeedatRStation7900 <=10,1562$,Yes:
                Else,1596$,Yes;

;
;
;   Model statements for module: Decide 1648
;
1558$    BRANCH,    1:
                If,
                NZ(RL16,7) &&NZ(RL16,8) &&NZ(RL16,9) &&NZ(RL16,10)&& NL(RL17) &&
NL(RL18) && NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,
                3532$,Yes:
                Else,3533$,Yes;
3532$    ASSIGN:    Decide 1648.NumberOut True=Decide 1648.NumberOut True +
1:NEXT(1553$);

3533$    ASSIGN:    Decide 1648.NumberOut False=Decide 1648.NumberOut False +
1:NEXT(1596$);

1553$    TRANSPORT:  LAGVT,RStation8100;

;
;
;   Model statements for module: Decide 1649
;
1559$    BRANCH,    1:
                If,
                NZ(RL17,7) &&NZ(RL17,8) &&NZ(RL17,9) &&NZ(RL17,10) && NL(RL18)&&
NL(RL19) &&NZ(RL20,1)&&NZ(RL20,2)== 0,
                3534$,Yes:

```

```

                Else,3535$,Yes;
3534$    ASSIGN:    Decide 1649.NumberOut True=Decide 1649.NumberOut True +
1:NEXT(1554$);

3535$    ASSIGN:    Decide 1649.NumberOut False=Decide 1649.NumberOut False +
1:NEXT(1596$);

1554$    TRANSPORT:  LAGVT,RStation8100;

;
;
;    Model statements for module: Decide 1650
;
1560$    BRANCH,    1:
                If,NZ(RL17,9) &&NZ(RL17,10) && NL(RL18)&& NL(RL19)
&&NZ(RL20,1)&&NZ(RL20,2)== 0,3536$,Yes:
                Else,3537$,Yes;
3536$    ASSIGN:    Decide 1650.NumberOut True=Decide 1650.NumberOut True +
1:NEXT(1555$);

3537$    ASSIGN:    Decide 1650.NumberOut False=Decide 1650.NumberOut False +
1:NEXT(1596$);

1555$    TRANSPORT:  LAGVT,RStation8100;

;
;
;    Model statements for module: Decide 1651
;
1561$    BRANCH,    1:
                If,NZ(RL17,9) &&NZ(RL17,10) && NL(RL18)&& NL(RL19) &&NZ(RL20,1)==
0,3538$,Yes:
                Else,3539$,Yes;
3538$    ASSIGN:    Decide 1651.NumberOut True=Decide 1651.NumberOut True +
1:NEXT(1556$);

3539$    ASSIGN:    Decide 1651.NumberOut False=Decide 1651.NumberOut False +
1:NEXT(1596$);

1556$    TRANSPORT:  LAGVT,RStation8100;

;
;
;    Model statements for module: Decide 1652
;
1562$    BRANCH,    1:
                If,NL(RL18) &&NL(RL19) &&NZ(RL20,1)== 0,3540$,Yes:
                Else,3541$,Yes;
3540$    ASSIGN:    Decide 1652.NumberOut True=Decide 1652.NumberOut True +
1:NEXT(1557$);

```

3541\$ ASSIGN: Decide 1652.NumberOut False=Decide 1652.NumberOut False +
1:NEXT(1596\$);

1557\$ TRANSPORT: LAGVT,RStation8100;

;
;
; Model statements for module: Decide 1659
;

1574\$ BRANCH, 1:
 If,SpeedatRStation7900>75,1580\$,Yes:
 If,SpeedatRStation7900 > 55 && SpeedatRStation7900<= 75,1581\$,Yes:
 If,SpeedatRStation7900 >35 && SpeedatRStation7900<=55,1582\$,Yes:
 If,SpeedatRStation7900>10 && SpeedatRStation7900 <=35,1583\$,Yes:
 If,SpeedatRStation7900 <=10,1584\$,Yes:
 Else,1596\$,Yes;

;
;
; Model statements for module: Decide 1660
;

1580\$ BRANCH, 1:
 If,NL(RL17) &&NL(RL18) &&
NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)&&NZ(RL20,3)== 0,3544\$,Yes:
 Else,3545\$,Yes;

3544\$ ASSIGN: Decide 1660.NumberOut True=Decide 1660.NumberOut True +
1:NEXT(1575\$);

3545\$ ASSIGN: Decide 1660.NumberOut False=Decide 1660.NumberOut False +
1:NEXT(1596\$);

1575\$ TRANSPORT: LAGVT,RStation8100;

;
;
; Model statements for module: Decide 1661
;

1581\$ BRANCH, 1:
 If,
 NZ(RL17,7) &&NZ(RL17,8) &&NZ(RL17,9) &&NZ(RL17,10) &&NL(RL18)&&
NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,
 3546\$,Yes:
 Else,3547\$,Yes;

3546\$ ASSIGN: Decide 1661.NumberOut True=Decide 1661.NumberOut True +
1:NEXT(1576\$);

3547\$ ASSIGN: Decide 1661.NumberOut False=Decide 1661.NumberOut False +
1:NEXT(1596\$);

1576\$ TRANSPORT: LA GVT,RStation8100;

```

;
;
;   Model statements for module: Decide 1662
;
1582$   BRANCH,      1:
           If,NZ(RL17,9) &&NZ(RL17,10) &&NL(RL18)&&
NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,3548$,Yes:
           Else,3549$,Yes;
3548$   ASSIGN:      Decide 1662.NumberOut True=Decide 1662.NumberOut True +
1:NEXT(1577$);

3549$   ASSIGN:      Decide 1662.NumberOut False=Decide 1662.NumberOut False +
1:NEXT(1596$);

1577$   TRANSPORT:   LAGVT,RStation8100;

;
;
;   Model statements for module: Decide 1663
;
1583$   BRANCH,      1:
           If,NL(RL18) && NL(RL19)&&NZ(RL20,1)== 0,3550$,Yes:
           Else,3551$,Yes;
3550$   ASSIGN:      Decide 1663.NumberOut True=Decide 1663.NumberOut True +
1:NEXT(1578$);

3551$   ASSIGN:      Decide 1663.NumberOut False=Decide 1663.NumberOut False +
1:NEXT(1596$);

1578$   TRANSPORT:   LAGVT,RStation8100;

;
;
;   Model statements for module: Decide 1664
;
1584$   BRANCH,      1:
           If,
           NZ(RL18,5) &&NZ(RL18,6) &&NZ(RL18,7) &&NZ(RL18,8) &&NZ(RL18,9)
&&NZ(RL18,10) &&NL(RL19)&&NZ(RL20,1)== 0,
           3552$,Yes:
           Else,3553$,Yes;
3552$   ASSIGN:      Decide 1664.NumberOut True=Decide 1664.NumberOut True +
1:NEXT(1579$);

3553$   ASSIGN:      Decide 1664.NumberOut False=Decide 1664.NumberOut False +
1:NEXT(1596$);

1579$   TRANSPORT:   LAGVT,RStation8100;

```

```

;
;
; Model statements for module: Decide 1665
;
1585$    BRANCH,    1:
          If,SpeedatRStation7900>75,1591$,Yes:
          If,SpeedatRStation7900 > 55 && SpeedatRStation7900<= 75,1592$,Yes:
          If,SpeedatRStation7900 >35 && SpeedatRStation7900<=55,1593$,Yes:
          If,SpeedatRStation7900>10 && SpeedatRStation7900 <=35,1594$,Yes:
          If,SpeedatRStation7900 <=10,1595$,Yes:
          Else,1596$,Yes;

;
;
; Model statements for module: Decide 1666
;
1591$    BRANCH,    1:
          If,NL(RL17) &&NL(RL18) && NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)==
0,3556$,Yes:
          Else,3557$,Yes;
3556$    ASSIGN:    Decide 1666.NumberOut True=Decide 1666.NumberOut True +
1:NEXT(1586$);
3557$    ASSIGN:    Decide 1666.NumberOut False=Decide 1666.NumberOut False +
1:NEXT(1596$);

1586$    TRANSPORT:  LAGVT,RStation8100;

;
;
; Model statements for module: Decide 1667
;
1592$    BRANCH,    1:
          If,NZ(RL17,9) &&NZ(RL17,10) && NL(RL18)&&
NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,3558$,Yes:
          Else,3559$,Yes;
3558$    ASSIGN:    Decide 1667.NumberOut True=Decide 1667.NumberOut True +
1:NEXT(1587$);

3559$    ASSIGN:    Decide 1667.NumberOut False=Decide 1667.NumberOut False +
1:NEXT(1596$);

1587$    TRANSPORT:  LAGVT,RStation8100;

;
;
; Model statements for module: Decide 1668
;
1593$    BRANCH,    1:
          If,NL(RL18) && NL(RL19)&&NZ(RL20,1)&&NZ(RL20,2)== 0,3560$,Yes:

```

```

                Else,3561$,Yes;
3560$    ASSIGN:    Decide 1668.NumberOut True=Decide 1668.NumberOut True +
1:NEXT(1588$);

3561$    ASSIGN:    Decide 1668.NumberOut False=Decide 1668.NumberOut False +
1:NEXT(1596$);

1588$    TRANSPORT:    LAGVT,RStation8100;

;
;
;    Model statements for module: Decide 1669
;
1594$    BRANCH,    1:
                If,NL(RL18)&&NL(RL19)&&NZ(RL20,1)== 0,3562$,Yes:
                Else,3563$,Yes;
3562$    ASSIGN:    Decide 1669.NumberOut True=Decide 1669.NumberOut True +
1:NEXT(1589$);

3563$    ASSIGN:    Decide 1669.NumberOut False=Decide 1669.NumberOut False +
1:NEXT(1596$);

1589$    TRANSPORT:    LAGVT,RStation8100;

;
;
;    Model statements for module: Decide 1670
;
1595$    BRANCH,    1:
                If,NZ(RL18,7) &&NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10)
&&NL(RL19)&&NZ(RL20,1)== 0,3564$,Yes:
                Else,3565$,Yes;
3564$    ASSIGN:    Decide 1670.NumberOut True=Decide 1670.NumberOut True +
1:NEXT(1590$);

3565$    ASSIGN:    Decide 1670.NumberOut False=Decide 1670.NumberOut False +
1:NEXT(1596$);

1590$    TRANSPORT:    LAGVT,RStation8100;

;
;
;    Model statements for module: Station 122
;
1602$    STATION,    RStation7950;
3568$    DELAY:    0.0,,VA:NEXT(1607$);

;

```

```

;
;   Model statements for module: Decide 1671
;
1607$   BRANCH,      1:
        If,Entity.Type==RCar,1603$,Yes:
        If,Entity.Type==RTruck,1604$,Yes:
        If,Entity.Type==LCar,1605$,Yes:
        Else,1606$,Yes;
1606$   TRANSPORT:   LAGVT,RStation8100;

1603$   TRANSPORT:   RAGV,RStation8100;

1604$   TRANSPORT:   RAGVT,RStation8100;

1605$   TRANSPORT:   LAGV,RStation8100;

;
;
;   Model statements for module: Station 77
;

1608$   STATION,      RStation8100;
3573$   DELAY:        0.0,,VA:NEXT(1610$);

;
;
;   Model statements for module: Decide 658
;
1610$   BRANCH,      1:
        If,Entity.Type==RCar,3574$,Yes:
        Else,3575$,Yes;
3574$   ASSIGN:       Decide 658.NumberOut True=Decide 658.NumberOut True + 1:NEXT(1609$);

3575$   ASSIGN:       Decide 658.NumberOut False=Decide 658.NumberOut False + 1:NEXT(1612$);

1609$   TRANSPORT:   RAGV,RStation8400;

;
;
;   Model statements for module: Decide 659
;
1612$   BRANCH,      1:
        If,Entity.Type==RTruck,3576$,Yes:
        Else,3577$,Yes;
3576$   ASSIGN:       Decide 659.NumberOut True=Decide 659.NumberOut True + 1:NEXT(1611$);

3577$   ASSIGN:       Decide 659.NumberOut False=Decide 659.NumberOut False + 1:NEXT(1613$);

1611$   TRANSPORT:   RAGVT,RStation8400;

```



```

;
;
;   Model statements for module: Decide 660
;
1613$   BRANCH,      1:
        If,Entity.Type==LCar,3578$,Yes:
        Else,3579$,Yes;
3578$   ASSIGN:      Decide 660.NumberOut True=Decide 660.NumberOut True + 1:NEXT(1614$);

3579$   ASSIGN:      Decide 660.NumberOut False=Decide 660.NumberOut False + 1:NEXT(1615$);

1614$   TRANSPORT:   LAGV,RStation8400;
1615$   TRANSPORT:   LAGVT,RStation8400;

;
;
;   Model statements for module: Station 123
;
1616$   STATION,      RStation8400;
3582$   DELAY:        0.0,,VA:NEXT(1621$);

;
;
;   Model statements for module: Decide 1672
;
1621$   BRANCH,      1:
        If,Entity.Type==RCar,1617$,Yes:
        If,Entity.Type==RTruck,1618$,Yes:
        If,Entity.Type==LCar,1619$,Yes:
        Else,1620$,Yes;
1620$   TRANSPORT:   LAGVT,RStation8750;

1617$   TRANSPORT:   RAGV,RStation8750;

1618$   TRANSPORT:   RAGVT,RStation8750;

1619$   TRANSPORT:   LAGV,RStation8750;

;
;
;   Model statements for module: Station 82
;
1622$   STATION,      LParkStation;
3587$   DELAY:        0.0,,VA:NEXT(1637$);

1637$   FREE::NEXT(1636$);

```

```

;
;
;   Model statements for module: Dispose 11
;
1636$   ASSIGN:   Dispose 11.NumberOut=Dispose 11.NumberOut + 1;
3588$   DISPOSE:   No;

;
;
;   Model statements for module: Station 83
;

1623$   STATION,   RParkStation;
3591$   DELAY:     0.0,,VA:NEXT(1637$);

;
;
;   Model statements for module: Station 85
;

1624$   STATION,   RStation8750;
3594$   DELAY:     0.0,,VA:NEXT(1638$);

;
;
;   Model statements for module: Assign 246
;
1638$   ASSIGN:   QueueLength=QueueLength+NQ(QueueatTaperC) + NQ(QueueatTaperT);
                        CumuWaitTime=CumuWaitTime+TAVG(QueueatTaperC.WaitingTime) +
TAVG(QueueatTaperT.WaitingTime)
                        :NEXT(1632$);

;
;
;   Model statements for module: ReadWrite 1
;
1632$   WRITE,     Statistics:
                        IDENT,
                        Entity.Type,
                        Entity.StartTime,
                        Entity.TranTime,
                        Entity.WaitTime,
                        NQ(QueueatTaperC) + NQ(QueueatTaperT),
                        TAVG(QueueatTaperC.WaitingTime) +
TAVG(QueueatTaperT.WaitingTime):NEXT(1635$);

```

```

;
;
;   Model statements for module: Record 1
;
1635$   TALLY:      Entity,0,1:NEXT(1625$);

;
;
;   Model statements for module: Decide 1680
;
1625$   BRANCH,      1:
                If,Entity.Type==RCar,3595$,Yes:
                Else,3596$,Yes;
3595$   ASSIGN:      Decide 1680.NumberOut True=Decide 1680.NumberOut True +
1:NEXT(1639$);

3596$   ASSIGN:      Decide 1680.NumberOut False=Decide 1680.NumberOut False +
1:NEXT(1626$);

;
;
;   Model statements for module: Record 7
;
1639$   COUNT:      Right Car Number Out,1:NEXT(1628$);

1628$   TRANSPORT:   RAGV,RParkStation,1000000;

;
;
;   Model statements for module: Decide 1681
;
1626$   BRANCH,      1:
                If,Entity.Type==RTruck,3597$,Yes:
                Else,3598$,Yes;
3597$   ASSIGN:      Decide 1681.NumberOut True=Decide 1681.NumberOut True +
1:NEXT(1640$);

3598$   ASSIGN:      Decide 1681.NumberOut False=Decide 1681.NumberOut False +
1:NEXT(1627$);

;
;
;   Model statements for module: Record 8
;
1640$   COUNT:      Right Truck Number Out,1:NEXT(1629$);

1629$   TRANSPORT:   RAGVT,RParkStationT,1000000;

```

```

;
;
;   Model statements for module: Decide 1682
;
1627$   BRANCH,      1:
          If,Entity.Type==LCar,3599$,Yes:
          Else,3600$,Yes;
3599$   ASSIGN:      Decide 1682.NumberOut True=Decide 1682.NumberOut True +
1:NEXT(1641$);

3600$   ASSIGN:      Decide 1682.NumberOut False=Decide 1682.NumberOut False +
1:NEXT(1642$);

;
;
;   Model statements for module: Record 9
;
1641$   COUNT:       Left Car Number Out,1:NEXT(1630$);

1630$   TRANSPORT:   LAGV,LParkStation,1000000;

;
;
;   Model statements for module: Record 10
;
1642$   COUNT:       Left Truck Number Out,1:NEXT(1631$);

1631$   TRANSPORT:   LAGVT,LParkStationT,1000000;

;
;
;   Model statements for module: Station 126
;

1633$   STATION,     RParkStationT;
3603$   DELAY:       0.0,,VA:NEXT(1637$);

;
;
;   Model statements for module: Station 127
;

1634$   STATION,     LParkStationT;
3606$   DELAY:       0.0,,VA:NEXT(1637$);

```

SIMAN Codes .exp File

[RESOURCES]

[VARIABLES]

Decide 1466.NumberOut False
CheckGapL7600Speed10less.NumberOut False
checkvehicletype7500.NumberOut True
CheckGapL7450Speed75.NumberOut False
Decide 1432.NumberOut True
Decide 1743.NumberOut False
CheckGapL7450Speed10t.NumberOut False
Decide 1463.NumberOut True
CheckGapL7600Speed75t.NumberOut False
Decide 1732.NumberOut False
Decide 1345.NumberOut False
CheckGapL7250Speed55_75.NumberOut False
SpeedatRStation7600
CheckGapL7850Speed55_75less.NumberOut True
checkvehicletype7750.NumberOut False
Decide 1375.NumberOut True
R.Vehicle Enters1345_1400.NumberOut
CheckGapL7800Speed35_55t.NumberOut False
CheckGapL7450Speed55_75lesst.NumberOut False
CheckGapL6450Speed75t.NumberOut True
Decide 1290.NumberOut False
CheckGapL7550Speed75.NumberOut False
Decide 1791.NumberOut False
L.Vehicle Entry900_915.NumberOut
Decide 1725.NumberOut False
Decide 1681.NumberOut False
CheckGapL7700Speed10_35lesst.NumberOut True
CheckGapL7600Speed35_55lesst.NumberOut True
Decide 1283.NumberOut True
Decide 1714.NumberOut False
Decide 1670.NumberOut False
CheckGapL7850Speed10t.NumberOut False
CheckGapL7800Speed10_35t.NumberOut True
CheckGapL7600Speed75lesst.NumberOut True
Decide 1283.NumberOut False
CheckGapL7050Speed35_55.NumberOut True
R.Vehicle Enters930_945.NumberOut
CheckGapL7650Speed75.NumberOut False
CheckGapL7500Speed55_75lesst.NumberOut True
checkvehicletype7250.NumberOut True
Decide 1784.NumberOut False
L.Vehicle Entry1800_1815.NumberOut
CheckGapL7050Speed10lesst.NumberOut True
Decide 1710.NumberOut True
CheckGapL7750Speed10_35less.NumberOut False
Decide 1773.NumberOut False
R.Vehicle Enters1030_1045.NumberOut
Decide 1718.NumberOut False

CheckGapL7650Speed10_35less.NumberOut True
checkvehicletype6700.NumberOut True
checkvehicletype7950.NumberOut True
CheckGapL7450Speed35_55less.NumberOut False
Decide 1763.NumberOut True
Decide 1741.NumberOut True
Decide 1707.NumberOut False
Decide 1663.NumberOut False
CheckGapL7550Speed10_35.NumberOut False
CheckGapL7050Speed55_75t.NumberOut False
R.Vehicle Enters1415_1430.NumberOut
Decide 1652.NumberOut False
CheckGapL7600Speed10_35t.NumberOut True
Decide 1794.NumberOut True
Decide 1772.NumberOut True
Decide 1750.NumberOut True
CheckGapL7550Speed10_35.NumberOut True
CheckGapL7500Speed75.NumberOut True
CheckGapL7450Speed55_75less.NumberOut True
CheckGapL7350Speed55_75less.NumberOut False
Decide 1777.NumberOut False
Decide 1662.NumberOut True
Decide 1640.NumberOut True
CheckGapL7750Speed75.NumberOut False
Decide 542.NumberOut True
CheckGapL6950Speed35_55t.NumberOut True
Decide 552.NumberOut False
CheckGapL6700Speed55_75.NumberOut False
Decide 1766.NumberOut False
L.Vehicle Entry845_900.NumberOut
Decide 1379.NumberOut False
CheckGapL6950Speed55_75.NumberOut True
L.Vehicle Entry715_73.NumberOut
CheckGapL6450Speed10lesst.NumberOut False
Decide 1693.NumberOut True
Decide 1656.NumberOut False
CheckGapL7500Speed10t.NumberOut False
checkvehicletype6950.NumberOut False
Decide 1561.NumberOut True
CheckGapL7350Speed10_35.NumberOut False
CheckGapL6450Speed10_35less.NumberOut True
Decide 1708.NumberOut True
Decide 1645.NumberOut False
CheckGapL6950Speed55_75t.NumberOut True
CheckGapL7750Speed55_75.NumberOut True
CheckGapL7550Speed10less.NumberOut True
CheckGapL7350Speed55_75lesst.NumberOut False
CheckGapL7150Speed75less.NumberOut False
Decide 1739.NumberOut True
Decide 1717.NumberOut True
CheckGapL7250Speed75t.NumberOut False
Decide 545.NumberOut False
Decide 1759.NumberOut False

R.Vehicle Enters1745_1800.NumberOut
 CheckGapL7550Speed10lesst.NumberOut True
 CheckGapL7500Speed55_75less.NumberOut True
 CheckGapL7450Speed75lesst.NumberOut True
 DecideVehicleTypeRight0.NumberOut True
 Decide 659.NumberOut False
 Decide 1480.NumberOut False
 CheckGapL7700Speed35_55t.NumberOut True
 Decide 1638.NumberOut True
 CheckGapL7500Speed75t.NumberOut True
 CheckGapL6450Speed75less.NumberOut True
 Decide 1638.NumberOut False
 CheckGapL7550Speed55_75t.NumberOut False
 CheckGapL7250Speed10_35less.NumberOut True
 CheckGapL7150Speed10_35.NumberOut False
 CheckGapL7600Speed35_55.NumberOut True
 Decide 1669.NumberOut True
 Decide 1627.NumberOut False
 CheckGapL7900Speed35_55less.NumberOut False
 CheckGapL7350Speed10less.NumberOut True
 CheckGapL6700Speed55_75lesst.NumberOut True
 L.Vehicle Entry1145_1200.NumberOut
 CheckGapL7900Speed10t.NumberOut False
 CheckGapL7550Speed35_55lesst.NumberOut False
 CheckGapL7050Speed75t.NumberOut True
 Decide 538.NumberOut False
 CheckGapL7700Speed35_55less.NumberOut False
 CheckGapL7150Speed10_35.NumberOut True
 CheckGapL6950Speed10_35less.NumberOut False
 CheckGapL6450Speed10less.NumberOut False
 CheckGapL7800Speed55_75less.NumberOut False
 CheckGapL7750Speed75less.NumberOut True
 CheckGapL6700Speed10lesst.NumberOut False
 Decide 1697.NumberOut False
 CheckGapL7500Speed35_55t.NumberOut True
 CheckGapL7250Speed10_35lesst.NumberOut True
 Decide 873.NumberOut True
 CheckGapL7600Speed75t.NumberOut True
 Decide4VehicleTypeR6700.NumberOut False
 R.Vehicle Enters715_730.NumberOut
 R.Vehicle Enters1430_1445.NumberOut
 CheckGapL7600Speed10.NumberOut True
 Decide2VehicleTypeR6700.NumberOut False
 Decide 1009.NumberOut False
 CheckGapL7150Speed10less.NumberOut True
 R.Vehicle Enters1100_1115.NumberOut
 Decide 1467.NumberOut True
 CheckGapL7650Speed75less.NumberOut False
 CheckGapL7350Speed55_75.NumberOut True
 CheckGapL7150Speed75t.NumberOut True
 R.Vehicle Enters1815_1830.NumberOut
 CheckGapL7500Speed75less.NumberOut False
 CheckGapL7250Speed35_55.NumberOut False

Decide 928.NumberOut True
CheckGapL6950Speed10_35lesst.NumberOut False
CheckGapL7900Speed10lesst.NumberOut True
CheckGapL7850Speed55_75t.NumberOut True
CheckGapL7700Speed75t.NumberOut True
Decide 1344.NumberOut True
CheckGapL7050Speed35_55less.NumberOut True
CheckGapL6700Speed10_35lesst.NumberOut False
CheckGapL7550Speed10_35lesst.NumberOut False
CheckGapL7900Speed55_75t.NumberOut False
Decide 1721.NumberOut False
L.Vehicle Entry1215_1230.NumberOut
CheckGapL7450Speed10_35t.NumberOut False
CheckGapL6450Speed75lesst.NumberOut True
Decide 1710.NumberOut False
CheckGapL7050Speed35_55.NumberOut False
CheckGapL6950Speed10less.NumberOut False
CheckGapL7150Speed35_55t.NumberOut False
Decide 748.NumberOut True
CheckGapL7800Speed75t.NumberOut True
Decide 1437.NumberOut False
Decide 1780.NumberOut False
L.Vehicle Entry2000_2015.NumberOut
CheckGapL7050Speed55_75.NumberOut False
CheckGapL6450Speed75.NumberOut True
CheckGapL7800Speed55_75lesst.NumberOut False
CheckGapL7500Speed10_35lesst.NumberOut True
CheckGapL7050Speed10lesst.NumberOut False
Decide 1745.NumberOut True
Decide 1723.NumberOut True
Decide 1703.NumberOut False
Decide 1701.NumberOut True
CheckGapL7900Speed55_75less.NumberOut True
CheckGapL7850Speed35_55less.NumberOut True
checkvehicletype7800.NumberOut True
CheckGapL6450Speed10_35lesst.NumberOut True
L.Vehicle Entry1545_1600.NumberOut
Decide 1776.NumberOut True
Decide 1754.NumberOut True
Decide 1732.NumberOut True
Decide 1644.NumberOut True
CheckGapL7450Speed10less.NumberOut False
CheckGapL6950Speed35_55.NumberOut True
Decide 546.NumberOut True
L.Vehicle Entry815_830.NumberOut
CheckGapL7550Speed10_35less.NumberOut False
Decide 1785.NumberOut True
Decide 1762.NumberOut False
checkvehicletype7600.NumberOut False
Decide 1375.NumberOut False
CheckGapL6700Speed35_55.NumberOut False
Decide 1631.NumberOut True
CheckGapL7450Speed75.NumberOut True

CheckGapL7250Speed35_55less.NumberOut False
 Decide 1751.NumberOut False
 R.Vehicle Enters900_915.NumberOut
 Decide 1697.NumberOut True
 SpeedatRStation7550
 CheckGapL6950Speed75lesst.NumberOut True
 R.Vehicle Enters1830_1845.NumberOut
 CheckGapL7750Speed35_55.NumberOut True
 R.Vehicle Enters1500_1515.NumberOut
 CheckGapL7650Speed35_55t.NumberOut False
 Decide 1008.NumberOut True
 CheckGapL7150Speed55_75less.NumberOut False
 CheckGapL6700Speed55_75less.NumberOut True
 Decide 1630.NumberOut False
 Decide 541.NumberOut False
 Decide 1755.NumberOut False
 CheckGapL7450Speed10_35lesst.NumberOut False
 CheckGapL7700Speed10_35less.NumberOut True
 CheckGapL6450Speed10_35t.NumberOut False
 Decide 1744.NumberOut False
 L.Vehicle Entry1230_1245.NumberOut
 SpeedatRStation6700
 Decide 1634.NumberOut False
 Decide1VehicleTypeR6450.NumberOut False
 CheckGapL7850Speed75.NumberOut False
 checkvehicletype7550.NumberOut True
 CheckGapL6950Speed75t.NumberOut False
 L.Vehicle Entry1615_1630.NumberOut
 CheckGapL7450Speed35_55less.NumberOut True
 Decide 1748.NumberOut False
 CheckGapL7800Speed35_55.NumberOut False
 Decide 1372.NumberOut True
 Decide 1737.NumberOut False
 CheckGapL7800Speed10less.NumberOut False
 CheckGapL6950Speed35_55lesst.NumberOut False
 CheckGapL6700Speed10_35lesst.NumberOut True
 Decide 1693.NumberOut False
 SpeedatRStation7700
 CheckGapL7500Speed55_75.NumberOut True
 R.Vehicle Enters845_900.NumberOut
 Decide 1726.NumberOut False
 Decide 1682.NumberOut False
 CheckGapL7850Speed35_55lesst.NumberOut True
 CheckGapL6700Speed35_55lesst.NumberOut False
 CheckGapL7800Speed55_75.NumberOut False
 CheckGapL7800Speed10_35less.NumberOut False
 CheckGapL7650Speed35_55lesst.NumberOut True
 CheckGapL7350Speed35_55.NumberOut True
 Decide 1284.NumberOut False
 CheckGapL7150Speed10lesst.NumberOut False
 Decide 1796.NumberOut False
 CheckGapL7750Speed55_75lesst.NumberOut True
 CheckGapL7500Speed35_55less.NumberOut False

Decide 1280.NumberOut True
Decide 1561.NumberOut False
CheckGapL7650Speed75t.NumberOut False
CheckGapL7600Speed55_75less.NumberOut False
CheckGapL7550Speed10.NumberOut True
CheckGapL7500Speed35_55less.NumberOut True
Decide 1317.NumberOut True
CheckGapL6450Speed35_55less.NumberOut False
CheckGapL7050Speed10_35lesst.NumberOut True
CheckGapL6950Speed10_35t.NumberOut False
CheckGapL6450Speed10t.NumberOut False
Decide 1719.NumberOut False
checkvehicletype7900.NumberOut False
CheckGapL7350Speed10lesst.NumberOut False
CheckGapL6950Speed75t.NumberOut True
Decide 1664.NumberOut False
CheckGapL7900Speed35_55lesst.NumberOut False
CheckGapL7850Speed35_55t.NumberOut True
CheckGapL7600Speed55_75.NumberOut False
DecideVehicleTypeLeft0.NumberOut True
CheckGapL7850Speed10_35.NumberOut True
Decide 1277.NumberOut False
Decide2VehicleTypeR6700.NumberOut True
Decide 1789.NumberOut False
Decide 1379.NumberOut True
Decide 1782.NumberOut True
Decide 1760.NumberOut True
CheckGapL7550Speed75less.NumberOut True
Decide 1778.NumberOut False
CheckGapL7800Speed75lesst.NumberOut True
Decide 552.NumberOut True
CheckGapL6700Speed10.NumberOut False
R.Vehicle Enters1900_1915.NumberOut
Decide 1791.NumberOut True
Decide 1668.NumberOut False
Decide 1278.NumberOut True
CheckGapL7250Speed10lesst.NumberOut True
Decide 1657.NumberOut False
Decide 1433.NumberOut False
CheckGapL7250Speed55_75lesst.NumberOut False
Decide 955.NumberOut False
DecideVehicleTypeLeft0.NumberOut False
CheckGapL7650Speed35_55t.NumberOut True
CheckGapL7250Speed75t.NumberOut True
L.Vehicle Entry1630_1645.NumberOut
CheckGapL7150Speed10t.NumberOut False
Decide 1690.NumberOut True
CheckGapL7250Speed55_75t.NumberOut False
L.Vehicle Entry1300_1315.NumberOut
CheckGapL7350Speed75less.NumberOut True
Decide 1727.NumberOut True
Decide 1705.NumberOut True
CheckGapL6700Speed75less.NumberOut False

CheckGapL7650Speed55_75t.NumberOut True
CheckGapL7900Speed10_35lesst.NumberOut False
Decide 1758.NumberOut True
Decide 1736.NumberOut True
Decide 1714.NumberOut True
SpeedatRStation7050
Decide 1481.NumberOut False
CheckGapL6950Speed10_35.NumberOut False
Decide 1626.NumberOut True
CheckGapL7700Speed10_35.NumberOut True
CheckGapL6950Speed10lesst.NumberOut False
Decide 1639.NumberOut False
CheckGapL7900Speed10_35.NumberOut False
CheckGapL7450Speed35_55lesst.NumberOut False
CheckGapL7350Speed75t.NumberOut True
Decide 1789.NumberOut True
Decide 1767.NumberOut True
Decide 1371.NumberOut False
CheckGapL6950Speed55_75lesst.NumberOut True
CheckGapL6700Speed35_55less.NumberOut False
Decide2VehicleTypeR6450.NumberOut True
Decide 1657.NumberOut True
CheckGapL7450Speed10_35.NumberOut True
CheckGapL7150Speed75less.NumberOut True
Decide 1666.NumberOut True
CheckGapL7900Speed75t.NumberOut True
CheckGapL7550Speed10t.NumberOut False
CheckGapL7450Speed55_75t.NumberOut True
CheckGapL7350Speed75less.NumberOut False
CheckGapL7250Speed10lesst.NumberOut False
Decide 1688.NumberOut True
CheckGapL7700Speed75t.NumberOut False
CheckGapL7550Speed55_75less.NumberOut True
Decide 1698.NumberOut False
CheckGapL7900Speed55_75.NumberOut True
Decide 1534.NumberOut True
CheckGapL7750Speed10lesst.NumberOut True
Decide 1463.NumberOut False
CheckGapL7350Speed10_35less.NumberOut False
CheckGapL6450Speed10.NumberOut False
CheckGapL7700Speed10_35.NumberOut False
CheckGapL7600Speed10lesst.NumberOut True
CheckGapL7450Speed75t.NumberOut True
CheckGapL6950Speed10less.NumberOut True
CheckGapL7650Speed55_75.NumberOut True
CheckGapL7450Speed10lesst.NumberOut False
CheckGapL7350Speed75.NumberOut True
CheckGapL7050Speed35_55less.NumberOut False
Decide 1740.NumberOut False
CheckGapL7900Speed10less.NumberOut True
CheckGapL7800Speed10_35t.NumberOut False
CheckGapL7750Speed55_75t.NumberOut False
R.Vehicle Enters1245_1300.NumberOut

Decide 1433.NumberOut True
 CheckGapL7900Speed35_55less.NumberOut True
 Decide 1467.NumberOut False
 CheckGapL7500Speed35_55t.NumberOut False
 CheckGapL7250Speed55_75t.NumberOut True
 Decide 1464.NumberOut True
 CheckGapL6450Speed10_35t.NumberOut True
 CheckGapL7550Speed75t.NumberOut True
 CheckGapL7500Speed10_35.NumberOut False
 checkvehicletype7250.NumberOut False
 Decide 1733.NumberOut False
 R.Vehicle Enters815_830.NumberOut
 L.Vehicle Entry730_745.NumberOut
 CheckGapL7700Speed10less.NumberOut True
 Decide 1722.NumberOut False
 CheckGapL7350Speed10_35less.NumberOut True
 CheckGapL7050Speed10_35.NumberOut True
 CheckGapL6700Speed55_75.NumberOut True
 CheckGapL7500Speed10lesst.NumberOut False
 Decide 1291.NumberOut False
 CheckGapL7850Speed75less.NumberOut False
 CheckGapL7750Speed10_35t.NumberOut True
 L.Vehicle Entry1700_1715.NumberOut
 Decide 1280.NumberOut False
 CheckGapL6700Speed35_55less.NumberOut True
 Decide 747.NumberOut False
 Decide 1792.NumberOut False
 CheckGapL7700Speed75less.NumberOut False
 Decide 1284.NumberOut True
 CheckGapL7150Speed55_75less.NumberOut True
 CheckGapL6450Speed10.NumberOut True
 Decide 1781.NumberOut False
 CheckGapL7700Speed55_75lesst.NumberOut False
 CheckGapL7650Speed75t.NumberOut True
 CheckGapL7350Speed35_55lesst.NumberOut False
 CheckGapL7150Speed10less.NumberOut False
 Decide 1715.NumberOut False
 QueueLength
 CheckGapL7750Speed10_35lesst.NumberOut True
 CheckGapL7500Speed10less.NumberOut True
 CheckGapL7250Speed55_75.NumberOut True
 R.Vehicle Enters1315_1330.NumberOut
 Decide 1711.NumberOut True
 Decide 1704.NumberOut False
 Decide 1660.NumberOut False
 CheckGapL7600Speed10_35less.NumberOut False
 Decide 1317.NumberOut False
 CheckGapL7450Speed35_55lesst.NumberOut True
 CheckGapL7600Speed35_55.NumberOut False
 CheckGapL7550Speed10_35t.NumberOut True
 Decide 1785.NumberOut False
 CheckGapL7900Speed75less.NumberOut True
 CheckGapL7600Speed10t.NumberOut False

CheckGapL7550Speed55_75lesst.NumberOut True
Decide 1764.NumberOut True
Decide 1742.NumberOut True
Decide 1720.NumberOut True
checkvehicletype7150.NumberOut True
Decide 1774.NumberOut False
CheckGapL7450Speed10.NumberOut True
Decide 1795.NumberOut True
Decide 1773.NumberOut True
Decide 1763.NumberOut False
Decide 1751.NumberOut True
Decide 1708.NumberOut False
CheckGapL7500Speed10.NumberOut True
CheckGapL6700Speed55_75t.NumberOut True
L.Vehicle Entry1130_1145.NumberOut
Decide 1663.NumberOut True
checkvehicletype7850.NumberOut True
CheckGapL7150Speed35_55less.NumberOut True
Decide 543.NumberOut True
CheckGapL7550Speed10lesst.NumberOut False
Decide 1650.NumberOut True
Decide 1642.NumberOut False
CheckGapL7500Speed10.NumberOut False
CheckGapL7350Speed10_35t.NumberOut True
R.Vehicle Enters1645_1700.NumberOut
Decide 1694.NumberOut True
CheckGapL7350Speed10_35lesst.NumberOut False
Decide 1767.NumberOut False
Decide 1562.NumberOut True
Decide 1709.NumberOut True
Decide 1681.NumberOut True
CheckGapL7150Speed75lesst.NumberOut True
Decide 542.NumberOut False
CheckGapL6450Speed55_75.NumberOut False
Decide 1756.NumberOut False
SpeedatRStation7650
CheckGapL7650Speed10less.NumberOut False
checkvehicletype7550.NumberOut False
Decide 1718.NumberOut True
Decide 1646.NumberOut False
CheckGapL6950Speed10_35less.NumberOut True
CheckGapL7500Speed10less.NumberOut False
CheckGapL6950Speed75.NumberOut True
L.Vehicle Entry1045_1100.NumberOut
Decide 1749.NumberOut True
CheckGapL7600Speed10.NumberOut False
Decide 546.NumberOut False
Decide 1639.NumberOut True
CheckGapL7750Speed55_75.NumberOut False
Decide 1470.NumberOut True
Decide 900.NumberOut True
CheckGapL7900Speed35_55lesst.NumberOut True
CheckGapL7850Speed55_75lesst.NumberOut False

CheckGapL7600Speed10lesst.NumberOut False
 Decide 1749.NumberOut False
 Decide 1507.NumberOut True
 Decide 1648.NumberOut True
 CheckGapL7900Speed35_55.NumberOut True
 CheckGapL6950Speed10_35lesst.NumberOut True
 Decide 1470.NumberOut False
 CheckGapL7450Speed35_55t.NumberOut True
 CheckGapL6450Speed10lesst.NumberOut True
 R.Vehicle Enters1330_1345.NumberOut
 Decide 1694.NumberOut False
 CheckGapL7800Speed55_75lesst.NumberOut True
 CheckGapL7750Speed10_35less.NumberOut True
 Decide 1628.NumberOut False
 CheckGapL7800Speed10lesst.NumberOut False
 Decide 981.NumberOut False
 R.Vehicle Enters1000_1015.NumberOut
 Decide 1406.NumberOut True
 Decide 539.NumberOut False
 Decide 874.NumberOut True
 R.Vehicle Enters1715_1730.NumberOut
 CheckGapL7750Speed35_55less.NumberOut False
 Decide 1437.NumberOut True
 Decide 1290.NumberOut True
 SpeedatRStation7800
 CheckGapL7850Speed55_75less.NumberOut False
 Decide 1507.NumberOut False
 CheckGapL7650Speed75lesst.NumberOut True
 CheckGapL7800Speed35_55lesst.NumberOut False
 Decide 1468.NumberOut True
 CheckGapL7500Speed75lesst.NumberOut True
 CheckGapL7250Speed35_55t.NumberOut True
 CheckGapL7150Speed10_35less.NumberOut False
 L.Vehicle Entry1115_1130.NumberOut
 CheckGapL7800Speed10_35less.NumberOut True
 CheckGapL7250Speed75.NumberOut True
 L.Vehicle Entry1530_1545.NumberOut
 CheckGapL7600Speed55_75t.NumberOut False
 CheckGapL6700Speed35_55.NumberOut True
 Decide 1345.NumberOut True
 CheckGapL7150Speed10_35t.NumberOut False
 L.Vehicle Entry1200_1215.NumberOut
 CheckGapL7650Speed10lesst.NumberOut False
 CheckGapL7600Speed55_75less.NumberOut True
 CheckGapL7150Speed55_75lesst.NumberOut False
 Decide 1669.NumberOut False
 CheckGapL7550Speed35_55less.NumberOut True
 CheckGapL7500Speed35_55.NumberOut True
 CheckGapL7050Speed35_55t.NumberOut True
 CheckGapL6950Speed10lesst.NumberOut True
 checkvehicletype7850.NumberOut False
 CheckGapL7850Speed10lesst.NumberOut False
 Decide 1434.NumberOut False

Dispose 11.NumberOut
Decide 1711.NumberOut False
CheckGapL7850Speed10_35.NumberOut False
CheckGapL7050Speed55_75t.NumberOut True
L.Vehicle Entry1445_1500.NumberOut
CheckGapL7800Speed10_35lesst.NumberOut False
CheckGapL6450Speed55_75.NumberOut True
Decide 1700.NumberOut False
CheckGapL6700Speed35_55t.NumberOut False
Decide 1438.NumberOut False
L.Vehicle Entry700_715.NumberOut
CheckGapL7700Speed10lesst.NumberOut False
Decide 1746.NumberOut True
Decide 1724.NumberOut True
Decide 1702.NumberOut True
CheckGapL7050Speed75less.NumberOut False
Decide 1770.NumberOut False
CheckGapL7600Speed35_55less.NumberOut True
Decide 538.NumberOut True
CheckGapL7450Speed10lesst.NumberOut True
CheckGapL6950Speed35_55less.NumberOut False
Decide 1777.NumberOut True
Decide 1755.NumberOut True
R.Vehicle Enters1730_1745.NumberOut
Decide 1733.NumberOut True
Decide 1372.NumberOut False
CheckGapL7550Speed10_35lesst.NumberOut True
CheckGapL6950Speed75less.NumberOut True
R.Vehicle Enters945_1000.NumberOut
Decide 1645.NumberOut True
CheckGapL7900Speed10lesst.NumberOut False
CheckGapL7650Speed10_35t.NumberOut False
Decide 547.NumberOut True
CheckGapL6700Speed35_55t.NumberOut True
R.Vehicle Enters1400_1415.NumberOut
CheckGapL7250Speed35_55lesst.NumberOut True
Decide 1786.NumberOut True
R.Vehicle Enters730_745.NumberOut
SpeedatRStation7150
CheckGapL7850Speed10less.NumberOut True
checkvehicletype7700.NumberOut True
CheckGapL7500Speed10_35t.NumberOut False
Decide3VehicleTypeR6700.NumberOut True
Decide 1654.NumberOut True
Decide 1632.NumberOut True
CheckGapL7350Speed55_75lesst.NumberOut True
CheckGapL7350Speed35_55t.NumberOut False
Decide 1698.NumberOut True
CheckGapL7750Speed75t.NumberOut True
checkvehicletype7700.NumberOut False
CheckGapL7350Speed10.NumberOut True
CheckGapL7150Speed55_75lesst.NumberOut True
CheckGapL7350Speed75t.NumberOut False

Decide 1752.NumberOut False
Decide 1009.NumberOut True
CheckGapL6450Speed35_55.NumberOut False
Decide 1741.NumberOut False
L.Vehicle Entry1515_1530.NumberOut
CheckGapL7700Speed75less.NumberOut True
L.Vehicle Entry1930_1945.NumberOut
Decide 1631.NumberOut False
Decide 1468.NumberOut False
L.Vehicle Entry1600_1615.NumberOut
L.Vehicle Entry645_700.NumberOut
CheckGapL7850Speed75t.NumberOut True
CheckGapL7750Speed10lesst.NumberOut False
CheckGapL7050Speed10.NumberOut False
CheckGapL7750Speed35_55.NumberOut False
CheckGapL7600Speed10_35.NumberOut True
CheckGapL7550Speed75less.NumberOut False
Decide 1745.NumberOut False
L.Vehicle Entry915_930.NumberOut
CheckGapL7800Speed35_55t.NumberOut True
Decide 1734.NumberOut False
Decide 1690.NumberOut False
CheckGapL7800Speed10lesst.NumberOut True
Decide 1373.NumberOut True
CheckGapL7150Speed10_35t.NumberOut True
Decide 1624.NumberOut False
CheckGapL7750Speed75t.NumberOut False
CheckGapL7500Speed75less.NumberOut True
checkvehicletype7450.NumberOut True
CheckGapL7850Speed35_55t.NumberOut False
CheckGapL7800Speed55_75t.NumberOut True
CheckGapL6950Speed10.NumberOut False
CheckGapL6700Speed75t.NumberOut False
L.Vehicle Entry1845_1900.NumberOut
CheckGapL7800Speed10_35lesst.NumberOut True
CheckGapL7800Speed55_75.NumberOut True
CheckGapL7700Speed35_55lesst.NumberOut True
CheckGapL7600Speed55_75lesst.NumberOut False
CheckGapL7250Speed35_55lesst.NumberOut False
CheckGapL7150Speed10.NumberOut False
Decide 748.NumberOut False
Decide 1793.NumberOut False
Decide 1738.NumberOut False
CheckGapL7700Speed35_55t.NumberOut False
CheckGapL7550Speed35_55.NumberOut False
Decide 900.NumberOut False
CheckGapL7500Speed35_55lesst.NumberOut True
Decide 1727.NumberOut False
CheckGapL7650Speed35_55.NumberOut True
CheckGapL7600Speed55_75lesst.NumberOut True
Decide 1281.NumberOut True
CheckGapL6450Speed35_55lesst.NumberOut True
CheckGapL7850Speed75.NumberOut True

CheckGapL7700Speed10.NumberOut False
 CheckGapL6450Speed55_75t.NumberOut False
 CheckGapL7850Speed10_35less.NumberOut False
 CheckGapL7550Speed55_75.NumberOut False
 Decide 1318.NumberOut True
 Decide 1285.NumberOut False
 Decide 1797.NumberOut False
 CheckGapL7900Speed75.NumberOut True
 L.Vehicle Entry1945_2000.NumberOut
 Decide 1562.NumberOut False
 CheckGapL7600Speed55_75t.NumberOut True
 CheckGapL7550Speed35_55less.NumberOut False
 Decide 1786.NumberOut False
 R.Vehicle Enters1800_1815.NumberOut
 SpeedatRStation5750
 CheckGapL7250Speed10.NumberOut False
 CheckGapL7650Speed55_75less.NumberOut False
 CheckGapL7250Speed10t.NumberOut False
 CheckGapL7900Speed75less.NumberOut False
 CheckGapL7450Speed55_75less.NumberOut False
 CheckGapL6700Speed10t.NumberOut True
 Decide 1761.NumberOut True
 Decide 1709.NumberOut False
 CheckGapL7800Speed10.NumberOut False
 Decide 1278.NumberOut False
 Decide 1654.NumberOut False
 CheckGapL7350Speed10less.NumberOut False
 CheckGapL7350Speed55_75.NumberOut False
 CheckGapL7250Speed10_35lesst.NumberOut False
 Decide 1792.NumberOut True
 Decide 1770.NumberOut True
 Decide 1779.NumberOut False
 Decide 1660.NumberOut True
 Decide 1279.NumberOut True
 Decide 540.NumberOut True
 CheckGapL6450Speed35_55.NumberOut True
 L.Vehicle Entry1915_1930.NumberOut
 CheckGapL7450Speed10_35less.NumberOut True
 CheckGapL7350Speed10.NumberOut False
 Decide 1768.NumberOut False
 checkvehicletype7050.NumberOut False
 Decide 1691.NumberOut True
 Decide 1658.NumberOut False
 CheckGapL6950Speed10.NumberOut True
 CheckGapL6700Speed75lesst.NumberOut True
 CheckGapL7250Speed35_55.NumberOut True
 R.Vehicle Enters1230_1245.NumberOut
 Decide 1728.NumberOut True
 Decide 1706.NumberOut True
 CheckGapL7900Speed10.NumberOut False
 CheckGapL7250Speed55_75less.NumberOut True
 CheckGapL6950Speed55_75t.NumberOut False
 CheckGapL7650Speed10t.NumberOut False

CheckGapL7150Speed55_75.NumberOut False
 CheckGapL7800Speed75t.NumberOut False
 CheckGapL7750Speed55_75lesst.NumberOut False
 Decide 1759.NumberOut True
 Decide 1737.NumberOut True
 Decide 1715.NumberOut True
 CheckGapL7900Speed10_35t.NumberOut True
 CheckGapL6950Speed10_35.NumberOut True
 Decide 547.NumberOut False
 SpeedatRStation7750
 Decide 1627.NumberOut True
 CheckGapL7500Speed55_75lesst.NumberOut False
 CheckGapL7450Speed10.NumberOut False
 CheckGapL6700Speed35_55lesst.NumberOut True
 Decide 1768.NumberOut True
 Decide 1658.NumberOut True
 Decide 1636.NumberOut True
 CheckGapL7800Speed35_55less.NumberOut False
 CheckGapL7750Speed10_35.NumberOut True
 Decide 1471.NumberOut False
 CheckGapL7350Speed75lesst.NumberOut True
 Decide 1667.NumberOut True
 CheckGapL7900Speed55_75less.NumberOut False
 CheckGapL7850Speed10less.NumberOut False
 CheckGapL7650Speed10_35.NumberOut False
 Decide 927.NumberOut False
 R.Vehicle Enters1145_1200.NumberOut
 Decide 1689.NumberOut True
 L.Vehicle Entry1015_1030.NumberOut
 CheckGapL7350Speed10_35lesst.NumberOut True
 L.Vehicle Entry945_1000.NumberOut
 Decide 1535.NumberOut True
 CheckGapL7700Speed10_35t.NumberOut True
 CheckGapL7450Speed55_75t.NumberOut False
 Decide 981.NumberOut True
 CheckGapL7050Speed35_55lesst.NumberOut True
 CheckGapL7700Speed35_55lesst.NumberOut False
 CheckGapL6450Speed75lesst.NumberOut False
 Decide 1699.NumberOut False
 Decide 1508.NumberOut False
 CheckGapL7050Speed10_35less.NumberOut True
 Decide 1464.NumberOut False
 Decide 1434.NumberOut True
 R.Vehicle Enters700_715.NumberOut
 CheckGapL7450Speed10_35.NumberOut False
 CheckGapL6450Speed55_75lesst.NumberOut False
 Decide 1465.NumberOut True
 Decide 1730.NumberOut False
 SpeedatRStation7900
 CheckGapL7650Speed10less.NumberOut True
 CheckGapL7500Speed10_35t.NumberOut True
 CheckGapL7050Speed75t.NumberOut False
 CheckGapL7050Speed55_75lesst.NumberOut False

CheckGapL6700Speed10_35t.NumberOut False
CheckGapL7800Speed35_55.NumberOut True
checkvehicletype7350.NumberOut False
CheckGapL7850Speed75lesst.NumberOut True
CheckGapL7350Speed10_35.NumberOut True
CheckGapL7700Speed75lesst.NumberOut True
CheckGapL7250Speed10_35.NumberOut False
R.Vehicle Enters1215_1230.NumberOut
Decide 1723.NumberOut False
CheckGapL7700Speed10_35lesst.NumberOut False
CheckGapL7700Speed10t.NumberOut False
R.Vehicle Enters1630_1645.NumberOut
CheckGapL7850Speed10_35less.NumberOut True
CheckGapL7450Speed10less.NumberOut True
Decide 1285.NumberOut True
Decide 1712.NumberOut False
CheckGapL7150Speed10lesst.NumberOut True
CheckGapL6700Speed75lesst.NumberOut False
R.Vehicle Enters1300_1315.NumberOut
CheckGapL7800Speed55_75t.NumberOut False
Decide 1281.NumberOut False
CheckGapL7350Speed10_35t.NumberOut False
CheckGapL7850Speed75less.NumberOut True
CheckGapL7600Speed10_35lesst.NumberOut True
Decide 1782.NumberOut False
R.Vehicle Enters2000_2015.NumberOut
CheckGapL7650Speed55_75less.NumberOut True
CheckGapL7600Speed35_55t.NumberOut True
CheckGapL7550Speed55_75.NumberOut True
Decide 1712.NumberOut True
CheckGapL7050Speed35_55t.NumberOut False
CheckGapL6700Speed75.NumberOut False
Decide 1771.NumberOut False
Decide 1716.NumberOut False
L.Vehicle Entry1030_1045.NumberOut
CheckGapL6700Speed10less.NumberOut False
Decide 1765.NumberOut True
Decide 1743.NumberOut True
R.Vehicle Enters915_930.NumberOut
Decide 1721.NumberOut True
Decide 1705.NumberOut False
L.Vehicle Entry830_845.NumberOut
Decide 1661.NumberOut False
Decide 1318.NumberOut False
CheckGapL7250Speed10less.NumberOut True
CheckGapL7650Speed10_35less.NumberOut False
R.Vehicle Enters1545_1600.NumberOut
L.Vehicle Entry1415_1430.NumberOut
Decide 1650.NumberOut False
CheckGapL7900Speed10_35less.NumberOut True
CheckGapL7800Speed75.NumberOut True
Decide 1774.NumberOut True
Decide 1752.NumberOut True

Decide 1796.NumberOut True
Decide 1730.NumberOut True
CheckGapL7350Speed35_55less.NumberOut False
CheckGapL7350Speed35_55.NumberOut False
Decide 1775.NumberOut False
Decide 1664.NumberOut True
Decide 1642.NumberOut True
CheckGapL7650Speed75less.NumberOut True
CheckGapL6950Speed10t.NumberOut False
Decide 544.NumberOut True
CheckGapL7500Speed10t.NumberOut True
checkvehicletype7050.NumberOut True
Decide 550.NumberOut False
Decide 1783.NumberOut True
Decide 1764.NumberOut False
Decide 1651.NumberOut True
CheckGapL7700Speed55_75less.NumberOut True
CheckGapL7250Speed55_75less.NumberOut False
CheckGapL6450Speed55_75less.NumberOut True
Decide 1695.NumberOut True
CheckGapL7650Speed35_55less.NumberOut True
CheckGapL7250Speed75less.NumberOut False
CheckGapL7150Speed75.NumberOut True
CheckGapL6450Speed55_75t.NumberOut True
SpeedatRStation7250
checkvehicletype7750.NumberOut True
Decide 1682.NumberOut True
Decide 1643.NumberOut False
CheckGapL7650Speed10lesst.NumberOut True
Decide 1632.NumberOut False
CheckGapL7850Speed10_35t.NumberOut False
CheckGapL7150Speed35_55.NumberOut False
CheckGapL6700Speed10_35.NumberOut False
Decide 1719.NumberOut True
Decide 543.NumberOut False
Decide 1757.NumberOut False
CheckGapL7700Speed10_35t.NumberOut False
checkvehicletype7650.NumberOut False
Decide2VehicleTypeR6450.NumberOut False
CheckGapL7850Speed10_35lesst.NumberOut False
CheckGapL7550Speed35_55t.NumberOut False
CheckGapL7150Speed55_75.NumberOut True
Decide 1746.NumberOut False
CheckGapL7050Speed75lesst.NumberOut False
Decide 1471.NumberOut True
CheckGapL7600Speed10_35lesst.NumberOut False
CheckGapL6950Speed10_35t.NumberOut True
Decide 901.NumberOut True
Decide 1636.NumberOut False
Decide 1691.NumberOut False
Decide 1508.NumberOut True
CheckGapL6450Speed35_55less.NumberOut True
R.Vehicle Enters1615_1630.NumberOut

Decide 1649.NumberOut True
 Decide 1625.NumberOut False
 Decide 1480.NumberOut True
 CheckGapL7150Speed35_55lesst.NumberOut False
 Decide 954.NumberOut True
 Decide 1370.NumberOut True
 CheckGapL7500Speed10_35less.NumberOut True
 R.Vehicle Enters1700_1715.NumberOut
 CheckGapL6700Speed10less.NumberOut True
 Decide 1739.NumberOut False
 CheckGapL7900Speed10_35less.NumberOut False
 Decide 1407.NumberOut True
 Decide4VehicleTypeR6700.NumberOut True
 Decide 1695.NumberOut False
 CheckGapL7850Speed55_75lesst.NumberOut True
 CheckGapL7750Speed75less.NumberOut False
 CheckGapL7600Speed35_55less.NumberOut False
 Decide 982.NumberOut False
 CheckGapL7900Speed10.NumberOut True
 Decide 1438.NumberOut True
 Decide 1291.NumberOut True
 CheckGapL6450Speed10t.NumberOut True
 L.Vehicle Entry1430_1445.NumberOut
 CheckGapL7700Speed10less.NumberOut False
 CheckGapL7700Speed55_75less.NumberOut False
 CheckGapL7600Speed75less.NumberOut False
 CheckGapL7250Speed35_55less.NumberOut True
 CheckGapL7550Speed10.NumberOut False
 CheckGapL7150Speed10_35lesst.NumberOut True
 L.Vehicle Entry1100_1115.NumberOut
 Decide 1469.NumberOut True
 CheckGapL7500Speed55_75less.NumberOut False
 checkvehicletype6950.NumberOut True
 Decide3VehicleTypeR6700.NumberOut False
 L.Vehicle Entry1815_1830.NumberOut
 CheckGapL6450Speed55_75less.NumberOut False
 Decide 1688.NumberOut False
 CumuWaitTime
 CheckGapL7250Speed10.NumberOut True
 Dispose 9.NumberOut
 CheckGapL7900Speed35_55t.NumberOut False
 CheckGapL7150Speed10_35lesst.NumberOut False
 Decide 1279.NumberOut False
 Decide 660.NumberOut True
 CheckGapL7650Speed10.NumberOut False
 Decide 1780.NumberOut True
 L.Vehicle Entry1345_1400.NumberOut
 Decide 550.NumberOut True
 checkvehicletype7950.NumberOut False
 CheckGapL7900Speed35_55.NumberOut False
 CheckGapL7850Speed10_35t.NumberOut True
 CheckGapL7550Speed35_55.NumberOut True
 CheckGapL7650Speed55_75lesst.NumberOut False

CheckGapL6950Speed55_75.NumberOut False
 CheckGapL7150Speed75lesst.NumberOut False
 CheckGapL7900Speed55_75.NumberOut False
 CheckGapL7750Speed10.NumberOut False
 Decide 1380.NumberOut False
 CheckGapL7050Speed75lesst.NumberOut True
 checkvehicletype.NumberOut False
 Decide 1747.NumberOut True
 Decide 1725.NumberOut True
 Decide 1703.NumberOut True
 Decide 1701.NumberOut False
 CheckGapL6950Speed35_55lesst.NumberOut True
 CheckGapL7450Speed75t.NumberOut False
 Decide 658.NumberOut True
 CheckGapL7700Speed35_55.NumberOut False
 CheckGapL7350Speed75lesst.NumberOut False
 CheckGapL7150Speed55_75t.NumberOut False
 Decide 1778.NumberOut True
 Decide 1756.NumberOut True
 Decide 1734.NumberOut True
 R.Vehicle Enters645_700.NumberOut
 Decide 1646.NumberOut True
 Decide 1624.NumberOut True
 CheckGapL7050Speed10_35.NumberOut False
 Decide 548.NumberOut True
 Decide 1787.NumberOut True
 Decide 1760.NumberOut False
 CheckGapL7850Speed35_55lesst.NumberOut False
 CheckGapL7750Speed75.NumberOut True
 Decide 1373.NumberOut False
 CheckGapL6700Speed55_75less.NumberOut False
 SpeedatRStation7850
 Decide 1655.NumberOut True
 Decide 1633.NumberOut True
 CheckGapL7850Speed10.NumberOut False
 CheckGapL7700Speed55_75.NumberOut False
 Decide 1699.NumberOut True
 Decide 660.NumberOut False
 CheckGapL7600Speed35_55lesst.NumberOut False
 L.Vehicle Entry1830_1845.NumberOut
 CheckGapL7450Speed10_35less.NumberOut False
 CheckGapL6450Speed35_55t.NumberOut True
 L.Vehicle Entry1500_1515.NumberOut
 checkvehicletype7600.NumberOut True
 CheckGapL7150Speed35_55less.NumberOut False
 L.Vehicle Entry800_815.NumberOut
 CheckGapL7150Speed35_55.NumberOut True
 Decide 1753.NumberOut False
 CheckGapL7850Speed75t.NumberOut False
 Warmup Entry.NumberOut
 CheckGapL7050Speed10t.NumberOut True
 CheckGapL7050Speed10less.NumberOut True
 CheckGapL7050Speed55_75less.NumberOut False

Decide 1742.NumberOut False
CheckGapL7750Speed35_55t.NumberOut True
CheckGapL7550Speed75lesst.NumberOut True
CheckGapL7500Speed55_75.NumberOut False
CheckGapL7050Speed75.NumberOut True
Decide 1344.NumberOut False
CheckGapL7500Speed10lesst.NumberOut True
CheckGapL7350Speed55_75less.NumberOut True
CheckGapL7900Speed10_35.NumberOut True
Decide 1469.NumberOut False
CheckGapL7450Speed75less.NumberOut True
CheckGapL6700Speed10_35less.NumberOut True
R.Vehicle Enters830_845.NumberOut
L.Vehicle Entry1745_1800.NumberOut
CheckGapL7650Speed55_75t.NumberOut False
CheckGapL7600Speed10t.NumberOut True
CheckGapL7750Speed55_75t.NumberOut True
CheckGapL6950Speed75lesst.NumberOut False
CheckGapL7650Speed10_35.NumberOut True
Decide 1374.NumberOut True
CheckGapL7500Speed55_75t.NumberOut False
CheckGapL7050Speed10_35t.NumberOut False
CheckGapL6950Speed10t.NumberOut True
Decide 1735.NumberOut False
CheckGapL7150Speed10t.NumberOut True
CheckGapL7700Speed35_55less.NumberOut True
CheckGapL7550Speed35_55t.NumberOut True
CheckGapL6450Speed75.NumberOut False
Decide 1724.NumberOut False
Decide 1680.NumberOut False
CheckGapL7250Speed75lesst.NumberOut False
CheckGapL7350Speed10t.NumberOut False
CheckGapL7250Speed75less.NumberOut True
CheckGapL7700Speed10t.NumberOut True
CheckGapL7500Speed75t.NumberOut False
Decide 1282.NumberOut False
checkvehicletype7350.NumberOut True
Decide 1794.NumberOut False
CheckGapL7850Speed10_35lesst.NumberOut True
CheckGapL7850Speed55_75.NumberOut True
CheckGapL7750Speed35_55lesst.NumberOut True
CheckGapL7550Speed55_75t.NumberOut True
Decide 1282.NumberOut True
Decide 901.NumberOut False
L.Vehicle Entry745_800.NumberOut
CheckGapL7850Speed10.NumberOut True
Decide 1783.NumberOut False
Decide 1728.NumberOut False
CheckGapL7700Speed10_35less.NumberOut False
CheckGapL7550Speed35_55lesst.NumberOut True
CheckGapL7450Speed75lesst.NumberOut False
CheckGapL7750Speed35_55lesst.NumberOut False
CheckGapL7250Speed10t.NumberOut True

CheckGapL7150Speed10_35less.NumberOut True
CheckGapL6700Speed10_35.NumberOut True
Decide 1717.NumberOut False
CheckGapL7800Speed10_35.NumberOut False
CheckGapL7650Speed55_75lesst.NumberOut True
Decide 1662.NumberOut False
CheckGapL7900Speed75lesst.NumberOut True
checkvehicletype7500.NumberOut False
CheckGapL7050Speed10less.NumberOut False
CheckGapL7050Speed75less.NumberOut True
Decide 1787.NumberOut False
CheckGapL7800Speed10t.NumberOut True
CheckGapL7500Speed10_35.NumberOut True
Decide 1762.NumberOut True
Decide 1740.NumberOut True
CheckGapL7350Speed55_75t.NumberOut True
Decide 1776.NumberOut False
L.Vehicle Entry1900_1915.NumberOut
CheckGapL7750Speed10t.NumberOut False
CheckGapL7550Speed10_35t.NumberOut False
CheckGapL7250Speed10_35.NumberOut True
CheckGapL7150Speed10.NumberOut True
Decide 1793.NumberOut True
Decide 1771.NumberOut True
Decide 1666.NumberOut False
CheckGapL7900Speed75t.NumberOut False
CheckGapL7350Speed10t.NumberOut True
R.Vehicle Enters1130_1145.NumberOut
Decide 1661.NumberOut True
CheckGapL7600Speed10_35.NumberOut False
Decide 541.NumberOut True
Decide 1655.NumberOut False
CheckGapL7500Speed75lesst.NumberOut False
CheckGapL7250Speed35_55t.NumberOut False
CheckGapL7800Speed10less.NumberOut True
Decide 1670.NumberOut True
CheckGapL7700Speed55_75.NumberOut True
CheckGapL6950Speed35_55.NumberOut False
R.Vehicle Enters1515_1530.NumberOut
Decide 1692.NumberOut True
CheckGapL7900Speed10t.NumberOut True
Decide 1769.NumberOut False
CheckGapL7750Speed10_35lesst.NumberOut False
Decide 1729.NumberOut True
Decide 1707.NumberOut True
Decide 1534.NumberOut False
CheckGapL7450Speed55_75.NumberOut True
Decide 1758.NumberOut False
CheckGapL7500Speed10_35lesst.NumberOut False
CheckGapL6450Speed35_55lesst.NumberOut False
SpeedatRStation7350
CheckGapL6700Speed75less.NumberOut True
checkvehicletype.NumberOut True

Decide 1738.NumberOut True
Decide 1716.NumberOut True
Decide 1648.NumberOut False
Decide 1628.NumberOut True
CheckGapL7750Speed55_75less.NumberOut True
CheckGapL7050Speed35_55lesst.NumberOut False
R.Vehicle Enters1045_1100.NumberOut
Decide 1637.NumberOut False
CheckGapL7600Speed10less.NumberOut True
Decide 1769.NumberOut True
CheckGapL7550Speed10less.NumberOut False
CheckGapL7450Speed75less.NumberOut False
Decide 548.NumberOut False
Decide 1637.NumberOut True
Decide 539.NumberOut True
CheckGapL6700Speed75.NumberOut True
Dispose 10.NumberOut
CheckGapL7900Speed55_75lesst.NumberOut True
CheckGapL7650Speed10_35t.NumberOut True
CheckGapL7500Speed75.NumberOut False
Decide 1668.NumberOut True
Decide 1696.NumberOut False
CheckGapL7900Speed10_35t.NumberOut False
CheckGapL7850Speed35_55less.NumberOut False
CheckGapL7150Speed75t.NumberOut False
CheckGapL7750Speed35_55t.NumberOut False
Decide 1406.NumberOut False
CheckGapL7550Speed75lesst.NumberOut False
Decide 982.NumberOut True
CheckGapL7050Speed75.NumberOut False
Decide 928.NumberOut False
checkvehicletype6700.NumberOut False
CheckGapL7650Speed75.NumberOut True
CheckGapL7800Speed55_75less.NumberOut True
CheckGapL7600Speed35_55t.NumberOut False
checkvehicletype7800.NumberOut False
CheckGapL7700Speed75.NumberOut True
Decide 1008.NumberOut False
Decide 873.NumberOut False
CheckGapL6450Speed10_35lesst.NumberOut False
L.Vehicle Entry1245_1300.NumberOut
DecideVehicleTypeRight0.NumberOut False
CheckGapL7500Speed35_55.NumberOut False
CheckGapL7450Speed10_35t.NumberOut True
CheckGapL7250Speed10_35less.NumberOut False
CheckGapL7600Speed75.NumberOut False
SpeedatRStation7500
Decide 1465.NumberOut False
CheckGapL7550Speed10_35less.NumberOut True
CheckGapL7050Speed10_35lesst.NumberOut False
CheckGapL7050Speed55_75.NumberOut True
Decide 1689.NumberOut False
CheckGapL7800Speed10t.NumberOut False

Decide 1466.NumberOut True
CheckGapL7150Speed75.NumberOut False
R.Vehicle Enters1115_1130.NumberOut
Decide 927.NumberOut True
R.Vehicle Enters1530_1545.NumberOut
Decide 1731.NumberOut False
CheckGapL7600Speed75lesst.NumberOut False
R.Vehicle Enters1200_1215.NumberOut
Decide 1720.NumberOut False
CheckGapL7900Speed10less.NumberOut False
CheckGapL7250Speed10_35t.NumberOut True
CheckGapL6950Speed55_75lesst.NumberOut False
CheckGapL7850Speed55_75.NumberOut False
CheckGapL7700Speed75.NumberOut False
Decide 1277.NumberOut True
R.Vehicle Enters1915_1930.NumberOut
CheckGapL7550Speed55_75lesst.NumberOut False
Decide 1790.NumberOut False
CheckGapL7800Speed75lesst.NumberOut False
CheckGapL7600Speed10_35less.NumberOut True
CheckGapL7250Speed75.NumberOut False
CheckGapL6450Speed75less.NumberOut False
R.Vehicle Enters800_815.NumberOut
Decide 747.NumberOut True
Decide 1713.NumberOut False
CheckGapL6700Speed75t.NumberOut True
R.Vehicle Enters1445_1500.NumberOut
L.Vehicle Entry1315_1330.NumberOut
checkvehicletype7900.NumberOut True
CheckGapL6700Speed10lesst.NumberOut True
Decide 1702.NumberOut False
CheckGapL7800Speed75.NumberOut False
CheckGapL7350Speed35_55less.NumberOut True
CheckGapL7650Speed10_35lesst.NumberOut True
Decide 1744.NumberOut True
Decide 1722.NumberOut True
Decide 1700.NumberOut True
CheckGapL7750Speed10.NumberOut True
CheckGapL7050Speed10t.NumberOut False
CheckGapL6700Speed55_75t.NumberOut False
Decide 1772.NumberOut False
SpeedatRStation6950
CheckGapL7500Speed10_35less.NumberOut False
CheckGapL7350Speed35_55lesst.NumberOut True
CheckGapL7350Speed35_55t.NumberOut True
CheckGapL7800Speed10.NumberOut True
CheckGapL7500Speed35_55lesst.NumberOut False
Decide 1797.NumberOut True
Decide 1775.NumberOut True
Decide 1761.NumberOut False
Decide 1753.NumberOut True
Decide 1731.NumberOut True
Decide 1706.NumberOut False

Decide 1374.NumberOut False
 CheckGapL7450Speed55_75lesst.NumberOut True
 CheckGapL6450Speed10_35less.NumberOut False
 CheckGapL6450Speed10_35.NumberOut True
 Decide 1643.NumberOut True
 CheckGapL7650Speed75lesst.NumberOut False
 Decide 545.NumberOut True
 Decide 1651.NumberOut False
 CheckGapL7700Speed35_55.NumberOut True
 CheckGapL7250Speed55_75lesst.NumberOut True
 Decide 1784.NumberOut True
 CheckGapL7900Speed75.NumberOut False
 CheckGapL7350Speed10lesst.NumberOut True
 CheckGapL7250Speed75lesst.NumberOut True
 Decide 1652.NumberOut True
 Decide 1640.NumberOut False
 Decide 1630.NumberOut True
 Decide 1696.NumberOut True
 L.Vehicle Entry1645_1700.NumberOut
 Decide 551.NumberOut False
 Decide 1765.NumberOut False
 CheckGapL7850Speed75lesst.NumberOut False
 CheckGapL7350Speed55_75t.NumberOut False
 CheckGapL7150Speed35_55t.NumberOut True
 CheckGapL7050Speed10t.NumberOut True
 CheckGapL6950Speed75less.NumberOut False
 Decide 540.NumberOut False
 CheckGapL6700Speed10t.NumberOut False
 CheckGapL6700Speed10_35t.NumberOut True
 Decide 1754.NumberOut False
 CheckGapL6950Speed55_75less.NumberOut True
 R.Vehicle Enters745_800.NumberOut
 Decide 1644.NumberOut False
 L.Vehicle Entry930_945.NumberOut
 checkvehicletype7650.NumberOut True
 R.Vehicle Enters1930_1945.NumberOut
 Decide 1633.NumberOut False
 CheckGapL7150Speed55_75t.NumberOut True
 checkvehicletype7150.NumberOut False
 CheckGapL6450Speed75t.NumberOut False
 CheckGapL7700Speed75lesst.NumberOut False
 Decide 544.NumberOut False
 R.Vehicle Enters1600_1615.NumberOut
 CheckGapL6450Speed10_35.NumberOut False
 CheckGapL7450Speed10t.NumberOut True
 Decide 1747.NumberOut False
 Decide 658.NumberOut False
 Decide 1481.NumberOut True
 Decide 955.NumberOut True
 Decide 1736.NumberOut False
 CheckGapL7900Speed75lesst.NumberOut False
 Decide 1692.NumberOut False
 L.Vehicle Entry1330_1345.NumberOut

CheckGapL7900Speed10_35lesst.NumberOut True
 CheckGapL7800Speed35_55lesst.NumberOut True
 Decide 1371.NumberOut True
 CheckGapL6450Speed35_55t.NumberOut False
 Decide 1626.NumberOut False
 CheckGapL6950Speed35_55less.NumberOut True
 L.Vehicle Entry1000_1015.NumberOut
 CheckGapL7850Speed10lesst.NumberOut True
 CheckGapL7750Speed75lesst.NumberOut True
 CheckGapL7750Speed10_35.NumberOut False
 Decide 1380.NumberOut True
 R.Vehicle Enters1845_1900.NumberOut
 L.Vehicle Entry1715_1730.NumberOut
 CheckGapL7700Speed10lesst.NumberOut True
 CheckGapL7700Speed55_75lesst.NumberOut True
 CheckGapL6700Speed10_35less.NumberOut False
 Decide 1795.NumberOut False
 CheckGapL7650Speed35_55lesst.NumberOut False
 CheckGapL7550Speed10t.NumberOut True
 CheckGapL7850Speed55_75t.NumberOut False
 CheckGapL6450Speed10less.NumberOut True
 Decide 1729.NumberOut False
 CheckGapL7700Speed55_75t.NumberOut False
 CheckGapL7650Speed35_55less.NumberOut False
 CheckGapL6450Speed55_75lesst.NumberOut True
 CheckGapL7250Speed10_35t.NumberOut False
 Decide1VehicleTypeR6450.NumberOut True
 WarmupRunRight.NumberOut
 CheckGapL7750Speed35_55less.NumberOut True
 CheckGapL7750Speed55_75less.NumberOut False
 CheckGapL7550Speed75.NumberOut True
 CheckGapL7800Speed10_35.NumberOut True
 R.Vehicle Enters1945_2000.NumberOut
 CheckGapL7800Speed75less.NumberOut True
 CheckGapL7750Speed75lesst.NumberOut False
 CheckGapL7600Speed75.NumberOut True
 CheckGapL7550Speed55_75less.NumberOut False
 Decide 1788.NumberOut False
 CheckGapL7750Speed10less.NumberOut True
 CheckGapL7050Speed10_35less.NumberOut False
 CheckGapL7650Speed10t.NumberOut True
 CheckGapL6950Speed75.NumberOut False
 CheckGapL7850Speed35_55.NumberOut False
 Decide 1781.NumberOut True
 Decide 1667.NumberOut False
 CheckGapL7900Speed35_55t.NumberOut True
 CheckGapL6950Speed35_55t.NumberOut False
 Decide 551.NumberOut True
 Decide 1432.NumberOut False
 CheckGapL7550Speed75t.NumberOut False
 Decide 954.NumberOut False
 Decide 1790.NumberOut True
 CheckGapL7800Speed75less.NumberOut False

CheckGapL7650Speed10_35lesst.NumberOut False
 Decide 1680.NumberOut True
 CheckGapL7800Speed35_55less.NumberOut True
 CheckGapL7600Speed75less.NumberOut True
 checkvehicletype7450.NumberOut False
 CheckGapL7850Speed35_55.NumberOut True
 CheckGapL6700Speed55_75lesst.NumberOut False
 SpeedatRStation6450
 CheckGapL7900Speed55_75t.NumberOut True
 CheckGapL7750Speed10t.NumberOut True
 CheckGapL7250Speed10less.NumberOut False
 Decide 1535.NumberOut False
 CheckGapL7650Speed35_55.NumberOut False
 Decide 1748.NumberOut True
 Decide 1726.NumberOut True
 Decide 1704.NumberOut True
 Decide 1649.NumberOut False
 CheckGapL7750Speed10_35t.NumberOut False
 CheckGapL6700Speed10.NumberOut True
 Decide 659.NumberOut True
 CheckGapL7600Speed10_35t.NumberOut False
 CheckGapL7050Speed10_35t.NumberOut True
 Decide 1779.NumberOut True
 Decide 1757.NumberOut True
 Decide 1735.NumberOut True
 Decide 1713.NumberOut True
 L.Vehicle Entry1730_1745.NumberOut
 CheckGapL7650Speed55_75.NumberOut False
 Decide 1370.NumberOut False
 CheckGapL7450Speed35_55t.NumberOut False
 Decide 549.NumberOut False
 Decide 1625.NumberOut True
 Decide 549.NumberOut True
 L.Vehicle Entry1400_1415.NumberOut
 SpeedatRStation7450
 CheckGapL7900Speed55_75lesst.NumberOut False
 CheckGapL7850Speed10t.NumberOut True
 CheckGapL7700Speed55_75t.NumberOut True
 Decide 1788.NumberOut True
 Decide 1766.NumberOut True
 Decide 1656.NumberOut True
 Decide 1634.NumberOut True
 CheckGapL7450Speed10_35lesst.NumberOut True
 CheckGapL7450Speed35_55.NumberOut False
 CheckGapL7350Speed75.NumberOut False
 CheckGapL7050Speed55_75less.NumberOut True
 CheckGapL7650Speed10.NumberOut True
 CheckGapL6950Speed55_75less.NumberOut False
 CheckGapL7150Speed35_55lesst.NumberOut True
 CheckGapL7700Speed10.NumberOut True
 Decide 1407.NumberOut False
 Decide 1750.NumberOut False
 R.Vehicle Enters1015_1030.NumberOut

CheckGapL7600Speed55_75.NumberOut True
CheckGapL7450Speed55_75.NumberOut False
CheckGapL7050Speed55_75lesst.NumberOut True
Decide 874.NumberOut False
CheckGapL7750Speed10less.NumberOut False
CheckGapL7500Speed55_75t.NumberOut True
CheckGapL7450Speed35_55.NumberOut True

