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
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. Zwahlen 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
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-hour 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
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 straightforward 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 connecters 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 DYMOD 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.

Rouphail 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.

![Comparison of Average Inter-arrival Times versus Number of Vehicles per Hour for Daytime and Nighttime Driving Lane Data](image)

**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)

<table>
<thead>
<tr>
<th>Overall (N=767)</th>
<th>%?=Misses-Phantoms</th>
<th>Multiplication Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 0.02086</td>
<td>2.09% Phantoms</td>
<td>1.30%</td>
</tr>
<tr>
<td>26 0.0339</td>
<td>3.39% Misses</td>
<td>1.0130</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Driving Lane (N=456)</th>
<th>%?=Misses-Phantoms</th>
<th>Multiplication Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 0.01754</td>
<td>1.75% Phantoms</td>
<td>0.22%</td>
</tr>
<tr>
<td>9 0.01974</td>
<td>1.97% Misses</td>
<td>1.0022</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Passing Lane (N=321)</th>
<th>%?=Misses-Phantoms</th>
<th>Multiplication Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 0.0249</td>
<td>2.49% Phantoms</td>
<td>2.80%</td>
</tr>
<tr>
<td>17 0.0530</td>
<td>5.30% Misses</td>
<td>1.0280</td>
</tr>
</tbody>
</table>

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

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

* 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.)

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

* 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)

<table>
<thead>
<tr>
<th>Number of Vehicles per hour</th>
<th>Cumulative Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
</tr>
<tr>
<td>150</td>
<td>0.1</td>
</tr>
<tr>
<td>200</td>
<td>0.1</td>
</tr>
<tr>
<td>250</td>
<td>0.1</td>
</tr>
<tr>
<td>300</td>
<td>0.1</td>
</tr>
<tr>
<td>350</td>
<td>0.1</td>
</tr>
<tr>
<td>400</td>
<td>0.1</td>
</tr>
<tr>
<td>450</td>
<td>0.1</td>
</tr>
<tr>
<td>500</td>
<td>0.1</td>
</tr>
<tr>
<td>550</td>
<td>0.1</td>
</tr>
<tr>
<td>600</td>
<td>0.1</td>
</tr>
<tr>
<td>650</td>
<td>0.1</td>
</tr>
<tr>
<td>700</td>
<td>0.1</td>
</tr>
<tr>
<td>750</td>
<td>0.1</td>
</tr>
<tr>
<td>800</td>
<td>0.1</td>
</tr>
</tbody>
</table>

**Model**

<table>
<thead>
<tr>
<th>Model</th>
<th>y = 122.06/x + 0.4947</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>y = 183.83/x + 0.5566</td>
</tr>
<tr>
<td></td>
<td>y = 198.45/x + 0.8047</td>
</tr>
<tr>
<td></td>
<td>y = 469.05/x + 0.0585</td>
</tr>
<tr>
<td></td>
<td>y = 1254.82/x + 0.6556</td>
</tr>
<tr>
<td></td>
<td>y = 198.45/x + 0.8047</td>
</tr>
<tr>
<td></td>
<td>y = 2322.24/x + 0.0579</td>
</tr>
<tr>
<td></td>
<td>y = 3208.26/x + 0.2549</td>
</tr>
<tr>
<td></td>
<td>y = 4295.84/x - 0.1387</td>
</tr>
<tr>
<td></td>
<td>y = 5390.20/x + 0.0199</td>
</tr>
<tr>
<td></td>
<td>y = 7592.15 - 2.2824</td>
</tr>
<tr>
<td></td>
<td>y = 10848.2/x - 2.2824</td>
</tr>
<tr>
<td></td>
<td>y = 12050.56/x - 0.3884</td>
</tr>
<tr>
<td></td>
<td>y = 13495.82/x - 0.3967</td>
</tr>
<tr>
<td>Number of Vehicles per hour</td>
<td>Cumulative Percentage</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>50</td>
<td>0%  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</td>
</tr>
<tr>
<td>100</td>
<td>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</td>
</tr>
<tr>
<td>150</td>
<td>2%  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</td>
</tr>
<tr>
<td>200</td>
<td>3%  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</td>
</tr>
<tr>
<td>250</td>
<td>4%  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</td>
</tr>
<tr>
<td>300</td>
<td>5%  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</td>
</tr>
<tr>
<td>350</td>
<td>6%  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</td>
</tr>
<tr>
<td>400</td>
<td>7%  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</td>
</tr>
<tr>
<td>450</td>
<td>8%  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</td>
</tr>
<tr>
<td>500</td>
<td>9%  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</td>
</tr>
<tr>
<td>550</td>
<td>10% 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</td>
</tr>
<tr>
<td>600</td>
<td>11% 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</td>
</tr>
<tr>
<td>650</td>
<td>12% 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</td>
</tr>
<tr>
<td>700</td>
<td>13% 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</td>
</tr>
</tbody>
</table>

Model 1

\[ y = 12.22x + 0.4733 \]

Model 2

\[ y = 29.33x + 0.5211 \]

Model 3

\[ y = 57.77x + 0.5776 \]

Model 4

\[ y = 116.77x + 0.6024 \]

Model 5

\[ y = 443.98x + 0.1633 \]

Model 6

\[ y = 762.81x - 0.0794 \]

Model 7

\[ y = 1266.43x - 1.3412 \]

Model 8

\[ y = 1976.05x - 0.6918 \]

Model 9

\[ y = 3166.15x - 1.3412 \]

Model 10

\[ y = 4388.64x - 1.0848 \]

Model 11

\[ y = 6357.94x - 1.0673 \]

Model 12

\[ y = 9498.70x - 0.2287 \]

Model 13

\[ y = 10960.47x + 5.3074 \]

Model 14

\[ y = 11411.96x + 14.6193 \]

Model 15

\[ y = 11656.48x + 21.5875 \]

Model 16

\[ y = 12419.46x + 32.9255 \]
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 concise 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

<table>
<thead>
<tr>
<th>Number of vehicles per Hour</th>
<th>Driving Lane</th>
<th>Passing Lane</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average IAT for Fit Distribution (sec)</td>
<td>Average IAT based on Volume (sec)</td>
</tr>
<tr>
<td>50</td>
<td>68.62</td>
<td>72.00</td>
</tr>
<tr>
<td>100</td>
<td>34.56</td>
<td>36.00</td>
</tr>
<tr>
<td>150</td>
<td>22.93</td>
<td>24.00</td>
</tr>
<tr>
<td>200</td>
<td>17.27</td>
<td>18.00</td>
</tr>
<tr>
<td>250</td>
<td>13.88</td>
<td>14.40</td>
</tr>
<tr>
<td>300</td>
<td>11.61</td>
<td>12.00</td>
</tr>
<tr>
<td>350</td>
<td>10.00</td>
<td>10.29</td>
</tr>
<tr>
<td>400</td>
<td>8.78</td>
<td>9.00</td>
</tr>
<tr>
<td>450</td>
<td>7.84</td>
<td>8.00</td>
</tr>
<tr>
<td>500</td>
<td>7.09</td>
<td>7.20</td>
</tr>
<tr>
<td>550</td>
<td>6.47</td>
<td>6.55</td>
</tr>
<tr>
<td>600</td>
<td>5.95</td>
<td>6.00</td>
</tr>
<tr>
<td>650</td>
<td>5.52</td>
<td>5.54</td>
</tr>
<tr>
<td>700</td>
<td>5.15</td>
<td>5.14</td>
</tr>
<tr>
<td>750</td>
<td>4.82</td>
<td>4.80</td>
</tr>
<tr>
<td>800</td>
<td>4.54</td>
<td>4.50</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Number of Vehicles per hour</th>
<th>Cumulative Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
</tr>
<tr>
<td>150</td>
<td>0.10</td>
</tr>
<tr>
<td>200</td>
<td>0.10</td>
</tr>
<tr>
<td>250</td>
<td>0.10</td>
</tr>
<tr>
<td>300</td>
<td>0.10</td>
</tr>
<tr>
<td>350</td>
<td>0.10</td>
</tr>
<tr>
<td>400</td>
<td>0.10</td>
</tr>
<tr>
<td>450</td>
<td>0.10</td>
</tr>
<tr>
<td>500</td>
<td>0.10</td>
</tr>
<tr>
<td>550</td>
<td>0.10</td>
</tr>
<tr>
<td>600</td>
<td>0.10</td>
</tr>
<tr>
<td>650</td>
<td>0.10</td>
</tr>
<tr>
<td>700</td>
<td>0.10</td>
</tr>
<tr>
<td>750</td>
<td>0.10</td>
</tr>
<tr>
<td>800</td>
<td>0.10</td>
</tr>
<tr>
<td>Number of Vehicles per hour</td>
<td>Cumulative Percentage</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td></td>
<td>0%</td>
</tr>
<tr>
<td>50</td>
<td>0.10</td>
</tr>
<tr>
<td>100</td>
<td>0.10</td>
</tr>
<tr>
<td>150</td>
<td>0.10</td>
</tr>
<tr>
<td>200</td>
<td>0.10</td>
</tr>
<tr>
<td>250</td>
<td>0.10</td>
</tr>
<tr>
<td>300</td>
<td>0.10</td>
</tr>
<tr>
<td>350</td>
<td>0.10</td>
</tr>
<tr>
<td>400</td>
<td>0.10</td>
</tr>
<tr>
<td>450</td>
<td>0.10</td>
</tr>
<tr>
<td>500</td>
<td>0.10</td>
</tr>
<tr>
<td>550</td>
<td>0.10</td>
</tr>
<tr>
<td>600</td>
<td>0.10</td>
</tr>
<tr>
<td>650</td>
<td>0.10</td>
</tr>
<tr>
<td>700</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Table 9: Corrected IATs in Seconds based on Hyperbolic Formulae Developed for Passing Lane (Min 66 vehicles/hour, Max 666 vehicles/hour)
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 diving 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>
<thead>
<tr>
<th>Cumulative Percentage</th>
<th>IAT (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01%</td>
<td>0.10</td>
</tr>
<tr>
<td>1%</td>
<td>0.67</td>
</tr>
<tr>
<td>2%</td>
<td>0.82</td>
</tr>
<tr>
<td>5%</td>
<td>1.09</td>
</tr>
<tr>
<td>10%</td>
<td>1.39</td>
</tr>
<tr>
<td>20%</td>
<td>1.90</td>
</tr>
<tr>
<td>30%</td>
<td>2.45</td>
</tr>
<tr>
<td>40%</td>
<td>3.11</td>
</tr>
<tr>
<td>50%</td>
<td>3.90</td>
</tr>
<tr>
<td>60%</td>
<td>4.85</td>
</tr>
<tr>
<td>70%</td>
<td>6.01</td>
</tr>
<tr>
<td>80%</td>
<td>7.74</td>
</tr>
<tr>
<td>90%</td>
<td>10.70</td>
</tr>
<tr>
<td>95%</td>
<td>13.26</td>
</tr>
<tr>
<td>98%</td>
<td>16.87</td>
</tr>
<tr>
<td>99%</td>
<td>19.25</td>
</tr>
<tr>
<td>100%</td>
<td>25.68</td>
</tr>
</tbody>
</table>

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).

\[ f(t) = \lambda \times e^{-\lambda t} \]  

(1)

where,

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

\( \lambda = 0.1922 \) 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)

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

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).

\[
 f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi \sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2}
\]

\(x = \text{IAT for which cumulative probability is investigated (} x \geq 0.1 \text{ second, the minimum } x \text{ value (IAT) was taken as 0.1 seconds)}\)
\[ \mu = 5.20 \text{ seconds} \ (\infty < \mu < \infty) \ \text{(average of the IATs for 15 minute time interval where no. of vphpl was 698)} \]

\[ \sigma = 3.88 \text{ seconds} \ (\sigma > 0) \ \text{(standard deviation of the IATs for 15 minute time interval where no. of vphpl was 698)} \]

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)**

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

\[
f(t) = \frac{2}{G(K)}[\gamma(t - a)]^K \frac{1}{\lambda} e^{-\frac{1}{\lambda}}
\]

where,

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

\( \lambda = \) parameter that is a function of the mean time headway and the two user specified parameters, \( K \) and \( \alpha \). \( \lambda = \frac{K}{t - \alpha} = 0.258 \), 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 \( \infty \) that affects the shape of the distribution \( K = \frac{\bar{t} - \alpha}{s} = 1.3777 \), 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

\( \alpha = 0.1 \) 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 \( \Delta t \) 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)

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

The comparison graphs for driving 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.
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
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
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.
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
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.

\[
\text{CONT}(0.0001, 0.1, 0.01, 0.71, 0.02, 0.88, 0.05, 1.16, 0.1, 1.54, 0.2, 2.20, 0.3, 2.88, 0.4, 3.6
8, 0.5, 4.68, 0.6, 5.93, 0.7, 7.46, 0.8, 9.56, 0.9, 13.25, 0.95, 16.91, 0.98, 20.93, 0.99, 23.57, 1, 30.23)
\]

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

<table>
<thead>
<tr>
<th>Vehicle Type</th>
<th>Driving Lane</th>
<th>Passing Lane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passenger Car</td>
<td>91%</td>
<td>9%</td>
</tr>
<tr>
<td>Truck</td>
<td>61%</td>
<td>39%</td>
</tr>
</tbody>
</table>

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

- N= 11870
- Average= 60 MPH
- Standard Deviation= 4.73 MPH
- Minimum= 44 MPH
- Maximum= 67 MPH

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

- N= 6987
- Average= 66 MPH
- Standard Deviation= 4.97 MPH
- Minimum= 51 MPH
- Maximum= 73 MPH
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>
<thead>
<tr>
<th>Table 15: Vehicle Speeds used for the Transporters in the ARENA Simulation Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Driving Lane</td>
</tr>
<tr>
<td>Passing Lane</td>
</tr>
</tbody>
</table>
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])

<table>
<thead>
<tr>
<th>Speed Difference (Speed on Adjacent Lane – Merging Vehicle) (ft/s)</th>
<th>Minimum Safe Spacing for Lane Changing (ft)</th>
<th>Passenger Car</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Lead Gap</td>
<td>Vehicle Length</td>
</tr>
<tr>
<td>Speed Difference &lt; -50 ft/s</td>
<td></td>
<td>200 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>-50 &lt; Speed Difference &lt;= -35 ft/s</td>
<td></td>
<td>160 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>-35 &lt; Speed Difference &lt;= -15 ft/s</td>
<td></td>
<td>100 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>-15 &lt; Speed Difference &lt;= 15 ft/s</td>
<td></td>
<td>20 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>15 &lt; Speed Difference &lt;= 35 ft/s</td>
<td></td>
<td>20 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>35 &lt; Speed Difference &lt;= 50 ft/s</td>
<td></td>
<td>20 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>50 &lt; Speed Difference</td>
<td></td>
<td>20 ft</td>
<td>20 ft</td>
</tr>
</tbody>
</table>

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])

<table>
<thead>
<tr>
<th>Speed</th>
<th>Required Gaps for Lane Changing (ft)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lead Gap</td>
<td>Vehicle Length</td>
<td>Lag Gap</td>
</tr>
<tr>
<td>Speed &lt;= 10 ft/s</td>
<td>40 ft</td>
<td>20 ft</td>
<td>40 ft</td>
</tr>
<tr>
<td>10 ft/s &lt; Speed &lt;= 35 ft/s</td>
<td>65 ft</td>
<td>20 ft</td>
<td>65 ft</td>
</tr>
<tr>
<td>35 ft/s &lt; Speed &lt;= 55 ft/s</td>
<td>100 ft</td>
<td>20 ft</td>
<td>100 ft</td>
</tr>
<tr>
<td>55 ft/s &lt; Speed &lt;= 75 ft/s</td>
<td>130 ft</td>
<td>20 ft</td>
<td>130 ft</td>
</tr>
<tr>
<td>75 ft/s &lt; Speed</td>
<td>165 ft</td>
<td>20 ft</td>
<td>165 ft</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Speed</th>
<th>Required Gaps for Lane Changing (ft)</th>
<th>Passenger Car</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lead Gap</td>
<td>Vehicle Length</td>
<td>Lag Gap</td>
</tr>
<tr>
<td>Speed &lt;= 10 ft/s</td>
<td>40 ft</td>
<td>20 ft</td>
<td>5 ft</td>
</tr>
<tr>
<td>10 ft/s &lt; Speed &lt;= 35 ft/s</td>
<td>65 ft</td>
<td>20 ft</td>
<td>10 ft</td>
</tr>
<tr>
<td>35 ft/s &lt; Speed &lt;= 55 ft/s</td>
<td>100 ft</td>
<td>20 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>55 ft/s &lt; Speed &lt;= 75 ft/s</td>
<td>130 ft</td>
<td>20 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>75 ft/s &lt; Speed</td>
<td>165 ft</td>
<td>20 ft</td>
<td>30 ft</td>
</tr>
</tbody>
</table>
Table 19: Required Space for Lane Changing when the Merging Vehicle Speed is less than the Desired Lane Speed

<table>
<thead>
<tr>
<th>Speed</th>
<th>Required Gaps for Lane Changing (ft)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lead Gap</td>
<td>Vehicle Length</td>
<td>Lag Gap</td>
<td>Total</td>
<td>Vehicle Length</td>
<td>Total</td>
<td></td>
</tr>
<tr>
<td>Speed &lt;= 10 ft/s</td>
<td>5 ft</td>
<td>20 ft</td>
<td>40 ft</td>
<td>65 ft</td>
<td>65 ft</td>
<td>110 ft</td>
<td></td>
</tr>
<tr>
<td>10 ft/s &lt; Speed &lt;= 35 ft/s</td>
<td>10 ft</td>
<td>20 ft</td>
<td>65 ft</td>
<td>95 ft</td>
<td>65 ft</td>
<td>140 ft</td>
<td></td>
</tr>
<tr>
<td>35 ft/s &lt; Speed &lt;= 55 ft/s</td>
<td>20 ft</td>
<td>20 ft</td>
<td>100 ft</td>
<td>140 ft</td>
<td>65 ft</td>
<td>185 ft</td>
<td></td>
</tr>
<tr>
<td>55 ft/s &lt; Speed &lt;= 75 ft/s</td>
<td>20 ft</td>
<td>20 ft</td>
<td>130 ft</td>
<td>170 ft</td>
<td>65 ft</td>
<td>215 ft</td>
<td></td>
</tr>
<tr>
<td>75 ft/s &lt; Speed</td>
<td>30 ft</td>
<td>20 ft</td>
<td>165 ft</td>
<td>215 ft</td>
<td>65 ft</td>
<td>260 ft</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Speed</th>
<th>Required Gaps for Lane Changing (ft)</th>
<th>Passenger Car</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lead Gap</td>
<td>Vehicle Length</td>
<td>Lag Gap</td>
</tr>
<tr>
<td>Speed &lt;= 10 ft/s</td>
<td>5 ft</td>
<td>20 ft</td>
<td>30 ft</td>
</tr>
<tr>
<td>10 ft/s &lt; Speed &lt;= 35 ft/s</td>
<td>10 ft</td>
<td>20 ft</td>
<td>45 ft</td>
</tr>
<tr>
<td>35 ft/s &lt; Speed &lt;= 55 ft/s</td>
<td>20 ft</td>
<td>20 ft</td>
<td>55 ft</td>
</tr>
<tr>
<td>55 ft/s &lt; Speed &lt;= 75 ft/s</td>
<td>20 ft</td>
<td>20 ft</td>
<td>65 ft</td>
</tr>
<tr>
<td>75 ft/s &lt; Speed</td>
<td>30 ft</td>
<td>20 ft</td>
<td>90 ft</td>
</tr>
</tbody>
</table>
Table 21: Required Space for Lane Changing when there are Less than 5 Vehicles waiting in the Queue at the Lane Closure Taper

<table>
<thead>
<tr>
<th>Speed</th>
<th>Required Gaps for Lane Changing (ft)</th>
<th>Passenger Car</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lead Gap</td>
<td>Vehicle Length</td>
<td>Lag Gap</td>
</tr>
<tr>
<td>Speed &lt;= 10 ft/s</td>
<td>5 ft</td>
<td>20 ft</td>
<td>20 ft</td>
</tr>
<tr>
<td>10 ft/s &lt; Speed &lt;= 35 ft/s</td>
<td>10 ft</td>
<td>20 ft</td>
<td>35 ft</td>
</tr>
<tr>
<td>35 ft/s &lt; Speed &lt;= 55 ft/s</td>
<td>20 ft</td>
<td>20 ft</td>
<td>45 ft</td>
</tr>
<tr>
<td>55 ft/s &lt; Speed &lt;= 75 ft/s</td>
<td>20 ft</td>
<td>20 ft</td>
<td>55 ft</td>
</tr>
<tr>
<td>75 ft/s &lt; Speed</td>
<td>30 ft</td>
<td>20 ft</td>
<td>80 ft</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Speed</th>
<th>Required Gaps for Lane Changing (ft)</th>
<th>Passenger Car</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lead Gap</td>
<td>Vehicle Length</td>
<td>Lag Gap</td>
</tr>
<tr>
<td>Speed &lt;= 10 ft/s</td>
<td>5 ft</td>
<td>20 ft</td>
<td>10 ft</td>
</tr>
<tr>
<td>10 ft/s &lt; Speed &lt;= 35 ft/s</td>
<td>10 ft</td>
<td>20 ft</td>
<td>25 ft</td>
</tr>
<tr>
<td>35 ft/s &lt; Speed &lt;= 55 ft/s</td>
<td>20 ft</td>
<td>20 ft</td>
<td>35 ft</td>
</tr>
<tr>
<td>55 ft/s &lt; Speed &lt;= 75 ft/s</td>
<td>20 ft</td>
<td>20 ft</td>
<td>45 ft</td>
</tr>
<tr>
<td>75 ft/s &lt; Speed</td>
<td>30 ft</td>
<td>20 ft</td>
<td>70 ft</td>
</tr>
</tbody>
</table>
Table 23: Required Space for Lane Changing when there are More than 10 Vehicles waiting in the Queue at the Lane Closure Taper

<table>
<thead>
<tr>
<th>Speed</th>
<th>Passenger Car</th>
<th>Truck</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed &lt;= 10 ft/s</td>
<td>5 ft 20 ft 5 ft 30 ft 65 ft 75 ft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 ft/s &lt; Speed &lt;= 35 ft/s</td>
<td>10 ft 20 ft 15 ft 45 ft 65 ft 90 ft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35 ft/s &lt; Speed &lt;= 55 ft/s</td>
<td>20 ft 20 ft 25 ft 65 ft 65 ft 110 ft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55 ft/s &lt; Speed &lt;= 75 ft/s</td>
<td>20 ft 20 ft 35 ft 75 ft 65 ft 120 ft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>75 ft/s &lt; Speed</td>
<td>30 ft 20 ft 60 ft 110 ft 65 ft 155 ft</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
Assumptions:

**DRIVING LANE**

- Truck Percentage = 39% on driving lane
- Assign entity (vehicle) types according to the truck and car percentages from the actual data.

**Transporter Car**:
- Initial Velocity = 95 ft/sec (free flow speed)
- Vehicle Length = 20 ft (4 ft added to the average passenger car length to prevent collision when the vehicles stopped)

**Passing Lane**

- Truck Percentage = 9% on passing lane
- Assign entity (vehicle) types according to the truck and car percentages from the actual data.

**Transporter Truck**:
- Initial Velocity = 95 ft/sec (free flow speed)
- Vehicle Length = 65 ft (5 ft added to the average truck length to prevent collision when the vehicles stopped)

**Assign Transporter Car or Transporter Truck according to the entity (vehicle) type**

Transporter Car and Transporter Truck have the same properties for both lanes.

Explanations:

- Inter-arrival Time (IAT) Distribution is dependent on the number of vehicles per hour per lane according to the time of the day.
- Assign entity (vehicle) types according to the truck and car percentages from the actual data.
- Decide entity (vehicle) type for the transporter assignment.

---

**Figure 23: Flowchart of the ARENA Traffic Simulation Model**
Assign different speeds for the transporters using Normal Distribution with mean 88 ft/sec and standard deviation 3.47 ft/sec

Assign different speeds for the transporters using Normal Distribution with mean 75 ft/sec and standard deviation 10.27 ft/sec

Assign different speeds for the vehicles using Normal Distribution with mean 97 ft/sec and standard deviation 3.67 ft/sec

Assign different speeds for the vehicles using Normal Distribution with mean 84 ft/sec and standard deviation 8.8 ft/sec

Assign different speed values for the entity-transporter pairs using a speed distribution derived from actual data collected

Entities (vehicles) travel through the mainline with free flow speed

Entities (vehicles) arrive to the first lane closure sign and new speeds assigned for the travel in the construction zone

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

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)
Entity truck-transporter pairs on the passing lane at the first “Left Lane Closed Ahead” sign check for the availability of space on the driving lane. If the specified gaps are available on the driving lane, entity truck-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)
Assumptions

The same gap acceptance and lane changing procedure continues until the beginning of the left lane closure taper.

Entity-transporter pairs on the passing lane continue to check for the available gap for merging to the driving lane. They check for the gap at the 2nd merge sign; 100 ft, 200 ft, 300 ft, 400 ft after the 2nd merge sign; at the 3rd merge sign: 50 ft, 100 ft, 150 ft, 200 ft, 250 ft, 300 ft, 350 ft, 400 ft, and 450 ft after the 3rd merge sign to approximate driver behavior.

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

Entities arrive to the station at the beginning of the left lane closure taper

Explanations:

Entities arrive to the station at the taper. According to the number of vehicles waiting at the taper, entities check for different gap availability for lane changing.

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

Assumptions

Explanations:

At the Beginning of Left Lane Closure Taper

- **Driving Lane Speed > 75 ft/s**
  - **YES**
  - **NO**

- **Lead Gap in Driving Lane >= 140 ft**
  - **YES**
  - **NO**

- **5 ft/s <= Driving Lane Speed > 55 ft/s**
  - **NO**

- **Lag Gap in Driving Lane >= 185 ft**
  - **YES**
  - **NO**

- **5 ft/s >= Driving Lane Speed **
  - **NO**

- **Lag Gap in Driving Lane >= 150 ft**
  - **YES**
  - **NO**

- **35 ft/s <= Driving Lane Speed > 35 ft/s**
  - **NO**

- **Lag Gap in Driving Lane >= 140 ft**
  - **YES**
  - **NO**

- **10 ft/s <= Driving Lane Speed > 10 ft/s**
  - **NO**

- **Lag Gap in Driving Lane >= 120 ft**
  - **YES**
  - **NO**

- **5 ft/s <= Driving Lane Speed > 10 ft/s**
  - **NO**

- **Lag Gap in Driving Lane >= 75 ft**
  - **YES**
  - **NO**

- **Lead Gap in Driving Lane >= 55 ft**
  - **YES**
  - **NO**

- **NO**

Hold until finding available Gap

PASSING LANE

- **Driving Lane Speed > 75 ft/s**
  - **YES**
  - **NO**

- **Lead Gap in Driving Lane >= 140 ft**
  - **YES**
  - **NO**

- **5 ft/s <= Driving Lane Speed > 55 ft/s**
  - **NO**

- **Lead Gap in Driving Lane >= 105 ft**
  - **YES**
  - **NO**

- **5 ft/s >= Driving Lane Speed **
  - **NO**

- **Lead Gap in Driving Lane >= 95 ft**
  - **YES**
  - **NO**

- **35 ft/s <= Driving Lane Speed > 35 ft/s**
  - **NO**

- **Lead Gap in Driving Lane >= 95 ft**
  - **YES**
  - **NO**

- **10 ft/s <= Driving Lane Speed > 10 ft/s**
  - **NO**

- **Lead Gap in Driving Lane >= 75 ft**
  - **YES**
  - **NO**

- **5 ft/s <= Driving Lane Speed > 10 ft/s**
  - **NO**

- **Lead Gap in Driving Lane >= 55 ft**
  - **YES**
  - **NO**

- **NO**

Merge to Next Station on Driving Lane

If less than 3 entities with transporters are waiting at the taper, check for the speeds and when the specified gap is available in the driving lane, merge to the driving lane; else wait in the queue until finding available gap for lane changing.

Figure 23: Flowchart of the ARENA Traffic Simulation Model (continued)
If less than 5 entities with transporters are waiting at the taper, check for the speeds and when the specified gap is available in the driving lane, merge to the driving lane; else wait in the queue until finding available gap for lane changing.

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

- **At the Beginning of Left Lane Closure Taper**
  - **Driving Lane Speed > 75 ft/s**
  - **Lead Gap in Driving Lane >= 120 ft**

- **Driving Lane Speed > 75 ft/s**
  - **Lag Gap in Driving Lane >= 165 ft**

- **Driving Lane Speed > 55 ft/s**
  - **Lag Gap in Driving Lane >= 130 ft**

- **Driving Lane Speed > 35 ft/s**
  - **Lag Gap in Driving Lane >= 120 ft**
  - **Lag Gap in Driving Lane >= 100 ft**

- **Lag Gap in Driving Lane >= 80 ft**

- **Hold until finding available Gap**

**Explanations:**

If less than 10 entities with transporters are waiting at the taper, check for the speeds and when the specified gap is available in the driving lane, merge to the driving lane; else wait in the queue until finding available gap for lane changing.

---

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

Explanations:

If more than 10 entities with transporters are waiting at the taper, check for the speeds and when the specified gap is available in the driving lane, merge to the driving lane; else wait in the queue until finding available gap for lane changing.

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

After the lane closure taper ends all entities on the driving lanes shift to the shoulder, and continue in the one-lane section of the construction zone.

Explanations:

Entities arrive to the last location in the simulation.

Entity statistics are recorded and entities are disposed.

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.

<table>
<thead>
<tr>
<th>Node Number</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3000.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>2300.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**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.
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

<table>
<thead>
<tr>
<th>Link #</th>
<th>A Node</th>
<th>B Node</th>
<th>Lanes</th>
<th>Capacity (vphpl)</th>
<th>Length (Miles)</th>
<th>Freeflow Speed (mph)</th>
<th>Jam Density (V/mi/L)</th>
<th>I or O</th>
<th>Type</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2100</td>
<td>0.1326</td>
<td>65</td>
<td>220</td>
<td>I</td>
<td>M</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2100</td>
<td>0.4356</td>
<td>55</td>
<td>220</td>
<td>I</td>
<td>WZ</td>
<td>0</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Time</th>
<th>Number of Vehicles per Hour</th>
<th>Hourly Demand Factors (Number of Vehicles per hour / Number of Vehicles per Day)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:00</td>
<td>264</td>
<td>1.40%</td>
</tr>
<tr>
<td>1:00</td>
<td>207</td>
<td>1.10%</td>
</tr>
<tr>
<td>2:00</td>
<td>244</td>
<td>1.29%</td>
</tr>
<tr>
<td>3:00</td>
<td>210</td>
<td>1.12%</td>
</tr>
<tr>
<td>4:00</td>
<td>276</td>
<td>1.47%</td>
</tr>
<tr>
<td>5:00</td>
<td>425</td>
<td>2.26%</td>
</tr>
<tr>
<td>6:00</td>
<td>881</td>
<td>4.68%</td>
</tr>
<tr>
<td>7:00</td>
<td>1143</td>
<td>6.07%</td>
</tr>
<tr>
<td>8:00</td>
<td>1027</td>
<td>5.45%</td>
</tr>
<tr>
<td>9:00</td>
<td>956</td>
<td>5.08%</td>
</tr>
<tr>
<td>10:00</td>
<td>1101</td>
<td>5.85%</td>
</tr>
<tr>
<td>11:00</td>
<td>1117</td>
<td>5.94%</td>
</tr>
<tr>
<td>12:00</td>
<td>1153</td>
<td>6.12%</td>
</tr>
<tr>
<td>13:00</td>
<td>1092</td>
<td>5.80%</td>
</tr>
<tr>
<td>14:00</td>
<td>1262</td>
<td>6.70%</td>
</tr>
<tr>
<td>15:00</td>
<td>1249</td>
<td>6.63%</td>
</tr>
<tr>
<td>16:00</td>
<td>1233</td>
<td>6.55%</td>
</tr>
<tr>
<td>17:00</td>
<td>1121</td>
<td>5.95%</td>
</tr>
<tr>
<td>18:00</td>
<td>1008</td>
<td>5.35%</td>
</tr>
<tr>
<td>19:00</td>
<td>821</td>
<td>4.36%</td>
</tr>
<tr>
<td>20:00</td>
<td>634</td>
<td>3.37%</td>
</tr>
<tr>
<td>21:00</td>
<td>602</td>
<td>3.20%</td>
</tr>
<tr>
<td>22:00</td>
<td>441</td>
<td>2.34%</td>
</tr>
<tr>
<td>23:00</td>
<td>361</td>
<td>1.92%</td>
</tr>
<tr>
<td>total</td>
<td>18827</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
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.
Table: Daily Traffic Counts for the Links

<table>
<thead>
<tr>
<th>Link</th>
<th>I or O</th>
<th>AADT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>18827</td>
</tr>
<tr>
<td>2</td>
<td>I</td>
<td>18827</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Title</th>
<th>Queue-Inbound</th>
<th>Delay-Inbound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Weekly Max (mi)</td>
<td>Weekly Total (mi)</td>
</tr>
<tr>
<td>Phase1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Phase1-Work1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th></th>
<th>Driving Lane</th>
<th></th>
<th>Passing Lane</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual Data</td>
<td>Simulation Output</td>
<td>Percent Difference (Simulation Output/Actual Data)</td>
<td>Actual Data</td>
</tr>
<tr>
<td>Car</td>
<td>7222</td>
<td>7177</td>
<td>99.377%</td>
<td>6358</td>
</tr>
<tr>
<td>Truck</td>
<td>4617</td>
<td>4612</td>
<td>99.892%</td>
<td>628</td>
</tr>
<tr>
<td>Total</td>
<td>11840</td>
<td>11789</td>
<td>99.569%</td>
<td>6987</td>
</tr>
</tbody>
</table>

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.
Comparison of Actual Number of Vehicles on Both Lanes and Simulated Number of Vehicles on Both Lanes as a Function of Time

\[ y = 0.9517x + 17.699 \]

\[ R^2 = 0.9978 \]

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

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

\[ y = 1.0036x - 3.912 \]

\[ R^2 = 0.9992 \]

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

\[ y = 1.0917x - 2.4297 \]

\[ R^2 = 0.9985 \]

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

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

\[ y = 1.0065x - 1.4823 \]

\[ R^2 = 0.9982 \]

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

\[ y = 0.8936x + 12.938 \]

\[ R^2 = 0.9932 \]

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

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

\[ y = 0.8936x + 12.938 \]

\[ R^2 = 0.9932 \]

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.

Average Speed on Driving Lane = 3000 ft / 39.59 sec = 75.77 ft/sec

Average Speed on Passing Lane = 3000 ft / 35.26 sec = 85.08 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.
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
Comparison of Average Wait Times per Vehicle per 15 Minute Intervals and Number of Vehicles per Hour 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

Comparison of Average Queue Lengths per 15 Minute Intervals and Number of Vehicles per Hour 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
### 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 Minimum Transfer Times per Vehicle per 15 Minute Interval**
  - Minimum Transfer Time: Minimum = 29.46 sec, Maximum = 30.79 sec, Average = 30.03 sec, Standard Deviation = 0.32 sec
  - N = 96

- **Number of Vehicles Per Hour**
  - Minimum = 0, Maximum = 700

### 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

- **Comparison of Average Wait Times per Vehicle per 15 Minute Interval**
  - Average Wait Time: Minimum = 0.013 sec, Maximum = 0.069 sec, Average = 0.045 sec, Standard Deviation = 0.016 sec
  - N = 96
Comparison of Maximum Wait Times per Vehicle per 15 Minute Intervals and Number of Vehicles per Hour 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

Comparison of Minimum Wait Times per Vehicle per 15 Minute Intervals and Number of Vehicles per Hour 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
Comparison of Average Number of Vehicle Waiting in the Queue per 15 Minute Intervals and Number of Vehicles per Hour for Passing Lane

- Average Number in Queue
  - Average = 0.005
  - Standard Deviation = 0.003
  - Minimum = 0
  - Maximum = 0.012
  - N = 96

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

Comparison of Maximum Number of Vehicle Waiting in the Queue per 15 Minute Intervals and Number of Vehicles per Hour for Passing Lane

- Maximum Number in Queue
  - Average = 0.454
  - Standard Deviation = 0.275
  - Minimum = 0
  - Maximum = 0.806
  - N = 96

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


[29] Supplementary User’s Manual for MD-QuickZone, MD State Highway Administration, Office of Traffic and Safety, August 2002.


Appendix A. Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime Driving/Passing Lane
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>7:00-7:15</td>
<td>46</td>
<td>20.90</td>
<td>19.29</td>
<td>1.41</td>
<td>74.83</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>6:45-7:00</td>
<td>47</td>
<td>17.81</td>
<td>14.16</td>
<td>1.09</td>
<td>65.72</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>7:15-7:30</td>
<td>59</td>
<td>15.31</td>
<td>14.99</td>
<td>1.18</td>
<td>77.47</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>7:45-8:00</td>
<td>65</td>
<td>13.72</td>
<td>12.55</td>
<td>1.05</td>
<td>53.01</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>7:30-7:45</td>
<td>67</td>
<td>13.45</td>
<td>11.78</td>
<td>1.67</td>
<td>51.37</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>8:00-8:15</td>
<td>67</td>
<td>13.56</td>
<td>11.05</td>
<td>0.96</td>
<td>54.74</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>6:45-7:00</td>
<td>72</td>
<td>12.03</td>
<td>8.60</td>
<td>0.65</td>
<td>47.06</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>7:00-7:15</td>
<td>78</td>
<td>11.78</td>
<td>11.77</td>
<td>0.90</td>
<td>58.43</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>8:15-8:30</td>
<td>83</td>
<td>10.78</td>
<td>10.07</td>
<td>0.97</td>
<td>52.45</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>7:15-7:30</td>
<td>86</td>
<td>10.72</td>
<td>10.56</td>
<td>0.92</td>
<td>56.15</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>8:30-8:45</td>
<td>88</td>
<td>9.99</td>
<td>9.49</td>
<td>0.75</td>
<td>47.34</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>8:00-8:15</td>
<td>104</td>
<td>8.72</td>
<td>6.95</td>
<td>0.69</td>
<td>31.51</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>8:45-9:00</td>
<td>107</td>
<td>8.61</td>
<td>8.38</td>
<td>0.72</td>
<td>43.00</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>7:45-8:00</td>
<td>111</td>
<td>7.99</td>
<td>5.73</td>
<td>0.74</td>
<td>32.08</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>20:00-20:15</td>
<td>116</td>
<td>7.74</td>
<td>6.89</td>
<td>0.54</td>
<td>36.33</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>8:15-8:30</td>
<td>116</td>
<td>7.78</td>
<td>6.65</td>
<td>0.58</td>
<td>36.77</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>7:30-7:45</td>
<td>118</td>
<td>7.63</td>
<td>7.33</td>
<td>0.57</td>
<td>49.19</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>19:45-20:00</td>
<td>120</td>
<td>7.47</td>
<td>6.25</td>
<td>0.95</td>
<td>28.71</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>19:30-19:45</td>
<td>122</td>
<td>7.34</td>
<td>5.94</td>
<td>0.60</td>
<td>34.02</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>9:00-9.15</td>
<td>123</td>
<td>7.37</td>
<td>7.20</td>
<td>0.48</td>
<td>42.23</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>9:30-9.45</td>
<td>124</td>
<td>7.24</td>
<td>6.56</td>
<td>0.62</td>
<td>29.69</td>
</tr>
</tbody>
</table>
### Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime Driving Lane

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average Standard Deviation</td>
<td>Min Max</td>
<td>Average Standard Deviation</td>
<td>Min Max</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>19.15-19.30</td>
<td>127</td>
<td>7.12</td>
<td>5.76</td>
<td>0.58</td>
<td>40.51</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70</td>
<td>4.30</td>
<td>61</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>20.00-20:15</td>
<td>129</td>
<td>6.97</td>
<td>5.98</td>
<td>0.76</td>
<td>32.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69</td>
<td>5.57</td>
<td>56</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>9.30-9.45</td>
<td>129</td>
<td>6.93</td>
<td>5.24</td>
<td>0.63</td>
<td>21.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>73</td>
<td>3.54</td>
<td>63</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>19.15-19.30</td>
<td>130</td>
<td>6.97</td>
<td>6.28</td>
<td>0.54</td>
<td>30.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
<td>4.81</td>
<td>54</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>19.45-20.00</td>
<td>130</td>
<td>6.87</td>
<td>5.11</td>
<td>0.72</td>
<td>27.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68</td>
<td>4.12</td>
<td>59</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>9.15-9.30</td>
<td>132</td>
<td>6.79</td>
<td>5.84</td>
<td>0.69</td>
<td>27.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
<td>5.93</td>
<td>53</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>19.00-19.15</td>
<td>134</td>
<td>6.94</td>
<td>7.34</td>
<td>0.31</td>
<td>54.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70</td>
<td>4.33</td>
<td>59</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>18.15-18.30</td>
<td>136</td>
<td>6.41</td>
<td>5.63</td>
<td>0.34</td>
<td>33.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68</td>
<td>4.57</td>
<td>60</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>10.15-10.30</td>
<td>136</td>
<td>6.52</td>
<td>5.70</td>
<td>0.70</td>
<td>25.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69</td>
<td>4.49</td>
<td>57</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>19.00-19.15</td>
<td>139</td>
<td>6.43</td>
<td>5.83</td>
<td>0.69</td>
<td>36.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>63</td>
<td>6.66</td>
<td>50</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>18.45-19.00</td>
<td>139</td>
<td>6.25</td>
<td>5.12</td>
<td>0.59</td>
<td>24.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68</td>
<td>5.75</td>
<td>53</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>10.00-10.15</td>
<td>139</td>
<td>6.67</td>
<td>5.35</td>
<td>0.57</td>
<td>30.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>69</td>
<td>4.18</td>
<td>56</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>6:45-7:00</td>
<td>141</td>
<td>6.41</td>
<td>4.96</td>
<td>0.63</td>
<td>39.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>64</td>
<td>5.14</td>
<td>51</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>18.15-18.30</td>
<td>141</td>
<td>6.39</td>
<td>5.57</td>
<td>0.38</td>
<td>29.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>62</td>
<td>4.09</td>
<td>53</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>18.45-19.00</td>
<td>141</td>
<td>6.37</td>
<td>4.73</td>
<td>0.49</td>
<td>21.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>67</td>
<td>4.41</td>
<td>58</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>17.15-17.30</td>
<td>141</td>
<td>6.40</td>
<td>4.05</td>
<td>0.65</td>
<td>20.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>71</td>
<td>3.90</td>
<td>63</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>10.15-10.30</td>
<td>142</td>
<td>6.31</td>
<td>5.56</td>
<td>0.88</td>
<td>27.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>65</td>
<td>5.12</td>
<td>53</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>10.30-10.45</td>
<td>142</td>
<td>6.44</td>
<td>5.81</td>
<td>0.49</td>
<td>28.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>68</td>
<td>6.13</td>
<td>53</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>19.30-19.45</td>
<td>143</td>
<td>6.29</td>
<td>5.06</td>
<td>0.77</td>
<td>33.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72</td>
<td>5.92</td>
<td>57</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>18.30-18.45</td>
<td>144</td>
<td>6.28</td>
<td>5.69</td>
<td>0.55</td>
<td>45.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>61</td>
<td>5.84</td>
<td>46</td>
</tr>
</tbody>
</table>
# Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

## Driving Lane

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/21/2004</td>
<td>9:45-10:00</td>
<td>144</td>
<td>6.29 5.83 0.69 36.36 65</td>
<td>4.86 53 75 576 577</td>
<td></td>
</tr>
<tr>
<td>8/22/2004</td>
<td>10:45-11:00</td>
<td>145</td>
<td>6.19 5.22 0.67 25.90 69</td>
<td>6.13 57 81 580 581</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>8:45-9:00</td>
<td>146</td>
<td>6.16 5.29 0.76 25.19 59</td>
<td>5.78 45 70 584 585</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>9:15-9:30</td>
<td>146</td>
<td>6.17 5.69 0.49 44.22 62</td>
<td>6.94 47 77 584 585</td>
<td></td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13:30-13:45</td>
<td>146</td>
<td>6.15 5.17 0.76 28.37 66</td>
<td>5.30 53 74 584 585</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>10:00-10:15</td>
<td>148</td>
<td>6.13 5.12 0.58 26.71 58</td>
<td>5.69 43 71 592 593</td>
<td></td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19:15-19:30</td>
<td>148</td>
<td>6.02 5.77 0.33 44.18 68</td>
<td>5.73 53 78 592 593</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>9:45-10:00</td>
<td>149</td>
<td>6.03 4.25 0.81 23.52 60</td>
<td>4.52 49 70 596 597</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>11:15-11:30</td>
<td>149</td>
<td>6.05 5.08 0.60 28.57 58</td>
<td>5.08 45 70 596 597</td>
<td></td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13:45-14:00</td>
<td>149</td>
<td>6.09 5.17 0.58 32.83 65</td>
<td>5.41 56 80 596 597</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16:00-16:15</td>
<td>150</td>
<td>5.98 4.59 0.76 22.70 63</td>
<td>4.19 53 72 600 601</td>
<td></td>
</tr>
<tr>
<td>8/21/2004</td>
<td>8:45-9:00</td>
<td>151</td>
<td>5.94 6.11 0.49 39.45 66</td>
<td>5.20 54 76 604 605</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>9:00-9:15</td>
<td>152</td>
<td>5.94 4.68 0.67 28.90 66</td>
<td>4.14 52 70 608 609</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>17:45-18:00</td>
<td>152</td>
<td>5.93 5.06 0.62 28.45 63</td>
<td>6.90 46 74 608 609</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>7:45-8:00</td>
<td>153</td>
<td>5.88 4.80 0.53 37.13 61</td>
<td>5.17 52 74 612 613</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>12:00-12:15</td>
<td>153</td>
<td>5.87 5.12 0.57 33.31 62</td>
<td>5.38 49 74 612 613</td>
<td></td>
</tr>
<tr>
<td>8/21/2004</td>
<td>17:45-18:00</td>
<td>153</td>
<td>5.84 4.37 0.81 23.15 68</td>
<td>3.67 60 80 612 613</td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>7:00-7:15</td>
<td>155</td>
<td>5.80 4.37 0.66 22.19 63</td>
<td>6.37 45 76 620 621</td>
<td></td>
</tr>
<tr>
<td>8/22/2004</td>
<td>14:00-14:15</td>
<td>155</td>
<td>5.82 5.11 0.51 33.72 66</td>
<td>6.37 50 79 620 621</td>
<td></td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19:30-19:45</td>
<td>155</td>
<td>5.85 4.57 0.49 26.70 67</td>
<td>5.01 55 75 620 621</td>
<td></td>
</tr>
</tbody>
</table>
**Hourly Traffic Counts (vph), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime Driving Lane**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------------------------------------------</td>
<td>---------</td>
<td>------------------</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>10.45-11.00</td>
<td>156</td>
<td>5.74</td>
<td>5.09</td>
<td>0.89</td>
<td>25.78</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.00-11.15</td>
<td>156</td>
<td>5.74</td>
<td>4.72</td>
<td>0.49</td>
<td>26.02</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.15-11.30</td>
<td>156</td>
<td>5.81</td>
<td>5.81</td>
<td>0.61</td>
<td>35.65</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13.45-14.00</td>
<td>156</td>
<td>5.74</td>
<td>4.56</td>
<td>0.68</td>
<td>22.19</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>16.00-16.15</td>
<td>157</td>
<td>5.71</td>
<td>4.68</td>
<td>0.70</td>
<td>26.24</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.15-11.30</td>
<td>156</td>
<td>5.74</td>
<td>4.89</td>
<td>0.63</td>
<td>23.79</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13.00-13.15</td>
<td>159</td>
<td>5.65</td>
<td>3.90</td>
<td>0.47</td>
<td>21.50</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>18.00-18.15</td>
<td>159</td>
<td>5.63</td>
<td>4.00</td>
<td>0.47</td>
<td>23.18</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13.00-13.15</td>
<td>159</td>
<td>5.66</td>
<td>5.68</td>
<td>0.53</td>
<td>50.56</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13.45-14.00</td>
<td>161</td>
<td>5.55</td>
<td>4.12</td>
<td>0.67</td>
<td>22.39</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>14.30-14.45</td>
<td>161</td>
<td>5.66</td>
<td>4.39</td>
<td>0.78</td>
<td>28.44</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>16.45-17.00</td>
<td>161</td>
<td>5.55</td>
<td>4.76</td>
<td>0.67</td>
<td>24.98</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>17.00-17.15</td>
<td>161</td>
<td>5.60</td>
<td>4.68</td>
<td>0.66</td>
<td>34.05</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>9.30-9.45</td>
<td>163</td>
<td>5.50</td>
<td>4.40</td>
<td>0.49</td>
<td>21.88</td>
</tr>
</tbody>
</table>
### Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

#### Driving Lane (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>12.00-12.15</td>
<td>163</td>
<td>5.50</td>
<td>4.42</td>
<td>0.66</td>
<td>19.75</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>12.15-12.30</td>
<td>163</td>
<td>5.55</td>
<td>4.29</td>
<td>0.60</td>
<td>27.17</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>11.30-11.45</td>
<td>164</td>
<td>5.49</td>
<td>4.01</td>
<td>0.52</td>
<td>21.61</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>14.15-14.30</td>
<td>164</td>
<td>5.40</td>
<td>4.49</td>
<td>0.51</td>
<td>26.66</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>10.15-10.30</td>
<td>165</td>
<td>5.44</td>
<td>4.22</td>
<td>0.56</td>
<td>25.61</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>17.00-17.15</td>
<td>165</td>
<td>5.53</td>
<td>4.60</td>
<td>0.59</td>
<td>21.57</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>15.45-16.00</td>
<td>165</td>
<td>5.47</td>
<td>5.06</td>
<td>0.75</td>
<td>31.33</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>16.30-16.45</td>
<td>165</td>
<td>5.45</td>
<td>4.18</td>
<td>0.58</td>
<td>23.41</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.00-11.15</td>
<td>166</td>
<td>5.46</td>
<td>4.91</td>
<td>0.53</td>
<td>31.63</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>10.30-10.45</td>
<td>167</td>
<td>5.39</td>
<td>4.25</td>
<td>0.46</td>
<td>20.48</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>16.00-16.15</td>
<td>167</td>
<td>5.27</td>
<td>4.32</td>
<td>0.55</td>
<td>26.24</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>8.00-8.15</td>
<td>169</td>
<td>5.36</td>
<td>4.67</td>
<td>0.60</td>
<td>28.95</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>8.15-8.30</td>
<td>169</td>
<td>5.95</td>
<td>4.91</td>
<td>0.83</td>
<td>26.66</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>12.15-12.30</td>
<td>169</td>
<td>5.39</td>
<td>4.46</td>
<td>0.59</td>
<td>21.36</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>12.30-12.45</td>
<td>169</td>
<td>5.32</td>
<td>3.61</td>
<td>0.62</td>
<td>15.36</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>15.00-15.15</td>
<td>169</td>
<td>5.38</td>
<td>4.45</td>
<td>0.60</td>
<td>23.42</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>12.30-12.45</td>
<td>169</td>
<td>5.30</td>
<td>4.46</td>
<td>0.65</td>
<td>24.25</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>10.45-11.00</td>
<td>170</td>
<td>5.32</td>
<td>4.40</td>
<td>0.38</td>
<td>26.60</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>12.15-12.30</td>
<td>170</td>
<td>5.32</td>
<td>3.82</td>
<td>0.75</td>
<td>19.23</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>12.45-13.00</td>
<td>170</td>
<td>5.29</td>
<td>4.50</td>
<td>0.50</td>
<td>20.52</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>13.15-13.30</td>
<td>170</td>
<td>5.29</td>
<td>4.53</td>
<td>0.56</td>
<td>30.97</td>
</tr>
</tbody>
</table>
Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>14.00-14.15</td>
<td>170</td>
<td>5.29</td>
<td>4.06</td>
<td>0.63</td>
<td>26.13</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13.15-13.30</td>
<td>170</td>
<td>5.25</td>
<td>4.66</td>
<td>0.49</td>
<td>30.23</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>14.45-15.00</td>
<td>170</td>
<td>5.29</td>
<td>4.22</td>
<td>0.30</td>
<td>24.48</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>15.45-16.00</td>
<td>170</td>
<td>5.29</td>
<td>4.12</td>
<td>0.79</td>
<td>22.37</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>16.15-16.30</td>
<td>171</td>
<td>5.25</td>
<td>3.74</td>
<td>0.63</td>
<td>23.57</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>17.30-17.45</td>
<td>172</td>
<td>5.23</td>
<td>4.15</td>
<td>0.82</td>
<td>32.61</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19.45-20.00</td>
<td>172</td>
<td>5.26</td>
<td>4.11</td>
<td>0.56</td>
<td>22.54</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.00-14.15</td>
<td>172</td>
<td>5.23</td>
<td>4.44</td>
<td>0.44</td>
<td>30.11</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>17.15-17.30</td>
<td>172</td>
<td>5.24</td>
<td>3.67</td>
<td>0.58</td>
<td>16.00</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>12.00-12.15</td>
<td>172</td>
<td>5.19</td>
<td>4.10</td>
<td>0.31</td>
<td>21.19</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19.45-20.00</td>
<td>172</td>
<td>5.26</td>
<td>4.11</td>
<td>0.56</td>
<td>22.54</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>17.30-17.45</td>
<td>173</td>
<td>5.18</td>
<td>3.78</td>
<td>0.65</td>
<td>20.82</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>15.15-15.30</td>
<td>173</td>
<td>5.16</td>
<td>4.01</td>
<td>0.75</td>
<td>19.93</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>18.30-18.45</td>
<td>173</td>
<td>5.29</td>
<td>4.21</td>
<td>0.59</td>
<td>27.82</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>17.30-17.45</td>
<td>174</td>
<td>5.13</td>
<td>4.27</td>
<td>0.40</td>
<td>29.74</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>16.15-16.30</td>
<td>174</td>
<td>5.19</td>
<td>4.17</td>
<td>0.61</td>
<td>23.64</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>17.30-17.45</td>
<td>174</td>
<td>5.16</td>
<td>4.49</td>
<td>0.55</td>
<td>28.72</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.30-14.45</td>
<td>175</td>
<td>5.12</td>
<td>4.35</td>
<td>0.62</td>
<td>30.14</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13.00-13.15</td>
<td>175</td>
<td>5.14</td>
<td>4.79</td>
<td>0.61</td>
<td>31.27</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>15.30-15.45</td>
<td>175</td>
<td>5.17</td>
<td>4.17</td>
<td>0.52</td>
<td>22.04</td>
</tr>
</tbody>
</table>
Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Driving Lane (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/20/2004</td>
<td>13.45-14.00</td>
<td>176</td>
<td>5.13</td>
<td>4.12</td>
<td>0.72</td>
<td>19.06</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.45-15.00</td>
<td>176</td>
<td>5.11</td>
<td>3.98</td>
<td>0.73</td>
<td>24.23</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>15.00-15.15</td>
<td>176</td>
<td>5.12</td>
<td>3.75</td>
<td>0.57</td>
<td>21.90</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>11.45-12.00</td>
<td>176</td>
<td>5.11</td>
<td>4.68</td>
<td>0.58</td>
<td>30.88</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>18.15-18.30</td>
<td>176</td>
<td>5.04</td>
<td>3.83</td>
<td>0.67</td>
<td>23.21</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>18.45-19.00</td>
<td>177</td>
<td>5.07</td>
<td>4.09</td>
<td>0.64</td>
<td>24.52</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>12.30-12.45</td>
<td>178</td>
<td>5.05</td>
<td>4.08</td>
<td>0.49</td>
<td>24.19</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>15.45-16.00</td>
<td>178</td>
<td>5.06</td>
<td>3.83</td>
<td>0.37</td>
<td>23.62</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.30-11.45</td>
<td>178</td>
<td>5.07</td>
<td>4.27</td>
<td>0.54</td>
<td>25.36</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13.15-13.30</td>
<td>178</td>
<td>5.01</td>
<td>3.52</td>
<td>0.53</td>
<td>17.56</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>7:15-7:30</td>
<td>179</td>
<td>5.03</td>
<td>4.30</td>
<td>0.44</td>
<td>26.27</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>11.30-11.45</td>
<td>180</td>
<td>4.99</td>
<td>4.01</td>
<td>0.66</td>
<td>27.08</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>14.15-14.30</td>
<td>180</td>
<td>4.90</td>
<td>4.34</td>
<td>0.38</td>
<td>29.24</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>11.00-11.15</td>
<td>181</td>
<td>4.98</td>
<td>3.51</td>
<td>0.80</td>
<td>17.02</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>11.45-12.00</td>
<td>181</td>
<td>4.97</td>
<td>4.94</td>
<td>0.60</td>
<td>33.89</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.15-14.30</td>
<td>181</td>
<td>4.97</td>
<td>3.54</td>
<td>0.39</td>
<td>17.29</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>15.30-15.45</td>
<td>181</td>
<td>4.97</td>
<td>4.14</td>
<td>0.91</td>
<td>26.49</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>18.00-18.15</td>
<td>181</td>
<td>4.95</td>
<td>4.81</td>
<td>0.82</td>
<td>44.09</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13.30-13.45</td>
<td>182</td>
<td>4.99</td>
<td>3.66</td>
<td>0.64</td>
<td>17.33</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>17.30-17.45</td>
<td>182</td>
<td>4.95</td>
<td>3.77</td>
<td>0.62</td>
<td>22.89</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>14.45-15.00</td>
<td>183</td>
<td>4.90</td>
<td>3.72</td>
<td>0.69</td>
<td>19.01</td>
</tr>
</tbody>
</table>
Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime Driving Lane (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>10.30-10.45</td>
<td>184</td>
<td>4.86</td>
<td>3.90</td>
<td>0.73</td>
<td>22.56</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>15.15-15.30</td>
<td>184</td>
<td>4.91</td>
<td>3.49</td>
<td>0.60</td>
<td>22.50</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>16.30-16.45</td>
<td>184</td>
<td>4.93</td>
<td>3.66</td>
<td>0.43</td>
<td>20.24</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16.45-17.00</td>
<td>185</td>
<td>4.80</td>
<td>4.29</td>
<td>0.39</td>
<td>26.57</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.45-12.00</td>
<td>185</td>
<td>4.82</td>
<td>3.75</td>
<td>0.81</td>
<td>19.80</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16.15-16.30</td>
<td>186</td>
<td>4.79</td>
<td>3.41</td>
<td>0.64</td>
<td>20.42</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19.00-19.15</td>
<td>186</td>
<td>4.83</td>
<td>3.46</td>
<td>0.59</td>
<td>20.90</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16.45-17.00</td>
<td>186</td>
<td>4.85</td>
<td>3.89</td>
<td>0.69</td>
<td>23.91</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16.30-16.45</td>
<td>189</td>
<td>4.82</td>
<td>4.16</td>
<td>0.76</td>
<td>25.79</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>17.45-18.00</td>
<td>190</td>
<td>4.75</td>
<td>3.68</td>
<td>0.52</td>
<td>19.20</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>17.00-17.15</td>
<td>191</td>
<td>4.71</td>
<td>3.49</td>
<td>0.26</td>
<td>21.94</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>15.00-15.15</td>
<td>195</td>
<td>4.61</td>
<td>3.95</td>
<td>0.38</td>
<td>26.87</td>
</tr>
<tr>
<td>N=</td>
<td></td>
<td></td>
<td>159</td>
<td>159</td>
<td></td>
<td>159</td>
</tr>
<tr>
<td>Average=</td>
<td></td>
<td></td>
<td>153</td>
<td>6.28</td>
<td>5.26</td>
<td>65</td>
</tr>
<tr>
<td>Standard Deviation=</td>
<td>29.97</td>
<td>2.31</td>
<td>2.23</td>
<td>3.84</td>
<td>1.16</td>
<td>120</td>
</tr>
<tr>
<td>Minimum=</td>
<td></td>
<td></td>
<td>46</td>
<td>4.61</td>
<td>3.41</td>
<td>56</td>
</tr>
<tr>
<td>Maximum=</td>
<td></td>
<td></td>
<td>195</td>
<td>20.90</td>
<td>19.29</td>
<td>73</td>
</tr>
</tbody>
</table>

*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**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/22/2004</td>
<td>7.15-7.30</td>
<td>16</td>
<td>70.29</td>
<td>86.33</td>
<td>2.73</td>
<td>240.03</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>7.45-8.00</td>
<td>16</td>
<td>56.44</td>
<td>48.93</td>
<td>0.81</td>
<td>157.66</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>6:45-7:00</td>
<td>20</td>
<td>36.67</td>
<td>41.87</td>
<td>0.81</td>
<td>176.74</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>7:30-7:45</td>
<td>21</td>
<td>41.47</td>
<td>50.31</td>
<td>0.71</td>
<td>172.08</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>8:15-8.30</td>
<td>23</td>
<td>39.12</td>
<td>41.36</td>
<td>0.69</td>
<td>131.26</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>8:45-9.00</td>
<td>26</td>
<td>37.01</td>
<td>34.79</td>
<td>0.63</td>
<td>121.23</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>7:00-7.15</td>
<td>30</td>
<td>33.55</td>
<td>38.95</td>
<td>2.22</td>
<td>155.96</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>7:15-7.30</td>
<td>30</td>
<td>30.55</td>
<td>37.71</td>
<td>0.48</td>
<td>149.27</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>7:45-8.00</td>
<td>31</td>
<td>29.17</td>
<td>35.79</td>
<td>0.57</td>
<td>137.21</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>9:15-9.30</td>
<td>31</td>
<td>29.45</td>
<td>33.96</td>
<td>0.65</td>
<td>149.79</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>9:30-9.45</td>
<td>32</td>
<td>25.64</td>
<td>37.34</td>
<td>0.88</td>
<td>137.21</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>10:00-10:15</td>
<td>40</td>
<td>22.44</td>
<td>28.74</td>
<td>0.64</td>
<td>148.22</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>10:00-10:15</td>
<td>46</td>
<td>20.59</td>
<td>27.56</td>
<td>0.85</td>
<td>145.36</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>10:30-10:45</td>
<td>49</td>
<td>17.29</td>
<td>20.73</td>
<td>0.69</td>
<td>81.57</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11:00-11:15</td>
<td>49</td>
<td>15.44</td>
<td>13.34</td>
<td>0.49</td>
<td>47.75</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>11:30-11:45</td>
<td>49</td>
<td>18.29</td>
<td>30.01</td>
<td>0.58</td>
<td>181.63</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>12:00-12:15</td>
<td>52</td>
<td>17.41</td>
<td>20.98</td>
<td>0.58</td>
<td>80.30</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>12:30-12:45</td>
<td>54</td>
<td>16.90</td>
<td>19.29</td>
<td>0.28</td>
<td>91.31</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13:00-13:15</td>
<td>55</td>
<td>15.27</td>
<td>16.91</td>
<td>1.05</td>
<td>78.39</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13:30-13:45</td>
<td>56</td>
<td>16.02</td>
<td>21.47</td>
<td>0.69</td>
<td>114.03</td>
</tr>
</tbody>
</table>
### Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

**Passing Lane (continued)**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>19.15-19.30</td>
<td>58</td>
<td>15.58</td>
<td>19.79</td>
<td>0.66</td>
<td>86.13</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>17.45-18.00</td>
<td>59</td>
<td>15.14</td>
<td>17.42</td>
<td>0.78</td>
<td>99.36</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>19.45-20.00</td>
<td>63</td>
<td>15.66</td>
<td>22.58</td>
<td>0.68</td>
<td>151.05</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>10.15-10.30</td>
<td>63</td>
<td>14.33</td>
<td>21.54</td>
<td>0.53</td>
<td>121.28</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>8.15-8.30</td>
<td>64</td>
<td>14.41</td>
<td>17.36</td>
<td>0.49</td>
<td>80.40</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>9.45-10.00</td>
<td>67</td>
<td>13.02</td>
<td>14.55</td>
<td>0.61</td>
<td>72.88</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>19.30-19.45</td>
<td>68</td>
<td>13.19</td>
<td>20.22</td>
<td>0.72</td>
<td>87.23</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>20.00-20.15</td>
<td>68</td>
<td>13.11</td>
<td>15.86</td>
<td>0.58</td>
<td>67.84</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>9.15-9.30</td>
<td>68</td>
<td>14.20</td>
<td>18.25</td>
<td>0.61</td>
<td>84.55</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>18.00-18.15</td>
<td>68</td>
<td>13.46</td>
<td>21.37</td>
<td>0.70</td>
<td>141.96</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>9.00-9.15</td>
<td>69</td>
<td>12.12</td>
<td>14.59</td>
<td>0.49</td>
<td>56.42</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>10.45-11.00</td>
<td>69</td>
<td>12.88</td>
<td>16.33</td>
<td>0.55</td>
<td>73.03</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>15.45-16.00</td>
<td>70</td>
<td>12.84</td>
<td>14.59</td>
<td>0.53</td>
<td>68.45</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>18.15-18.30</td>
<td>71</td>
<td>12.67</td>
<td>20.18</td>
<td>0.56</td>
<td>92.68</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>19.00-19.15</td>
<td>74</td>
<td>12.24</td>
<td>14.53</td>
<td>0.81</td>
<td>63.66</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>8.45-9.00</td>
<td>74</td>
<td>12.08</td>
<td>15.70</td>
<td>0.48</td>
<td>103.44</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>11.15-11.30</td>
<td>74</td>
<td>13.14</td>
<td>21.34</td>
<td>0.53</td>
<td>121.05</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>8.30-8.45</td>
<td>75</td>
<td>12.51</td>
<td>17.75</td>
<td>0.48</td>
<td>68.04</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>9.00-9.15</td>
<td>76</td>
<td>11.71</td>
<td>15.35</td>
<td>0.40</td>
<td>78.09</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>19.15-19.30</td>
<td>77</td>
<td>11.14</td>
<td>14.19</td>
<td>0.62</td>
<td>82.89</td>
</tr>
</tbody>
</table>
Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>6:45-7:00</td>
<td>78</td>
<td>11.31</td>
<td>11.72</td>
<td>0.60</td>
<td>65.84</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>18:45-19:00</td>
<td>79</td>
<td>11.32</td>
<td>15.46</td>
<td>0.54</td>
<td>74.36</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13:00-13:15</td>
<td>79</td>
<td>10.75</td>
<td>15.07</td>
<td>0.51</td>
<td>81.51</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>10:15-10:30</td>
<td>80</td>
<td>11.15</td>
<td>16.67</td>
<td>0.50</td>
<td>84.57</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>11:45-12:00</td>
<td>80</td>
<td>10.94</td>
<td>13.86</td>
<td>0.69</td>
<td>67.44</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>18:45-19:00</td>
<td>81</td>
<td>11.09</td>
<td>12.73</td>
<td>0.74</td>
<td>50.93</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>19:45-20:00</td>
<td>81</td>
<td>11.57</td>
<td>14.56</td>
<td>0.41</td>
<td>65.69</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>10:00-10:15</td>
<td>81</td>
<td>11.12</td>
<td>13.81</td>
<td>0.62</td>
<td>66.63</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>18:30-18:45</td>
<td>81</td>
<td>11.43</td>
<td>12.44</td>
<td>0.53</td>
<td>53.31</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>9:30-9:45</td>
<td>82</td>
<td>10.47</td>
<td>15.67</td>
<td>0.33</td>
<td>78.55</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>16:45-17:00</td>
<td>82</td>
<td>10.94</td>
<td>11.30</td>
<td>0.50</td>
<td>49.31</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>17:00-17:15</td>
<td>82</td>
<td>10.75</td>
<td>13.31</td>
<td>0.35</td>
<td>61.42</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13:30-13:45</td>
<td>83</td>
<td>10.83</td>
<td>15.89</td>
<td>0.60</td>
<td>76.68</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>17:30-17:45</td>
<td>83</td>
<td>11.03</td>
<td>14.50</td>
<td>0.44</td>
<td>90.81</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>12:00-12:15</td>
<td>83</td>
<td>11.06</td>
<td>10.53</td>
<td>0.49</td>
<td>45.79</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13:45-14:00</td>
<td>83</td>
<td>10.74</td>
<td>12.30</td>
<td>0.51</td>
<td>50.54</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>14:45-15:00</td>
<td>83</td>
<td>10.84</td>
<td>10.94</td>
<td>0.46</td>
<td>51.48</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11:00-11:15</td>
<td>84</td>
<td>10.85</td>
<td>12.40</td>
<td>0.60</td>
<td>66.21</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13:00-13:15</td>
<td>86</td>
<td>10.88</td>
<td>13.56</td>
<td>0.59</td>
<td>64.65</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>16:30-16:45</td>
<td>86</td>
<td>10.36</td>
<td>11.56</td>
<td>0.47</td>
<td>63.05</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>18:00-18:15</td>
<td>86</td>
<td>10.27</td>
<td>11.84</td>
<td>0.54</td>
<td>63.26</td>
</tr>
</tbody>
</table>
Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Inter-arrival Time</td>
<td>Speed</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>16.00-16.15</td>
<td>88</td>
<td>10.04</td>
<td>14.51</td>
<td>0.53</td>
<td>68.43</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>16.15-16.30</td>
<td>88</td>
<td>10.43</td>
<td>12.03</td>
<td>0.51</td>
<td>57.15</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>11.30-11.45</td>
<td>88</td>
<td>10.17</td>
<td>10.99</td>
<td>0.57</td>
<td>43.39</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>12.00-12.15</td>
<td>89</td>
<td>10.15</td>
<td>12.71</td>
<td>0.55</td>
<td>49.34</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>12.30-12.45</td>
<td>89</td>
<td>10.17</td>
<td>12.08</td>
<td>0.56</td>
<td>62.65</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13.15-13.30</td>
<td>89</td>
<td>10.59</td>
<td>13.75</td>
<td>0.56</td>
<td>68.60</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>13.15-13.30</td>
<td>92</td>
<td>9.81</td>
<td>12.78</td>
<td>0.52</td>
<td>78.58</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>11.15-11.30</td>
<td>92</td>
<td>9.72</td>
<td>13.22</td>
<td>0.67</td>
<td>65.29</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.15-11.30</td>
<td>92</td>
<td>9.84</td>
<td>12.21</td>
<td>0.58</td>
<td>60.89</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>15.30-15.45</td>
<td>92</td>
<td>9.86</td>
<td>11.43</td>
<td>0.68</td>
<td>70.78</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19.15-19.30</td>
<td>92</td>
<td>9.61</td>
<td>11.44</td>
<td>0.52</td>
<td>57.25</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>7.00-7.15</td>
<td>93</td>
<td>9.63</td>
<td>13.01</td>
<td>0.59</td>
<td>70.70</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>10.15-10.30</td>
<td>94</td>
<td>9.68</td>
<td>11.40</td>
<td>0.69</td>
<td>50.06</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>12.00-12.15</td>
<td>94</td>
<td>9.34</td>
<td>11.27</td>
<td>0.49</td>
<td>59.64</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>12.45-13.00</td>
<td>94</td>
<td>9.71</td>
<td>11.43</td>
<td>0.32</td>
<td>59.22</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>8.30-8.45</td>
<td>95</td>
<td>9.49</td>
<td>8.46</td>
<td>0.51</td>
<td>40.78</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>14.30-14.45</td>
<td>95</td>
<td>9.60</td>
<td>12.39</td>
<td>0.36</td>
<td>65.23</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.45-12.00</td>
<td>96</td>
<td>9.30</td>
<td>12.73</td>
<td>0.60</td>
<td>77.35</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19.30-19.45</td>
<td>96</td>
<td>9.30</td>
<td>13.28</td>
<td>0.36</td>
<td>75.20</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>11.30-11.45</td>
<td>97</td>
<td>9.32</td>
<td>10.32</td>
<td>0.34</td>
<td>54.16</td>
</tr>
</tbody>
</table>
Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>18.15-18.30</td>
<td>107</td>
<td>8.29</td>
<td>12.39</td>
<td>0.47</td>
<td>78.41</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>17.30-17.45</td>
<td>108</td>
<td>8.54</td>
<td>10.30</td>
<td>0.51</td>
<td>54.99</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>10.45-11.00</td>
<td>109</td>
<td>8.44</td>
<td>12.41</td>
<td>0.50</td>
<td>92.56</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>13.15-13.30</td>
<td>109</td>
<td>8.20</td>
<td>8.91</td>
<td>0.41</td>
<td>35.90</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>18.45-19.00</td>
<td>109</td>
<td>8.45</td>
<td>11.68</td>
<td>0.42</td>
<td>55.51</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.00-14.15</td>
<td>110</td>
<td>8.07</td>
<td>11.09</td>
<td>0.58</td>
<td>73.68</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16.00-16.15</td>
<td>110</td>
<td>8.19</td>
<td>9.19</td>
<td>0.60</td>
<td>40.65</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>12.45-13.00</td>
<td>110</td>
<td>7.85</td>
<td>9.39</td>
<td>0.69</td>
<td>44.12</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>12.30-12.45</td>
<td>110</td>
<td>8.06</td>
<td>8.22</td>
<td>0.50</td>
<td>39.97</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>11.45-12.00</td>
<td>111</td>
<td>8.03</td>
<td>11.36</td>
<td>0.49</td>
<td>64.75</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>14.00-14.15</td>
<td>112</td>
<td>8.22</td>
<td>11.44</td>
<td>0.40</td>
<td>63.89</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>15.15-15.30</td>
<td>112</td>
<td>7.94</td>
<td>10.49</td>
<td>0.38</td>
<td>52.48</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>17.45-18.00</td>
<td>113</td>
<td>7.95</td>
<td>9.80</td>
<td>0.49</td>
<td>46.51</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>11.30-11.45</td>
<td>115</td>
<td>7.82</td>
<td>10.45</td>
<td>0.43</td>
<td>55.96</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>7.15-7.30</td>
<td>116</td>
<td>7.83</td>
<td>10.51</td>
<td>0.28</td>
<td>62.54</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>14.15-14.30</td>
<td>116</td>
<td>7.56</td>
<td>9.67</td>
<td>0.37</td>
<td>45.91</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16.45-17.00</td>
<td>117</td>
<td>7.90</td>
<td>11.50</td>
<td>0.42</td>
<td>88.21</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>17.15-17.30</td>
<td>117</td>
<td>7.49</td>
<td>8.89</td>
<td>0.43</td>
<td>47.02</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>12.15-12.30</td>
<td>118</td>
<td>7.54</td>
<td>11.45</td>
<td>0.48</td>
<td>89.97</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>18.15-18.30</td>
<td>118</td>
<td>7.59</td>
<td>8.75</td>
<td>0.55</td>
<td>35.47</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>10.30-10.45</td>
<td>119</td>
<td>7.40</td>
<td>9.36</td>
<td>0.48</td>
<td>44.01</td>
</tr>
</tbody>
</table>
**Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime Passing Lane (continued)**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>18.30-18.45</td>
<td>119</td>
<td>7.69</td>
<td>10.14</td>
<td>0.42</td>
<td>55.10</td>
</tr>
<tr>
<td>8/21/2004</td>
<td>14.45-15.00</td>
<td>119</td>
<td>7.47</td>
<td>10.40</td>
<td>0.47</td>
<td>73.44</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>13.30-13.45</td>
<td>119</td>
<td>7.62</td>
<td>10.12</td>
<td>0.51</td>
<td>48.00</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>16.30-16.45</td>
<td>119</td>
<td>7.61</td>
<td>8.86</td>
<td>0.38</td>
<td>45.15</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>15.30-15.45</td>
<td>120</td>
<td>7.63</td>
<td>9.85</td>
<td>0.51</td>
<td>59.38</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>18.30-18.45</td>
<td>121</td>
<td>7.29</td>
<td>8.77</td>
<td>0.44</td>
<td>40.40</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>15.30-15.45</td>
<td>125</td>
<td>7.24</td>
<td>10.08</td>
<td>0.38</td>
<td>65.50</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19.45-20.00</td>
<td>125</td>
<td>7.39</td>
<td>8.95</td>
<td>0.40</td>
<td>57.82</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.45-15.00</td>
<td>126</td>
<td>7.18</td>
<td>9.52</td>
<td>0.54</td>
<td>50.87</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>13.45-14.00</td>
<td>127</td>
<td>7.08</td>
<td>9.24</td>
<td>0.29</td>
<td>63.52</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>16.15-16.30</td>
<td>127</td>
<td>7.13</td>
<td>9.23</td>
<td>0.56</td>
<td>44.40</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>15.15-15.30</td>
<td>128</td>
<td>7.01</td>
<td>9.24</td>
<td>0.50</td>
<td>60.20</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>11.00-11.15</td>
<td>129</td>
<td>7.01</td>
<td>9.79</td>
<td>0.42</td>
<td>59.77</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16.15-16.30</td>
<td>129</td>
<td>6.82</td>
<td>9.05</td>
<td>0.42</td>
<td>49.23</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>15.00-15.15</td>
<td>130</td>
<td>6.95</td>
<td>8.30</td>
<td>0.36</td>
<td>45.13</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>17.15-17.30</td>
<td>130</td>
<td>7.01</td>
<td>8.73</td>
<td>0.33</td>
<td>51.96</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>15.45-16.00</td>
<td>131</td>
<td>6.86</td>
<td>8.94</td>
<td>0.63</td>
<td>56.10</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>7.30-7.45</td>
<td>132</td>
<td>6.80</td>
<td>9.25</td>
<td>0.53</td>
<td>68.40</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.30-14.45</td>
<td>132</td>
<td>6.81</td>
<td>8.74</td>
<td>0.51</td>
<td>46.47</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>16.00-16.15</td>
<td>132</td>
<td>6.88</td>
<td>8.98</td>
<td>0.29</td>
<td>50.66</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>12.30-12.45</td>
<td>134</td>
<td>6.79</td>
<td>8.98</td>
<td>0.44</td>
<td>49.13</td>
</tr>
</tbody>
</table>
Hourly Traffic Counts (vphpl), Average IAT, Average Speed, and Standard Deviations for I-76 Westbound Daytime

Passing Lane (continued)

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>19.00-19.15</td>
<td>134</td>
<td>6.72</td>
<td>10.16</td>
<td>0.51</td>
<td>58.59</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.45-15.00</td>
<td>137</td>
<td>6.56</td>
<td>7.97</td>
<td>0.28</td>
<td>43.76</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>17.30-17.45</td>
<td>144</td>
<td>6.19</td>
<td>7.89</td>
<td>0.30</td>
<td>48.79</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>17.00-17.15</td>
<td>146</td>
<td>6.18</td>
<td>8.21</td>
<td>0.36</td>
<td>42.74</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>17.45-18.00</td>
<td>150</td>
<td>5.91</td>
<td>8.17</td>
<td>0.44</td>
<td>45.37</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>16.30-16.45</td>
<td>151</td>
<td>5.91</td>
<td>7.47</td>
<td>0.54</td>
<td>51.02</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>18.00-18.15</td>
<td>151</td>
<td>6.12</td>
<td>9.07</td>
<td>0.42</td>
<td>53.72</td>
</tr>
<tr>
<td>8/22/2004</td>
<td>16.45-17.00</td>
<td>157</td>
<td>5.64</td>
<td>7.53</td>
<td>0.54</td>
<td>53.81</td>
</tr>
<tr>
<td>8/20/2004</td>
<td>14.15-14.30</td>
<td>162</td>
<td>5.62</td>
<td>7.66</td>
<td>0.50</td>
<td>45.05</td>
</tr>
<tr>
<td>N=</td>
<td></td>
<td>156</td>
<td>156</td>
<td>156</td>
<td>156</td>
<td>156</td>
</tr>
<tr>
<td>Average=</td>
<td></td>
<td>92</td>
<td>12.07</td>
<td>15.02</td>
<td>70</td>
<td>3.30</td>
</tr>
<tr>
<td>Standard Deviation=</td>
<td>30.83</td>
<td>8.70</td>
<td>9.80</td>
<td>2.14</td>
<td>0.91</td>
<td>123</td>
</tr>
<tr>
<td>Minimum=</td>
<td></td>
<td>16</td>
<td>5.62</td>
<td>7.47</td>
<td>63</td>
<td>1.19</td>
</tr>
<tr>
<td>Maximum=</td>
<td></td>
<td>162</td>
<td>70.29</td>
<td>86.33</td>
<td>76</td>
<td>5.91</td>
</tr>
</tbody>
</table>

*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

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Number of Vehicles for 15 Minutes Intervals</th>
<th>Inter-arrival Time (second)</th>
<th>Speed (mph)</th>
<th>Number of Vehicles per Hour</th>
<th>Adjusted Number of Vehicles per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Average</td>
<td>Standard Deviation</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>*8/22/2004</td>
<td>9.00-9.15</td>
<td>37</td>
<td>24.46</td>
<td>27.66</td>
<td>0.62</td>
<td>91.45</td>
</tr>
<tr>
<td>*8/20/2004</td>
<td>13.00-13.15</td>
<td>102</td>
<td>8.76</td>
<td>12.58</td>
<td>0.33</td>
<td>59.92</td>
</tr>
<tr>
<td>*8/20/2004</td>
<td>12.45-13.00</td>
<td>126</td>
<td>7.20</td>
<td>9.40</td>
<td>0.45</td>
<td>46.04</td>
</tr>
<tr>
<td>**8/22/2004</td>
<td>6:45-7:00</td>
<td>5</td>
<td>152.50</td>
<td>111.55</td>
<td>56.79</td>
<td>324.31</td>
</tr>
<tr>
<td>**8/22/2004</td>
<td>7:00-7.15</td>
<td>6</td>
<td>129.10</td>
<td>142.13</td>
<td>0.90</td>
<td>382.97</td>
</tr>
<tr>
<td>**8/22/2004</td>
<td>8.00-8.15</td>
<td>15</td>
<td>61.82</td>
<td>41.23</td>
<td>12.81</td>
<td>139.87</td>
</tr>
</tbody>
</table>
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%

\[ y = \frac{122.05}{x} + 0.4946 \]
\[ R \text{ square} = 0.1824 \]

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

\[ y = \frac{183.82}{x} + 0.5566 \]
\[ R \text{ square} = 0.3271 \]
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%

\[ y = \frac{892.66}{x} + 0.6246 \]

\[ R^2 = 0.8409 \]

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

\[ y = \frac{1254.82}{x} + 0.6555 \]

\[ R^2 = 0.9090 \]
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 60%

\[ y = \frac{3208.25}{x} + 0.2548 \]
\[ R^2 = 0.9541 \]

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

\[ y = \frac{4295.84}{x} - 0.1387 \]
\[ R^2 = 0.9640 \]
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 80%

\[ y = \frac{5390.19}{x} + 0.0199 \]
\[ R^2 = 0.9641 \]

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

\[ y = \frac{7592.15}{x} - 0.1767 \]
\[ R^2 = 0.9167 \]
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 95%

\[ y = \frac{10848.19}{x} - 2.2824 \]
\[ R \text{ square} = 0.9179 \]

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

\[ y = \frac{12050.26}{x} - 0.3884 \]
\[ R \text{ square} = 0.8841 \]
IAT as a Function of Volume (vphpl) on Driving Lane during Daytime for Cumulative Percentage Value 99%

\[ y = \frac{12842.41}{x} + 0.8546 \]
\[ R^2 = 0.8292 \]

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

\[ y = \frac{13495.82}{x} + 6.3495 \]
\[ R^2 = 0.6329 \]
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 1%

\[ y = \frac{12.22}{x} + 0.4753 \]
\[ R^2 = 0.0931 \]
\[ N = 154 \]

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

\[ y = \frac{29.33}{x} + 0.5211 \]
\[ R^2 = 0.1605 \]
\[ N = 154 \]
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 5%

\[ y = \frac{57.77}{x} + 0.5776 \]
\[ R^2 = 0.2743 \]
\[ N = 156 \]

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

\[ y = \frac{116.77}{x} + 0.6024 \]
\[ R^2 = 0.3636 \]
\[ N = 156 \]
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 20%

\[ y = \frac{443.98}{x} + 0.1633 \]
\[ R^2 = 0.6755 \]
\[ N = 156 \]

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

\[ y = \frac{762.81}{x} - 0.0794 \]
\[ R^2 = 0.7104 \]
\[ N = 156 \]
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 40%

\[ y = \frac{1266.43}{x} - 0.3614 \]
\[ R^2 = 0.8680 \]
\[ N = 156 \]

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

\[ y = \frac{1976.05}{x} - 0.6918 \]
\[ R^2 = 0.8915 \]
\[ N = 156 \]
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 60%

\[ y = \frac{4388.64}{x} - 1.0848 \]
\[ R^2 = 0.9378 \]
\[ N = 156 \]

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

\[ y = \frac{4295.84}{x} - 0.1387 \]
\[ R^2 = 0.9640 \]
\[ N = 159 \]
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 80%

\[ y = 6357.94/x - 1.0673 \]
\[ R^2 = 0.9319 \]
\[ N = 156 \]

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

\[ y = 9498.70/x - 0.2287 \]
\[ R^2 = 0.8455 \]
\[ N = 156 \]
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 95%

\[ y = 10960.47/x + 5.3074 \]

\[ R^2 = 0.8692 \]

\[ N = 156 \]

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

\[ y = 11411.96/x + 14.6193 \]

\[ R^2 = 0.8042 \]

\[ N = 154 \]
IAT as a Function of Volume (vphpl) on Passing Lane during Daytime for Cumulative Percentage Value 99%

\[ y = \frac{11656.48}{x} + 21.5875 \]
\[ \text{R square} = 0.7491 \]
\[ N = 154 \]

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

\[ y = \frac{12419.46}{x} + 32.9255 \]
\[ \text{R square} = 0.6723 \]
\[ N = 156 \]
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

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
<td>1%</td>
</tr>
<tr>
<td>0:00-0:15</td>
<td>0.10</td>
<td>1.04</td>
</tr>
<tr>
<td>0:15-0:30</td>
<td>0.10</td>
<td>1.15</td>
</tr>
<tr>
<td>0:30-0:45</td>
<td>0.10</td>
<td>1.11</td>
</tr>
<tr>
<td>0:45-1:00</td>
<td>0.10</td>
<td>1.11</td>
</tr>
<tr>
<td>1:00-1:15</td>
<td>0.10</td>
<td>1.43</td>
</tr>
<tr>
<td>1:15-1:30</td>
<td>0.10</td>
<td>1.29</td>
</tr>
<tr>
<td>1:30-1:45</td>
<td>0.10</td>
<td>1.31</td>
</tr>
<tr>
<td>1:45-2:00</td>
<td>0.10</td>
<td>1.11</td>
</tr>
<tr>
<td>2:00-2:15</td>
<td>0.10</td>
<td>1.24</td>
</tr>
<tr>
<td>2:15-2:30</td>
<td>0.10</td>
<td>1.05</td>
</tr>
<tr>
<td>2:30-2:45</td>
<td>0.10</td>
<td>1.12</td>
</tr>
<tr>
<td>2:45-3:00</td>
<td>0.10</td>
<td>1.31</td>
</tr>
<tr>
<td>3:00-3:15</td>
<td>0.10</td>
<td>1.34</td>
</tr>
<tr>
<td>3:15-3:30</td>
<td>0.10</td>
<td>1.17</td>
</tr>
<tr>
<td>3:30-3:45</td>
<td>0.10</td>
<td>1.33</td>
</tr>
<tr>
<td>3:45-4:00</td>
<td>0.10</td>
<td>1.15</td>
</tr>
<tr>
<td>4:00-4:15</td>
<td>0.10</td>
<td>1.02</td>
</tr>
<tr>
<td>4:15-4:30</td>
<td>0.10</td>
<td>1.20</td>
</tr>
<tr>
<td>4:30-4:45</td>
<td>0.10</td>
<td>1.11</td>
</tr>
<tr>
<td>4:45-5:00</td>
<td>0.10</td>
<td>1.02</td>
</tr>
</tbody>
</table>
Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane for ARENA Simulation Model (continued)

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
<td>1%</td>
</tr>
<tr>
<td>5:00-5:15</td>
<td>0.10</td>
<td>1.04</td>
</tr>
<tr>
<td>5:15-5:30</td>
<td>0.10</td>
<td>0.85</td>
</tr>
<tr>
<td>5:30-5:45</td>
<td>0.10</td>
<td>0.84</td>
</tr>
<tr>
<td>5:45-6:00</td>
<td>0.10</td>
<td>0.92</td>
</tr>
<tr>
<td>6:00-6:15</td>
<td>0.10</td>
<td>0.80</td>
</tr>
<tr>
<td>6:15-6:30</td>
<td>0.10</td>
<td>0.73</td>
</tr>
<tr>
<td>6:30-6:45</td>
<td>0.10</td>
<td>0.68</td>
</tr>
<tr>
<td>6:45-7:00</td>
<td>0.10</td>
<td>0.72</td>
</tr>
<tr>
<td>7:00-7:15</td>
<td>0.10</td>
<td>0.70</td>
</tr>
<tr>
<td>7:15-7:30</td>
<td>0.10</td>
<td>0.66</td>
</tr>
<tr>
<td>7:30-7:45</td>
<td>0.10</td>
<td>0.67</td>
</tr>
<tr>
<td>7:45-8:00</td>
<td>0.10</td>
<td>0.70</td>
</tr>
<tr>
<td>8:00-8:15</td>
<td>0.10</td>
<td>0.68</td>
</tr>
<tr>
<td>8:15-8:30</td>
<td>0.10</td>
<td>0.68</td>
</tr>
<tr>
<td>8:30-8:45</td>
<td>0.10</td>
<td>0.69</td>
</tr>
<tr>
<td>8:45-9:00</td>
<td>0.10</td>
<td>0.71</td>
</tr>
<tr>
<td>9:00-9:15</td>
<td>0.10</td>
<td>0.70</td>
</tr>
<tr>
<td>9:15-9:30</td>
<td>0.10</td>
<td>0.71</td>
</tr>
<tr>
<td>9:30-9:45</td>
<td>0.10</td>
<td>0.68</td>
</tr>
<tr>
<td>9:45-10:00</td>
<td>0.10</td>
<td>0.70</td>
</tr>
<tr>
<td>10:00-10:15</td>
<td>0.10</td>
<td>0.71</td>
</tr>
</tbody>
</table>
## Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane

for ARENA Simulation Model (continued)

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0% 1% 2% 5% 10% 20% 30% 40% 50% 60% 70% 80% 90% 95% 98% 99% 100% (Max)</td>
<td></td>
</tr>
<tr>
<td>10.15-10.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>10.30-10.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>10.45-11.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>11.00-11.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>11.15-11.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>11.30-11.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>11.45-12.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>12.00-12.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>12.15-12.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>12.30-12.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>12.45-13.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>13.00-13.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>13.15-13.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>13.30-13.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>13.45-14.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>14.00-14.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>14.15-14.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>14.30-14.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>14.45-15.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>15.00-15.15</td>
<td>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</td>
<td></td>
</tr>
</tbody>
</table>
Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane
for ARENA Simulation Model (continued)

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
<td>1%</td>
</tr>
<tr>
<td>15.15-15.30</td>
<td>0.10</td>
<td>0.66</td>
</tr>
<tr>
<td>15.30-15.45</td>
<td>0.10</td>
<td>0.66</td>
</tr>
<tr>
<td>15.45-16.00</td>
<td>0.10</td>
<td>0.66</td>
</tr>
<tr>
<td>16.00-16.15</td>
<td>0.10</td>
<td>0.70</td>
</tr>
<tr>
<td>16.15-16.30</td>
<td>0.10</td>
<td>0.66</td>
</tr>
<tr>
<td>16.30-16.45</td>
<td>0.10</td>
<td>0.65</td>
</tr>
<tr>
<td>16.45-17.00</td>
<td>0.10</td>
<td>0.66</td>
</tr>
<tr>
<td>17.00-17.15</td>
<td>0.10</td>
<td>0.68</td>
</tr>
<tr>
<td>17.15-17.30</td>
<td>0.10</td>
<td>0.67</td>
</tr>
<tr>
<td>17.30-17.45</td>
<td>0.10</td>
<td>0.67</td>
</tr>
<tr>
<td>17.45-18.00</td>
<td>0.10</td>
<td>0.70</td>
</tr>
<tr>
<td>18.00-18.15</td>
<td>0.10</td>
<td>0.69</td>
</tr>
<tr>
<td>18.15-18.30</td>
<td>0.10</td>
<td>0.72</td>
</tr>
<tr>
<td>18.30-18.45</td>
<td>0.10</td>
<td>0.71</td>
</tr>
<tr>
<td>18.45-19.00</td>
<td>0.10</td>
<td>0.72</td>
</tr>
<tr>
<td>19.00-19.15</td>
<td>0.10</td>
<td>0.72</td>
</tr>
<tr>
<td>19.15-19.30</td>
<td>0.10</td>
<td>0.74</td>
</tr>
<tr>
<td>19.30-19.45</td>
<td>0.10</td>
<td>0.76</td>
</tr>
<tr>
<td>19.45-20.00</td>
<td>0.10</td>
<td>0.76</td>
</tr>
<tr>
<td>20.00-20:15</td>
<td>0.10</td>
<td>0.77</td>
</tr>
</tbody>
</table>
Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Driving Lane for ARENA Simulation Model (continued)

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
<td>1%</td>
</tr>
<tr>
<td>20:15-20:30</td>
<td>0.10</td>
<td>0.81</td>
</tr>
<tr>
<td>20:30-20:45</td>
<td>0.10</td>
<td>0.77</td>
</tr>
<tr>
<td>20:45-21:00</td>
<td>0.10</td>
<td>0.79</td>
</tr>
<tr>
<td>21:00-21:15</td>
<td>0.10</td>
<td>0.77</td>
</tr>
<tr>
<td>21:15-21:30</td>
<td>0.10</td>
<td>0.83</td>
</tr>
<tr>
<td>21:30-21:45</td>
<td>0.10</td>
<td>0.79</td>
</tr>
<tr>
<td>21:45-22:00</td>
<td>0.10</td>
<td>0.85</td>
</tr>
<tr>
<td>22:00-22:15</td>
<td>0.10</td>
<td>0.91</td>
</tr>
<tr>
<td>22:15-22:30</td>
<td>0.10</td>
<td>0.85</td>
</tr>
<tr>
<td>22:30-22:45</td>
<td>0.10</td>
<td>0.90</td>
</tr>
<tr>
<td>22:45-23:00</td>
<td>0.10</td>
<td>0.92</td>
</tr>
<tr>
<td>23:00-23:15</td>
<td>0.10</td>
<td>1.00</td>
</tr>
<tr>
<td>23:15-23:30</td>
<td>0.10</td>
<td>0.91</td>
</tr>
<tr>
<td>23:30-23:45</td>
<td>0.10</td>
<td>0.91</td>
</tr>
<tr>
<td>23:45-0:00</td>
<td>0.10</td>
<td>1.03</td>
</tr>
</tbody>
</table>
Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane

for ARENA Simulation Model

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0% 1% 2% 5% 10% 20% 30% 40% 50% 60% 70% 80% 90% 95% 98% 99% 100% (Max)</td>
<td></td>
</tr>
<tr>
<td>0:00-0:15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>0:15-0:30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>0:30-0:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>0:45-1:00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>1:00-1:15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>1:15-1:30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>1:30-1:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>1:45-2:00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>2:00-2:15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>2:15-2:30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>2:30-2:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>2:45-3:00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>3:00-3:15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>3:15-3:30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>3:30-3:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>3:45-4:00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>4:00-4:15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>4:15-4:30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>4:30-4:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>4:45-5:00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>5:00-5:15</td>
<td>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</td>
<td></td>
</tr>
</tbody>
</table>
Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane
for ARENA Simulation Model (continued)

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0% 1% 2% 5% 10% 20% 30% 40% 50% 60% 70% 80% 90% 95% 98% 99% 100% (Max)</td>
<td></td>
</tr>
<tr>
<td>5:15-5:30</td>
<td>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</td>
<td>103</td>
</tr>
<tr>
<td>5:30-5:45</td>
<td>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</td>
<td>152</td>
</tr>
<tr>
<td>5:45-6:00</td>
<td>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</td>
<td>107</td>
</tr>
<tr>
<td>6:00-6:15</td>
<td>0.10 0.57 0.73 0.97 1.39 3.07 4.91 7.91 12.13 19.33 27.59 40.49 61.90 77.18 89.74 98.54 115.24</td>
<td>160</td>
</tr>
<tr>
<td>6:15-6:30</td>
<td>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</td>
<td>333</td>
</tr>
<tr>
<td>6:30-6:45</td>
<td>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</td>
<td>498</td>
</tr>
<tr>
<td>6:45-7:00</td>
<td>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</td>
<td>321</td>
</tr>
<tr>
<td>7:00-7:15</td>
<td>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</td>
<td>382</td>
</tr>
<tr>
<td>7:15-7:30</td>
<td>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</td>
<td>477</td>
</tr>
<tr>
<td>7:30-7:45</td>
<td>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</td>
<td>543</td>
</tr>
<tr>
<td>7:45-8:00</td>
<td>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</td>
<td>518</td>
</tr>
<tr>
<td>8:00-8:15</td>
<td>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</td>
<td>428</td>
</tr>
<tr>
<td>8:15-8:30</td>
<td>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</td>
<td>378</td>
</tr>
<tr>
<td>8:30-8:45</td>
<td>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</td>
<td>391</td>
</tr>
<tr>
<td>8:45-9:00</td>
<td>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</td>
<td>333</td>
</tr>
<tr>
<td>9:00-9:15</td>
<td>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</td>
<td>313</td>
</tr>
<tr>
<td>9:15-9:30</td>
<td>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</td>
<td>403</td>
</tr>
<tr>
<td>9:30-9:45</td>
<td>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</td>
<td>387</td>
</tr>
<tr>
<td>9:45-10:00</td>
<td>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</td>
<td>276</td>
</tr>
<tr>
<td>10:00-10:15</td>
<td>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</td>
<td>407</td>
</tr>
<tr>
<td>10:15-10:30</td>
<td>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</td>
<td>387</td>
</tr>
</tbody>
</table>
Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane

for ARENA Simulation Model (continued)

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0% 1% 2% 5% 10% 20% 30% 40% 50% 60% 70% 80% 90% 95% 98% 99%</td>
<td>(Max)</td>
</tr>
<tr>
<td>10.30-10.45</td>
<td>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</td>
<td>47.69 60.66</td>
</tr>
<tr>
<td>10.45-11.00</td>
<td>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</td>
<td>50.76 63.99</td>
</tr>
<tr>
<td>11.00-11.15</td>
<td>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</td>
<td>45.13 57.88</td>
</tr>
<tr>
<td>11.15-11.30</td>
<td>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</td>
<td>52.61 65.99</td>
</tr>
<tr>
<td>11.30-11.45</td>
<td>0.10 0.50 0.59 0.72 0.89 1.27 1.83 2.81 4.25 6.58 9.90 14.84 23.58 32.72 43.14</td>
<td>50.71 63.93</td>
</tr>
<tr>
<td>11.45-12.00</td>
<td>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</td>
<td>49.59 62.72</td>
</tr>
<tr>
<td>12.00-12.15</td>
<td>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</td>
<td>53.70 67.17</td>
</tr>
<tr>
<td>12.15-12.30</td>
<td>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</td>
<td>47.92 60.91</td>
</tr>
<tr>
<td>12.30-12.45</td>
<td>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</td>
<td>43.77 56.40</td>
</tr>
<tr>
<td>12.45-13.00</td>
<td>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</td>
<td>45.70 58.50</td>
</tr>
<tr>
<td>13.00-13.15</td>
<td>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</td>
<td>52.82 66.23</td>
</tr>
<tr>
<td>13.15-13.30</td>
<td>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</td>
<td>53.68 67.15</td>
</tr>
<tr>
<td>13.30-13.45</td>
<td>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</td>
<td>53.96 67.46</td>
</tr>
<tr>
<td>13.45-14.00</td>
<td>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</td>
<td>45.51 58.29</td>
</tr>
<tr>
<td>14.00-14.15</td>
<td>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</td>
<td>49.81 62.97</td>
</tr>
<tr>
<td>14.15-14.30</td>
<td>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</td>
<td>39.11 51.28</td>
</tr>
<tr>
<td>14.30-14.45</td>
<td>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</td>
<td>44.51 57.20</td>
</tr>
<tr>
<td>14.45-15.00</td>
<td>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</td>
<td>43.29 55.87</td>
</tr>
<tr>
<td>15.00-15.15</td>
<td>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</td>
<td>44.89 57.62</td>
</tr>
<tr>
<td>15.15-15.30</td>
<td>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</td>
<td>45.32 58.09</td>
</tr>
<tr>
<td>15.30-15.45</td>
<td>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</td>
<td>45.89 58.71</td>
</tr>
</tbody>
</table>
### Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane

for ARENA Simulation Model (continued)

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr (Max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0% 1% 2% 5% 10% 20% 30% 40% 50% 60% 70% 80% 90% 95% 98% 99% 100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.45-16.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>16.00-16.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>16.15-16.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>16.30-16.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>16.45-17.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>17.00-17.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>17.15-17.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>17.30-17.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>17.45-18.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>18.00-18.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>18.15-18.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>18.30-18.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>18.45-19.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>19.00-19.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>19.15-19.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>19.30-19.45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>19.45-20.00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>20.00-20.15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>20.15-20.30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>20:30-20:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>20:45-21:00</td>
<td>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</td>
<td></td>
</tr>
</tbody>
</table>
Calculated Inter-arrival Times for 15 Minute Intervals using Adjusted Number of Vehicles per Hour for Passing Lane

for ARENA Simulation Model (continued)

<table>
<thead>
<tr>
<th>Time</th>
<th>Cumulative percentage</th>
<th>No of Veh/hr</th>
</tr>
</thead>
<tbody>
<tr>
<td>21:00-21:15</td>
<td>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 89.85 105.95 117.01 132</td>
<td>218</td>
</tr>
<tr>
<td>21:15-21:30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>21:30-21:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>21:45-22:00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>22:00-22:15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>22:15-22:30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>22:30-22:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>22:45-23:00</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>23:00-23:15</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>23:15-23:30</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>23:30-23:45</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>23:45-0:00</td>
<td>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</td>
<td></td>
</tr>
</tbody>
</table>
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(0.0001,0.2,0.01,0.1,0.1,0.02,0.1,0.2,0.05,0.1,0.3,0.07,0.5,0.1,0.9,0.2,1.5,0.3,2.3,0.4,3.5,0.5,
        5.4,0.6,8.5,0.7,12.6,0.8,18.7,0.9,29.3,0.95,39.4,0.98,50.2,0.99,57.9,1.7,1.65)),
     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


1659$ DISPOSE: No;
Model statements for module: Create 337

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

SecondtoBaseTime(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, SecondtoBaseTime(1900), Vehicles:

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

; ; Model statements for module: Create 339 ;

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$);

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

; ; Model statements for module: Decide 1691 ;

BRANCH, 1:
If,Entity.CreateTime<3700,1676$,Yes:
Else,1677$,Yes;

ASSIGN: Decide 1691.NumberOut True=Decide 1691.NumberOut True + 1:NEXT(2$);

ASSIGN: Decide 1691.NumberOut False=Decide 1691.NumberOut False + 1:NEXT(21$);

; ; Model statements for module: Create 340 ;

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$);

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

; ; Model statements for module: Decide 1692 ;

BRANCH, 1:
If,Entity.CreateTime<4600,1682$,Yes:
Else,1683$,Yes;

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, 197)), 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;
ASSIGN:  Decide 1694.NumberOut True=Decide 1694.NumberOut True + 1:NEXT(2$);

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

;  Model statements for module:  Create 343
;

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$);

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

;  Model statements for module:  Decide 1695
;

BRANCH,  1:

If,Entity.CreateTime<7300,1700$,Yes:
Else,1701$,Yes;

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

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

;  Model statements for module:  Create 344
;

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$);

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

;  Model statements for module:  Decide 1696
;

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, 67.62)),
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, 0.61, 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, Yes:
Else, Yes;

ASSIGN: Decide 1698.NumberOut True = Decide 1698.NumberOut True + 1; NEXT(2$);

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

; Model statements for module: Create 347
;

CREATE, 1, SecondstoBaseTime (10000), Vehicles:
SecondstoBaseTime (DISC(0.0001, 0.001, 0.01, 0.02, 0.03, 0.05, 0.07, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0, 2.0, 3.0, 5.0, 7.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 70.0, 80.0, 90.0, 100.0), 97; NEXT(1721$);

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

; Model statements for module: Decide 1699
;

BRANCH, 1:
If, Entity.CreateTime < 10900, Yes:
Else, Yes;

ASSIGN: Decide 1699.NumberOut True = Decide 1699.NumberOut True + 1; NEXT(2$);

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

; Model statements for module: Create 348
;

CREATE, 1, SecondstoBaseTime (10900), Vehicles:
SecondstoBaseTime (DISC(0.0001, 0.001, 0.01, 0.02, 0.03, 0.05, 0.07, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0, 2.0, 3.0, 5.0, 7.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 70.0, 80.0, 90.0, 100.0), 69; NEXT(1727$);

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

; Model statements for module: Decide 1700
;
180

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, 62)), 97:NEXT(1739$);
1739$ ASSIGN: L.Vehicle Entry1015_1030.NumberOut = L.Vehicle Entry1015_1030.NumberOut + 1:NEXT(35$);

; ; Model statements for module: Decide 1702
181

\[ \text{BRANCH, 35$: If, Entity.CreateTime < 13600, 1742$, Yes: } \\
\quad \text{Else, 1743$, Yes; } \\
\text{ASSIGN: Decide 1702.NumberOut True = Decide 1702.NumberOut True + 1; NEXT(2$); } \\
\text{ASSIGN: Decide 1702.NumberOut False = Decide 1702.NumberOut False + 1; NEXT(21$); } \\
\]

\[ \text{CREATE, 1, SecondstoBaseTime(13600), Vehicles: } \\
\text{SecondstoBaseTime(DISC(0.0001, 0.1, 0.01, 0.50, 0.02, 0.58, 0.05, 0.70, 0.1, 0.84, 0.2, 1.07, 0.3, 1.48, 0.4, 2.23, 0.5, } \\
\quad 3.35, 0.6, 5.13, 0.7, 7.88, 0.8, 11.93, 0.9, 19.18, 0.95, 27.71, 0.98, 37.94, 0.99, 45.41, 1.58, 31)), 122; NEXT(1745$); \\
\text{ASSIGN: L.Vehicle Entry1030_1045.NumberOut = L.Vehicle Entry1030_1045.NumberOut + 1; NEXT(36$); } \\
\]

\[ \text{BRANCH, 36$: If, Entity.CreateTime < 14500, 1748$, Yes: } \\
\quad \text{Else, 1749$, Yes; } \\
\text{ASSIGN: Decide 1703.NumberOut True = Decide 1703.NumberOut True + 1; NEXT(2$); } \\
\text{ASSIGN: Decide 1703.NumberOut False = Decide 1703.NumberOut False + 1; NEXT(21$); } \\
\]

\[ \text{CREATE, 1, SecondstoBaseTime(14500), Vehicles: } \\
\text{SecondstoBaseTime(DISC(0.0001, 0.1, 0.01, 0.50, 0.02, 0.59, 0.05, 0.71, 0.1, 0.86, 0.2, 1.15, 0.3, 1.62, 0.4, 2.46, 0.5, } \\
\quad 3.72, 0.6, 5.72, 0.7, 8.71, 0.8, 13.12, 0.9, 20.96, 0.95, 29.76, 0.98, 40.08, 0.99, 47.59, 1.60, 63)), 112; NEXT(1751$); \\
\text{ASSIGN: L.Vehicle Entry1045_1100.NumberOut = L.Vehicle Entry1045_1100.NumberOut + 1; NEXT(37$); } \\
\]
; 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;

ASSIGN: Decide 1706.NumberOut True = Decide 1706.NumberOut True + 1:NEXT(2$);
ASSIGN: Decide 1706.NumberOut False = Decide 1706.NumberOut False + 1:NEXT(21$);

Model statements for module: Create 355

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, 8.1, 0.5, 4.26, 0.6, 6.6, 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, 0.6)), 100:NEXT(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;

ASSIGN: Decide 1707.NumberOut True = Decide 1707.NumberOut True + 1:NEXT(2$);
ASSIGN: Decide 1707.NumberOut False = Decide 1707.NumberOut False + 1:NEXT(21$);

Model statements for module: Create 356

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, 4.2, 0.5, 3.64, 0.6, 5.6, 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, 1.4)), 114:NEXT(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(0.0001,0.1,0.01,0.5,0.02,0.57,0.05,0.68,0.1,0.81,0.2,0.97,0.3,1.30,0.4,1.94,0.5, 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(0.0001,0.1,0.01,0.5,0.02,0.58,0.05,0.69,0.1,0.83,0.2,1.02,0.3,1.39,0.4,2.08,0.5, 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(0.0001,0.1,0.01,0.5,0.02,0.59,0.05,0.72,0.1,0.88,0.2,1.22,0.3,1.74,0.4,2.66,0.5,
  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(0.0001,0.1,0.01,0.5,0.02,0.59,0.05,0.72,0.1,0.89,0.2,1.25,0.3,1.79,0.4,2.75,0.5,
  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$);
ASSIGN: L.Vehicle Entry1315_1330.NumberOut=L.Vehicle Entry1315_1330.NumberOut + 1:NEXT(47$);

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$);

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,  
  4.21,0.6,6.52,0.7,9.81,0.8,14.71,0.9,23.34,0.95,32.35,0.98,42.94,0.99,50.51,1.63,74)),  
  101:NEXT(1817$);
1817$ ASSIGN: L.Vehicle Entry1330_1345.NumberOut=L.Vehicle Entry1330_1345.NumberOut + 1:NEXT(48$);

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(0.0001,0.1,0.01,0.49,0.02,0.57,0.05,0.66,0.1,0.78,0.2,0.83,0.3,1.07,0.4,1.54,0.5,2.27,0.6,3.410.7,5.50,0.8,8.48,0.9,14.03,0.95,21.76,0.98,31.75,0.99,39.09,1.51.57)),

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:
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(0.0001,0.1,0.01,0.5,0.01,0.5,0.57,0.05,0.68,0.1,0.81,0.2,0.95,0.3,1.27,0.4,1.89,0.5,
        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)),
        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
CREATE, 1,SecondstoBaseTime(30700),Vehicles:

SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.50,0.05,0.69,0.1,0.82,0.2,1.01,0.3,1.37,0.4,2.04,0.5,
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$);

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

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

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

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

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$);

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

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

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

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

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$);

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

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

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

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, 0.5, 0, 0.02, 0.58, 0.05, 0.68, 0.1, 0.82, 0.2, 0.099, 0.31, 0.34, 0.4, 1, 0.99, 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, 0.5, 0.02, 0.59, 0.05, 0.71, 0.1, 0.86, 0.2, 1.14, 0.31, 0.61, 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
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$);

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

BRANCH, 1:
If,Entity.CreateTime<35200,1886$,Yes:
Else,1887$,Yes;

ASSIGN:  Decide 1726.NumberOut True=Decide 1726.NumberOut True + 1:NEXT(2$);
ASSIGN:  Decide 1726.NumberOut False=Decide 1726.NumberOut False + 1:NEXT(21$);

Model statements for module: Create 375

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$);

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

Model statements for module: Decide 1727

BRANCH, 1:
If,Entity.CreateTime<36100,1892$,Yes:
Else,1893$,Yes;

ASSIGN:  Decide 1727.NumberOut True=Decide 1727.NumberOut True + 1:NEXT(2$);
ASSIGN:  Decide 1727.NumberOut False=Decide 1727.NumberOut False + 1:NEXT(21$);
194

: Model statements for module: Create 376

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

SecondstoBaseTime(DIS(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(DIS(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(DIS(0.0001, 0.1, 0.01, 0.50, 0.02, 0.58, 0.05, 0.70, 0.01, 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(DIS(0.0001, 0.1, 0.01, 0.50, 0.02, 0.58, 0.05, 0.70, 0.01, 0.85, 0.2, 1.16, 0.3, 1.64, 0.4, 2.49, 0.5, 3.76, 0.6, 5, 79, 0.7, 8, 04, 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, 110));

$1931$ 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$));

$1937$ 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$);
1942$ 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$);

1944$ 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$);

1950$ ASSIGN: Decide 1736.NumberOut True=Decide 1736.NumberOut True + 1:NEXT(2$);

1951$ ASSIGN: Decide 1736.NumberOut False=Decide 1736.NumberOut False + 1:NEXT(21$);

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$);


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

; ; 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$);


; ; 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(0.0001,0.1,0.01,0.52,0.02,0.63,0.05,0.78,0.1,1,0.02,0.2,1.75,0.3,2.65,0.4,4.17,0.5, 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$);


; ; Model statements for module: Decide 1741
;
74$ BRANCH, 1:
If,Entity.CreateTime<48700,1976$,Yes:
Else,1977$,Yes;
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$);
1:NEXT(76$);

;  
;  
;   Model statements for module:  Decide 1742
;  
76$    BRANCH,    1:
   If,Entity.CreateTime<1000,1988$,Yes:
   Else,1989$,Yes;
202

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$);


;
;
; 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(1000),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$);

Model statements for module: Decide 1744

79$ BRANCH, 1:
    If,Entity.CreateTime<2800,2001$, Yes:
        Decide 1744.NumberOut True = Decide 1744.NumberOut True + 1:NEXT(3$);
    Else,2002$, Yes:
        Decide 1744.NumberOut False = Decide 1744.NumberOut False + 1:NEXT(77$);

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$);

Model statements for module: Decide 1745

80$ BRANCH, 1:
    If,Entity.CreateTime<3700,2007$, Yes:
        Decide 1745.NumberOut True = Decide 1745.NumberOut True + 1:NEXT(3$);
    Else,2008$, Yes:
        Decide 1745.NumberOut False = Decide 1745.NumberOut False + 1:NEXT(77$);

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$);
ASSIGN: R.Vehicle Enters815_830.NumberOut=R.Vehicle Enters815_830.NumberOut + 1:NEXT(83$);

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$);

CREATE, 1,SecondstoBaseTime(6400),Vehicles:
SecondstoBaseTime(DISC(0.0001,0.1,0.1,0.70,0.02,0.87,0.05,1.14,0.1,1.152,0.2,2.15,0.3,2.80,0.4,3.58,0.5, 4.54,0.6,5.74,0,7.20,0.8,9.23,0.9,12.79,0.95,16.25,0.98,20.20,0.99,22.80,1.29.41));

ASSIGN: R.Vehicle Enters830_845.NumberOut=R.Vehicle Enters830_845.NumberOut + 1:NEXT(84$);

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$);

CREATE, 1,SecondstoBaseTime(7300),Vehicles:
SecondstoBaseTime(DISC(0.0001,0.1,0.1,0.70,0.02,0.87,0.05,1.14,0.1,1.152,0.2,2.15,0.3,2.80,0.4,3.58,0.5, 4.54,0.6,5.74,0,7.20,0.8,9.23,0.9,12.79,0.95,16.25,0.98,20.20,0.99,22.80,1.29.41));
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
; ;
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$);

ASSIGN: R.Vehicle Enters945_1000.NumberOut=R.Vehicle Enters945_1000.NumberOut + 1:NEXT(89$);

Model statements for module: Decide 1754

BRANCH, 1:
If,Entity.CreateTime<11800,2061$, Yes:
Else,2062$, Yes;

ASSIGN: Decide 1754.NumberOut True=Decide 1754.NumberOut True + 1:NEXT(3$);

ASSIGN: Decide 1754.NumberOut False=Decide 1754.NumberOut False + 1:NEXT(77$);

Model statements for module: Create 403

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$);

ASSIGN: R.Vehicle Enters1000_1015.NumberOut=R.Vehicle Enters1000_1015.NumberOut + 1:NEXT(90$);

Model statements for module: Decide 1755

BRANCH, 1:
If,Entity.CreateTime<12700,2067$, Yes:
Else,2068$, Yes;

ASSIGN: Decide 1755.NumberOut True=Decide 1755.NumberOut True + 1:NEXT(3$);

ASSIGN: Decide 1755.NumberOut False=Decide 1755.NumberOut False + 1:NEXT(77$);

Model statements for module: Create 404
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$);
ASSIGN: R.Vehicle Enters1015_1030.NumberOut=R.Vehicle Enters1015_1030.NumberOut + 1:NEXT(91$);
Model statements for module: Decide 1756
BRANCH, 1:
If, Entity.CreateTime< 13600, 2073$, Yes:
Else, 2074$, Yes;
ASSIGN: Decide 1756.NumberOut True=Decide 1756.NumberOut True + 1:NEXT(3$);
ASSIGN: Decide 1756.NumberOut False=Decide 1756.NumberOut False + 1:NEXT(77$);
Model statements for module: Create 405
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, 6.0, 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$);
ASSIGN: R.Vehicle Enters1030_1045.NumberOut=R.Vehicle Enters1030_1045.NumberOut + 1:NEXT(92$);
Model statements for module: Decide 1757
BRANCH, 1:
If, Entity.CreateTime< 14500, 2079$, Yes:
Else, 2080$, Yes;
ASSIGN: Decide 1757.NumberOut True=Decide 1757.NumberOut True + 1:NEXT(3$);
ASSIGN: Decide 1757.NumberOut False=Decide 1757.NumberOut False + 1:NEXT(77$);
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,
3.98,0.6,4.96,0.7,6.16,0.8,7.93,0.9,10.96,0.95,13.64,0.98,17.29,0.99,19.70,1.26,1.5)),
170:NEXT(2082$);


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$);

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$);

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

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, 2.2, 12.0, 3.2, 76.0, 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$);

ASSIGN: R.Vehicle Enters1115_1130.NumberOut=R.Vehicle Enters1115_1130.NumberOut + 1:NEXT(95$);

Model statements for module: Decide 1760

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

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$);

ASSIGN: R.Vehicle Enters1130_1145.NumberOut=R.Vehicle Enters1130_1145.NumberOut + 1:NEXT(96$);

Model statements for module: Decide 1761

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$);
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$);

ASSIGN: R.Vehicle Enters1145_1200.NumberOut=R.Vehicle Enters1145_1200.NumberOut + 1:NEXT(97$);

BRANCH, 1:
If,Entity.CreateTime<19000,2109$,Yes:
Else,2110$,Yes;

ASSIGN: Decide 1762.NumberOut True=Decide 1762.NumberOut True + 1:NEXT(3$);
ASSIGN: Decide 1762.NumberOut False=Decide 1762.NumberOut False + 1:NEXT(77$);

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$);

ASSIGN: R.Vehicle Enters1200_1215.NumberOut=R.Vehicle Enters1200_1215.NumberOut + 1:NEXT(98$);

BRANCH, 1:
If,Entity.CreateTime<19900,2115$,Yes:
Else,2116$,Yes;

ASSIGN: Decide 1763.NumberOut True=Decide 1763.NumberOut True + 1:NEXT(3$);
ASSIGN: Decide 1763.NumberOut False=Decide 1763.NumberOut False + 1:NEXT(77$);
Model statements for module: Create 412

2117$ CREATE, 1,SecondstoBaseTime(19900),Vehicles:
SecondstoBaseTime(DIS(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(DIS(0.0001,0.1,0.01,0.67,0.02,0.81,0.05,1.10,0.1,1.37,0.2,1.88,0.3,2.41,0.4,3.06,0.5,3.83,0.6,4.75,0.7,5.88,0.8,7.57,0.9,10.46,0.95,12.92,0.98,16.50,0.99,18.85,1.25,26)), 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, 1.50)), 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, 2.60)), 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$);
$\text{CREATE, 1,SecondstoBaseTime(23500),Vehicles:}$

SecondstoBaseTime(DISC(0.0001,0.10,0.01,0.67,0.02,0.83,0.05,1.10,0.10,1.40,0.21,0.93,0.32,2.50,0.43,1.70,0.53,3.98,0.64,9.60,7.61,6.00,8.79,0.90,1.90,9.95,13.64,0.98,17.29,0.99,19.70,1.26,15))


$\text{BRANCH, 1:}$

If,Entity.CreateTime<24400,2145$,Yes:
Else,2146$,Yes;

$\text{ASSIGN: Decide 1768.NumberOut True=Decide 1768.NumberOut True + 1:NEXT(3$});$

$\text{ASSIGN: Decide 1768.NumberOut False=Decide 1768.NumberOut False + 1:NEXT(77$});$

$\text{CREATE, 1,SecondstoBaseTime(24400),Vehicles:}$

SecondstoBaseTime(DISC(0.0001,0.10,0.01,0.70,0.02,0.87,0.05,1.14,0.11,1.52,0.22,1.55,0.32,2.80,0.43,3.58,0.54,4.54,0.65,74,0.77,20.08,9.23,0.91,12.79,0.95,16.25,0.98,20.09,22.80,1.29,41))


$\text{BRANCH, 1:}$

If,Entity.CreateTime<25300,2151$,Yes:
Else,2152$,Yes;

$\text{ASSIGN: Decide 1769.NumberOut True=Decide 1769.NumberOut True + 1:NEXT(3$});$
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(0.0001,0.1,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.38,0.2,1.89,0.3,2.43,0.4,3.09,0.5,
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(0.0001,0.1,0.1,0.01,0.67,0.02,0.82,0.05,1.09,0.1,1.39,0.2,1.92,0.3,2.48,0.4,3.14,0.5,
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$);
ASSIGN: Decide 1771.NumberOut False=Decide 1771.NumberOut False + 1:NEXT(77$);

CREATE, 1,SecondstoBaseTime(27100),Vehicles:
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.66,0.02,0.81,0.05,1,0.08,0.1,1.36,0.2,1.85,0.3,2.38,0.4,3.02,0.5, 
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$);


CREATE, 1,SecondstoBaseTime(28000),Vehicles:
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1,0.09,0.1,1.38,0.2,1.90,0.3,2.44,0.4,3.10,0.5, 
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$);


CREATE, 1,SecondstoBaseTime(28000),Vehicles:
SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.67,0.02,0.82,0.05,1,0.09,0.1,1.38,0.2,1.90,0.3,2.44,0.4,3.10,0.5, 
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$);

ASSIGN: Decide 1773.NumberOut True=Decide 1773.NumberOut True + 1:NEXT(3$);
ASSIGN: Decide 1773.NumberOut False=Decide 1773.NumberOut False + 1:NEXT(77$);

; Model statements for module: Create 422
;

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$);

ASSIGN: R.Vehicle Enters1445_1500.NumberOut=R.Vehicle Enters1445_1500.NumberOut + 1:NEXT(109$);

; Model statements for module: Decide 1774
;
BRANCH, 1: If,Entity.CreateTime<29800,2181$,Yes:
Else,2182$,Yes:
ASSIGN: Decide 1774.NumberOut True=Decide 1774.NumberOut True + 1:NEXT(3$);
ASSIGN: Decide 1774.NumberOut False=Decide 1774.NumberOut False + 1:NEXT(77$);

; Model statements for module: Create 423
;
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$);

ASSIGN: R.Vehicle Enters1500_1515.NumberOut=R.Vehicle Enters1500_1515.NumberOut + 1:NEXT(110$);

; Model statements for module: Decide 1775
;
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$);

; ; 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(DIS(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, 4.5)), 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(DIS(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, 9.9, 84, 0.95, 12.24, 0.98, 15.77, 0.99, 18.08, 1.24, 16)), 189:NEXT(2220$);

; ; 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(DIS(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(DIS(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$\begin{align*}
&\text{BRANCH, 1:} \\
&\quad \text{If,Entity.CreateTime<39700,2247$,Yes:} \\
&\quad \text{Else,2248$,Yes;} \\
&2247$\quad \text{ASSIGN: Decide 1785.NumberOut True=Decide 1785.NumberOut True + 1:NEXT(3$);} \\
&2248$\quad \text{ASSIGN: Decide 1785.NumberOut False=Decide 1785.NumberOut False + 1:NEXT(77$);}
\end{align*}

Model statements for module: Create 434

2249$\begin{align*}
&\text{CREATE, 1,SecondstoBaseTime(39700),Vehicles:} \\
&\quad \text{SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.70,0.02,0.86,0.05,1.13,0.1,1.48,0.2,2.09,0.3,2.71,0.4,3.47,0.5,} \\
&\quad \quad \quad \quad \quad 4.38,0.6,5.52,0.7,6.91,0.8,8.87,0.9,12.28,0.95,15.52,0.98,19.39,0.99,21.93,12.850)),} \\
&\quad \quad \quad \quad \quad 152:\text{NEXT(2250$);} \\
\end{align*}

Model statements for module: Decide 1786

121$\begin{align*}
&\text{BRANCH, 1:} \\
&\quad \text{If,Entity.CreateTime<40600,2253$,Yes:} \\
&\quad \text{Else,2254$,Yes;} \\
&2253$\quad \text{ASSIGN: Decide 1786.NumberOut True=Decide 1786.NumberOut True + 1:NEXT(3$);} \\
&2254$\quad \text{ASSIGN: Decide 1786.NumberOut False=Decide 1786.NumberOut False + 1:NEXT(77$);}
\end{align*}

Model statements for module: Create 435

2255$\begin{align*}
&\text{CREATE, 1,SecondstoBaseTime(40600),Vehicles:} \\
&\quad \text{SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.69,0.02,0.85,0.05,1.12,0.1,1.45,0.2,2.03,0.3,2.62,0.4,3.34,0.5,} \\
&\quad \quad \quad \quad \quad 4.22,0.6,5.29,0.7,6.60,0.8,8.48,0.9,11.73,0.95,14.74,0.98,18.52,0.99,21.00,12.752)),} \\
&\quad \quad \quad \quad \quad 159:\text{NEXT(2256$);} \\
&2256$\quad \text{ASSIGN: R.Vehicle Enters1800_1815.NumberOut=R.Vehicle Enters1800_1815.NumberOut + 1:NEXT(122$);}
\end{align*}
Model statements for module: Decide 1787

122$ \text{BRANCH, 1:}
\quad \text{If,Entity.CreateTime<41500,2259$,Yes:}
\quad \text{Else,2260$,Yes;}
2259$ \text{ASSIGN: } \text{Decide 1787.NumberOut True=} \text{Decide 1787.NumberOut True + 1:NEXT(3$)};
2260$ \text{ASSIGN: } \text{Decide 1787.NumberOut False=} \text{Decide 1787.NumberOut False + 1:NEXT(77$)};

Model statements for module: Create 436

2261$ \text{CREATE, 1,SecondstoBaseTime(41500),Vehicles:}
\text{SecondstoBaseTime(DISC(0.0001,0.1,0.01,0.71,0.02,0.88,0.05,1.16,0.1,1.54,0.2,2.2,0.3,2.88,0.4,3.68,0.5,}
\text{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))},
\text{141:NEXT(2262$)};
2262$ \text{ASSIGN: } \text{R.Vehicle Enters1815_1830.NumberOut=} \text{R.Vehicle Enters1815_1830.NumberOut + 1:NEXT(123$)};

Model statements for module: Decide 1788

123$ \text{BRANCH, 1:}
\quad \text{If,Entity.CreateTime<42400,2265$,Yes:}
\quad \text{Else,2266$,Yes;}
2265$ \text{ASSIGN: } \text{Decide 1788.NumberOut True=} \text{Decide 1788.NumberOut True + 1:NEXT(3$)};
2266$ \text{ASSIGN: } \text{Decide 1788.NumberOut False=} \text{Decide 1788.NumberOut False + 1:NEXT(77$)};

Model statements for module: Create 437

2267$ \text{CREATE, 1,SecondstoBaseTime(42400),Vehicles:}
\text{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,}
\text{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))},
\text{144:NEXT(2268$)};
2268$ \text{ASSIGN: } \text{R.Vehicle Enters1830_1845.NumberOut=} \text{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, Yes:
    Else, 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$);

Model statements for module: Decide 1792

127$ BRANCH, 1:
    If, Entity.CreateTime<46000, Yes:
    Else, 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, 4.45, 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$);

; ; 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(0.0001,0.1,0.01,0.75,0.02,0.94,0.05,1.22,0.1,1.69,0.2,2.48,0.3,3.3,2.6,0.4,4.4,2.1,0.5,
5.4,0.6,6.92,0.7,8.79,0.8,11.22,0.9,15.61,0.95,20.27,0.98,24.66,0.99,27.55,1.34,40)),
120:NEXT(2298$);

; ; 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(0.0001,0.1,0.01,0.76,0.02,0.95,0.05,1.23,0.1,1.72,0.2,2.54,0.3,3.3,5,0.4,4.4,3.3,0.5,
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$);

; ; 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(0.0001,0.1,0.01,0.71,0.02,0.88,0.05,1.16,0.1,1.54,0.2,2.2,0.3,2.88,0.4,3.68,0.5,4.68,0.6,5.93,0.7,7.46,0.8,9.56,0.9,13.25,0.95,16.91,0.98,20.93,0.99,23.57,1,30.23)),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 1,Station6450;
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$);
ASSIGN: CheckGapL6450Speed10.NumberOut
False=CheckGapL6450Speed10.NumberOut False + 1:NEXT(182$);

TRANSPORT: LAGV,RStation6700;

TRANSPORT: LAGV,LStation6700;

;  ;
;  ;  ; Model statements for module: Decide 775
;  ;

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,
    ASSIGN: CheckGapL6450Speed75.NumberOut
    True=CheckGapL6450Speed75.NumberOut True + 1:NEXT(159$);
  ELSE, ASSIGN: CheckGapL6450Speed75.NumberOut
    False=CheckGapL6450Speed75.NumberOut False + 1:NEXT(160$);

TRANSPORT: LAGV,RStation6700;

TRANSPORT: LAGV,LStation6700;

;  ;
;  ;  ; Model statements for module: Decide 849
;  ;

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$,
    ASSIGN: CheckGapL6450Speed55_75.NumberOut
    True=CheckGapL6450Speed55_75.NumberOut True + 1:NEXT(172$);
  ELSE, ASSIGN: CheckGapL6450Speed55_75.NumberOut
    False=CheckGapL6450Speed55_75.NumberOut False + 1:NEXT(173$);

TRANSPORT: LAGV,RStation6700;

TRANSPORT: LAGV,LStation6700;

;  ;
;  ;  ; Model statements for module: Decide 850
;  ;
232

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: LAGV,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;
ASSIGN: CheckGapL6450Speed10less.NumberOut
   True=CheckGapL6450Speed10less.NumberOut True + 1:NEXT(197$);

ASSIGN: CheckGapL6450Speed10less.NumberOut
   False=CheckGapL6450Speed10less.NumberOut False + 1:NEXT(198$);

TRANSPORT: LAGV,RStation6700;

TRANSPORT: LAGV,LStation6700;

; ;
; Model statements for module: Decide 854
;
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;

ASSIGN: CheckGapL6450Speed75less.NumberOut
   True=CheckGapL6450Speed75less.NumberOut True + 1:NEXT(185$);

ASSIGN: CheckGapL6450Speed75less.NumberOut
   False=CheckGapL6450Speed75less.NumberOut False + 1:NEXT(186$);

TRANSPORT: LAGV,RStation6700;

TRANSPORT: LAGV,LStation6700;

; ;
; Model statements for module: Decide 855
;
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;

ASSIGN: CheckGapL6450Speed55_75less.NumberOut
   True=CheckGapL6450Speed55_75less.NumberOut True + 1
   ; NEXT(188$);

ASSIGN: CheckGapL6450Speed55_75less.NumberOut
   False=CheckGapL6450Speed55_75less.NumberOut False + 1
   ; NEXT(189$);

TRANSPORT: LAGV,RStation6700;

TRANSPORT: LAGV,LStation6700;
Model statements for module: Decide 856

BRANCH, 1:

If,NZ(RL2,25) &&NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) == 0, Yes:

ASSIGN: CheckGapL6450Speed35_55less.NumberOut True=CheckGapL6450Speed35_55less.NumberOut True + 1

ELSE, Yes;

ASSIGN: CheckGapL6450Speed35_55less.NumberOut False=CheckGapL6450Speed35_55less.NumberOut False + 1

NEXT(191$);

Model statements for module: Decide 857

BRANCH, 1:

If,NZ(RL2,26) &&NZ(RL2,27) &&NZ(RL2,28) && NZ(RL3,1) == 0, Yes:

ASSIGN: CheckGapL6450Speed10_35less.NumberOut True=CheckGapL6450Speed10_35less.NumberOut True + 1

ELSE, Yes;

ASSIGN: CheckGapL6450Speed10_35less.NumberOut False=CheckGapL6450Speed10_35less.NumberOut False + 1

NEXT(194$);

Model statements for module: Assign 221

ASSIGN: VTU(LAGVT,LTruck#)=NORM(83,4):NEXT(156$);

Model statements for module: Decide 748
235$ \textbf{BRANCH, 1:} \\
\quad \textbf{If,} \text{SpeedatRStation5750} \geq \text{VTU(LAGVT,LTruck#)}, 2346$, \textbf{Yes:} \\
\quad \quad \text{Else,} 2347$, \textbf{Yes;} \\
2346$ \textbf{ASSIGN:} \quad \text{Decide 748.NumberOut True=} \text{Decide 748.NumberOut True} + 1; \text{NEXT(199$)}; \\
2347$ \textbf{ASSIGN:} \quad \text{Decide 748.NumberOut False=} \text{Decide 748.NumberOut False} + 1; \text{NEXT(205$)}; \\

; \\
; \\
Model statements for module: Decide 860 \\
;

199$ \textbf{BRANCH, 1:} \\
\quad \textbf{If,} \text{SpeedatRStation5750} > 75, 200$, \textbf{Yes:} \\
\quad \quad \textbf{If,} \text{SpeedatRStation5750} > 55 \quad \text{&&} \quad \text{SpeedatRStation5750} \leq 75, 201$, \textbf{Yes:} \\
\quad \quad \quad \textbf{If,} \text{SpeedatRStation5750} > 35 \quad \text{&&} \quad \text{SpeedatRStation5750} \leq 55, 202$, \textbf{Yes:} \\
\quad \quad \quad \quad \textbf{If,} \text{SpeedatRStation5750} > 10 \quad \text{&&} \quad \text{SpeedatRStation5750} \leq 35, 203$, \textbf{Yes:} \\
\quad \quad \quad \quad \quad \text{Else,} 204$, \textbf{Yes;} \\
204$ \textbf{ASSIGN:} \quad \text{CheckGapL6450Speed10t.NumberOut True=} \text{CheckGapL6450Speed10t.NumberOut True} + 1; \text{NEXT(169$)}; \\
205$ \textbf{ASSIGN:} \quad \text{CheckGapL6450Speed10t.NumberOut False=} \text{CheckGapL6450Speed10t.NumberOut False} + 1; \text{NEXT(170$)}; \\
169$ \textbf{TRANSPORT:} \quad \text{LAGVT,RStation6700;} \\
170$ \textbf{TRANSPORT:} \quad \text{LAGVT,LStation6700;} \\

; \\
; \\
Model statements for module: Decide 865 \\
;

204$ \textbf{BRANCH, 1:} \\
\quad \textbf{If,} \text{NZ(RL3,1) && NZ(RL3,2) && NZ(RL3,3) && NZ(RL3,4) && NZ(RL2,28) == 0, 2350$, \textbf{Yes:} \\
\quad \quad \quad \text{Else,} 2351$, \textbf{Yes;} \\
2350$ \textbf{ASSIGN:} \quad \text{CheckGapL6450Speed75t.NumberOut True=} \text{CheckGapL6450Speed75t.NumberOut True} + 1; \text{NEXT(161$)}; \\
2351$ \textbf{ASSIGN:} \quad \text{CheckGapL6450Speed75t.NumberOut False=} \text{CheckGapL6450Speed75t.NumberOut False} + 1; \text{NEXT(170$)}; \\
169$ \textbf{TRANSPORT:} \quad \text{LAGVT,RStation6700;} \\
170$ \textbf{TRANSPORT:} \quad \text{LAGVT,LStation6700;} \\

; \\
; \\
Model statements for module: Decide 861 \\
;

200$ \textbf{BRANCH, 1:} \\
\quad \textbf{If,} \\
\quad \quad \text{NZ(RL3,1) && NZ(RL3,2) && NZ(RL3,3) && NZ(RL3,4) && NZ(RL3,5) \\
\quad \quad \quad && NZ(RL3,6) && NZ(RL3,7) && NZ(RL3,8) && NZ(RL3,9) \quad \text{&&} \quad \text{NZ(RL2,28) \quad \text{&&} \quad \text{NZ(RL2,27) == 0,} \\
\quad \quad \quad \quad \quad \text{2352$, \textbf{Yes:} \\
\quad \quad \quad \quad \quad \quad \text{Else,} 2353$, \textbf{Yes;} \\
2352$ \textbf{ASSIGN:} \quad \text{CheckGapL6450Speed75t.NumberOut True=} \text{CheckGapL6450Speed75t.NumberOut True} + 1; \text{NEXT(161$)}; \\
2353$ \textbf{ASSIGN:} \quad \text{CheckGapL6450Speed75t.NumberOut False=} \text{CheckGapL6450Speed75t.NumberOut False} + 1; \text{NEXT(162$)};
TRANSPORT: LAGVT,RStation6700;

TRANSPORT: LAGVT,LStation6700;

; ; Model statements for module: Decide 862 ;

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;

ASSIGN: CheckGapL6450Speed55_75t.NumberOut
True=CheckGapL6450Speed55_75t.NumberOut True + 1:NEXT(164$);

ASSIGN: CheckGapL6450Speed55_75t.NumberOut
False=CheckGapL6450Speed55_75t.NumberOut False + 1:NEXT(163$);

TRANSPORT: LAGVT,RStation6700;

TRANSPORT: LAGVT,LStation6700;

; ; Model statements for module: Decide 863 ;

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,

ASSIGN: CheckGapL6450Speed35_55t.NumberOut
True=CheckGapL6450Speed35_55t.NumberOut True + 1:NEXT(165$);

ASSIGN: CheckGapL6450Speed35_55t.NumberOut
False=CheckGapL6450Speed35_55t.NumberOut False + 1:NEXT(166$);

TRANSPORT: LAGVT,RStation6700;

TRANSPORT: LAGVT,LStation6700;

; ; Model statements for module: Decide 864 ;

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(RL2,20) && 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: CheckGapL6450Speed75less.NumberOut
True=CheckGapL6450Speed75less.NumberOut True + 1:NEXT(211$);
2365$ ASSIGN: CheckGapL6450Speed75less.NumberOut
False=CheckGapL6450Speed75less.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_75less.NumberOut
True=CheckGapL6450Speed55_75less.NumberOut True + 1
:NEXT(213$);
2367$ ASSIGN: CheckGapL6450Speed55_75less.NumberOut
False=CheckGapL6450Speed55_75less.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_55less.NumberOut
True=CheckGapL6450Speed35_55less.NumberOut True + 1
:NEXT(215$);
2369$ ASSIGN: CheckGapL6450Speed35_55less.NumberOut
False=CheckGapL6450Speed35_55less.NumberOut False + 1
:NEXT(216$);
TRANSPORT:  LAGVT,RStation6700;
TRANSPORT:  LAGVT,LStation6700;

TRANSPORT:  LAGVT,RStation6700;
TRANSPORT:  LAGVT,LStation6700;

Model statements for module: Decide 870

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;
ASSIGN:  CheckGapL6450Speed10_35lesst.NumberOut
True=CheckGapL6450Speed10_35lesst.NumberOut True + 1
:NEXT(217$);
ASSIGN:  CheckGapL6450Speed10_35lesst.NumberOut
False=CheckGapL6450Speed10_35lesst.NumberOut False + 1
:NEXT(218$);
TRANSPORT:  LAGVT,RStation6700;
TRANSPORT:  LAGVT,LStation6700;

Model statements for module: Station 41

STATION, RStation6450;
DELAY:  0.0,VA:NEXT(148$);

Model statements for module: Decide 719

BRANCH, 1:
If,Entity.Type==RCar,2375$,Yes:
Else,2376$,Yes;
ASSIGN:  Decide1VehicleTypeR6450.NumberOut
True=Decide1VehicleTypeR6450.NumberOut True + 1:NEXT(290$);
ASSIGN:  Decide1VehicleTypeR6450.NumberOut
False=Decide1VehicleTypeR6450.NumberOut False + 1:NEXT(291$);

Model statements for module: Assign 222
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,RStation 6700;

; ; 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$);
221$ 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$);

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$);

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$, Yes;
247$ 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,
    Yes:
    Else, 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$,
    Yes:
    Else, Yes;
2392$ 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$,
    Yes:
    Else,2394$ 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

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,

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   Yes:

   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:
    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$);
If Speed at R Station 6450 > 10 && Speed at R Station 6450 <= 35, 270$, 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, 0, 2414$, Yes:
Else, 2415$, Yes;
2414$ ASSIGN: Check Gap L 6700 Speed 10t.NumberOut
True = Check Gap L 6700 Speed 10t.NumberOut True + 1:NEXT(236$);
2415$ ASSIGN: Check Gap L 6700 Speed 10t.NumberOut
False = Check Gap L 6700 Speed 10t.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: Check Gap L 6700 Speed 75t.NumberOut
True = Check Gap L 6700 Speed 75t.NumberOut True + 1:NEXT(228$);
2417$ ASSIGN: Check Gap L 6700 Speed 75t.NumberOut
False = Check Gap L 6700 Speed 75t.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;
ASSIGN: CheckGapL6700Speed55_75t.NumberOut
True=CheckGapL6700Speed55_75t.NumberOut True + 1:NEXT(231$);

ASSIGN: CheckGapL6700Speed55_75t.NumberOut
False=CheckGapL6700Speed55_75t.NumberOut False + 1:NEXT(230$);

TRANSPORT: LAGVT,RStation6950;

TRANSPORT: LAGVT,LStation6950;

Model statements for module: Decide 890

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,
Yes:
ASSIGN: CheckGapL6700Speed35_55t.NumberOut
True=CheckGapL6700Speed35_55t.NumberOut True + 1:NEXT(232$);

ASSIGN: CheckGapL6700Speed35_55t.NumberOut
False=CheckGapL6700Speed35_55t.NumberOut False + 1:NEXT(233$);

TRANSPORT: LAGVT,RStation6950;

TRANSPORT: LAGVT,LStation6950;

Model statements for module: Decide 891

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;

ASSIGN: CheckGapL6700Speed10_35t.NumberOut
True=CheckGapL6700Speed10_35t.NumberOut True + 1:NEXT(234$);

ASSIGN: CheckGapL6700Speed10_35t.NumberOut
False=CheckGapL6700Speed10_35t.NumberOut False + 1:NEXT(235$);

TRANSPORT: LAGVT,RStation6950;

TRANSPORT: LAGVT,LStation6950;
Model statements for module: Decide 893

272$ BRANCH 1:
If, SpeedatRStation6450 > 75, Yes;
If, SpeedatRStation6450 > 55 && SpeedatRStation6450 <= 75, Yes;
If, SpeedatRStation6450 > 35 && SpeedatRStation6450 <= 55, Yes;
If, SpeedatRStation6450 > 10 && SpeedatRStation6450 <= 35, Yes;
Else, 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, Yes;
Else, Yes;

2426$ ASSIGN: CheckGapL6700Speed10lesst.NumberOut True = CheckGapL6700Speed10lesst.NumberOut True + 1; NEXT(286$);

2427$ ASSIGN: CheckGapL6700Speed10lesst.NumberOut False = CheckGapL6700Speed10lesst.NumberOut False + 1; NEXT(278$);

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, Yes;
Else, 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
$249$

274$\text{BRANCH, 1:}$

\text{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$Yes; \text{Else,} 2431$Yes;$

2430$\text{ASSIGN: CheckGapL6700Speed55_75less.NumberOut}$

True=CheckGapL6700Speed55_75less.NumberOut True + 1  
:NEXT(280$);$

2431$\text{ASSIGN: CheckGapL6700Speed55_75less.NumberOut}$

False=CheckGapL6700Speed55_75less.NumberOut False + 1  
:NEXT(281$);$

280$\text{TRANSPORT: LAGVT,RStation6950;}$

281$\text{TRANSPORT: LAGVT,LStation6950;}$

$; ;$

; Model statements for module: Decide 896 ;

275$\text{BRANCH, 1:}$

\text{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$Yes; \text{Else,} 2433$Yes;$

2432$\text{ASSIGN: CheckGapL6700Speed35_55less.NumberOut}$

True=CheckGapL6700Speed35_55less.NumberOut True + 1  
:NEXT(282$);$

2433$\text{ASSIGN: CheckGapL6700Speed35_55less.NumberOut}$

False=CheckGapL6700Speed35_55less.NumberOut False + 1  
:NEXT(283$);$

282$\text{TRANSPORT: LAGVT,RStation6950;}$

283$\text{TRANSPORT: LAGVT,LStation6950;}$

$; ;$

; Model statements for module: Decide 897 ;

276$\text{BRANCH, 1:}$

\text{If,}$
\NZ(RL3,6) \&\& \NZ(RL3,7) \&\& \NZ(RL3,8) \&\& \NZ(RL3,9) \&\& \NZ(RL3,10) \&\& \\
NZ(RL4,1) == 0, 2434$Yes; \text{Else,} 2435$Yes;$

2434$\text{ASSIGN: CheckGapL6700Speed10_35less.NumberOut}$

True=CheckGapL6700Speed10_35less.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

STATION, RStation6950;
DELAY: 0.0,VA:NEXT(341$);

Model statements for module: Decide 723

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

ASSIGN: SpeedatRStation6950=VTU(LAGVT,LTruck#):NEXT(305$);

Model statements for module: Decide 541

BRANCH, 1:
If,Entity.Type==RCar,2452$,Yes:
Else,2453$,Yes;
ASSIGN: Decide 541.NumberOut True=Decide 541.NumberOut True + 1:NEXT(304$);
ASSIGN: Decide 541.NumberOut False=Decide 541.NumberOut False + 1:NEXT(307$);
TRANSPORT: RAGV,RStation7050;

Model statements for module: Decide 542

BRANCH, 1:
If,Entity.Type==RTruck,2454$,Yes:
Else,2455$,Yes;
ASSIGN: Decide 542.NumberOut True=Decide 542.NumberOut True + 1:NEXT(306$);
ASSIGN: Decide 542.NumberOut False=Decide 542.NumberOut False + 1:NEXT(308$);
TRANSPORT: RAGVT,RStation7050;
Model statements for module: Decide 543

308$ \text{BRANCH, 1:}
   \text{If, Entity.Type} == \text{LCar, 2456$, Yes:}
   \text{ Else, 2457$, Yes;}
2456$ \text{ASSIGN: Decide 543.NumberOut True} = \text{Decide 543.NumberOut True} + 1; \text{NEXT(309$);}
2457$ \text{ASSIGN: Decide 543.NumberOut False} = \text{Decide 543.NumberOut False} + 1; \text{NEXT(310$);}
309$ \text{TRANSPORT: LAGV, RStation7050;}
310$ \text{TRANSPORT: LAGVT, RStation7050;}

Model statements for module: Assign 75

337$ \text{ASSIGN: Speed at RStation6950} = \text{VTU(RAGV, RCar#)}; \text{NEXT(305$);}

Model statements for module: Assign 76

338$ \text{ASSIGN: Speed at RStation6950} = \text{VTU(RAGVT, RTruck#)}; \text{NEXT(305$);}

Model statements for module: Assign 77

339$ \text{ASSIGN: Speed at RStation6950} = \text{VTU(LAGV, LCar#)}; \text{NEXT(305$);}

Model statements for module: Station 47

293$ \text{STATION, RStation7050;}
2460$ \text{DELAY: 0.0, VA; NEXT(336$);}

Model statements for module: Decide 722

336$ \text{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: RAGV, 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;
TRANSPORT: LAGVT, RStation7150;

ASSIGN: SpeedatRStation7050 = VTU(RAGV, RCar#): NEXT(312$);

ASSIGN: SpeedatRStation7050 = VTU(RAGVT, RTruck#): NEXT(312$);

ASSIGN: SpeedatRStation7050 = VTU(LAGV, LCar#): NEXT(312$);

STATION, RStation7150;
DELAY: 0.0, VA: NEXT(346$);

BRANCH, 1:
  If, Entity.Type == RCar, 342$, Yes:
  If, Entity.Type == RTruck, 343$, Yes:
  If, Entity.Type == LCar, 344$, Yes:
  Else, 345$, Yes;

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: SpeedatRSta
tion7150=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

;
258

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==L.Car, 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==L.Car, 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;

ASSIGN: Decide 900.NumberOut True = Decide 900.NumberOut True + 1; NEXT(361$);

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;

ASSIGN: CheckGapL6950Speed10.NumberOut True = CheckGapL6950Speed10.NumberOut True + 1; NEXT(385$);

ASSIGN: CheckGapL6950Speed10.NumberOut False = CheckGapL6950Speed10.NumberOut False + 1; NEXT(386$);

TRANSPORT: LAGV, RStation7050;

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$ \text{BRANCH, 1:} \\
\quad \text{If,} NZ(\text{RL5}, 1) \&\& NZ(\text{RL5}, 2) \&\& NZ(\text{RL5}, 3) \&\& \text{NZ(RL4, 10)} == 0.2519$, Yes: \\
\quad \text{Else,} 2520$, Yes; \\
2519$ \text{ASSIGN: } \text{CheckGapL6950Speed10}_35.\text{NumberOut} \\
\text{True}=\text{CheckGapL6950Speed10}_35.\text{NumberOut True} + 1; \text{NEXT}(382$); \\
2520$ \text{ASSIGN: } \text{CheckGapL6950Speed10}_35.\text{NumberOut} \\
\text{False}=\text{CheckGapL6950Speed10}_35.\text{NumberOut False} + 1; \text{NEXT}(383$); \\
382$ \text{TRANSPORT: LA GV, RStation7050; } \\
383$ \text{TRANSPORT: LAGV, LStation7050; } \\
387$ \text{BRANCH, 1:} \\
\quad \text{If,} \text{SpeedatRStation6700} > 75, 388$, Yes: \\
\quad \text{If,} \text{SpeedatRStation6700} > 55 \&\& \text{SpeedatRStation6700} \leq 75, 391$, Yes: \\
\quad \text{If,} \text{SpeedatRStation6700} > 35 \&\& \text{SpeedatRStation6700} \leq 55, 394$, Yes: \\
\quad \text{If,} \text{SpeedatRStation6700} > 10 \&\& \text{SpeedatRStation6700} \leq 35, 397$, Yes: \\
\quad \text{Else,} 400$, Yes; \\
400$ \text{BRANCH, 1:} \\
\quad \text{If,} \text{NZ(RL4, 9)} \&\& \text{NZ(RL4, 10)} \&\& \text{NZ(RL5, 1)} == 0.2523$, Yes: \\
\quad \text{Else,} 2524$, Yes; \\
2523$ \text{ASSIGN: } \text{CheckGapL6950Speed10less}_35.\text{NumberOut} \\
\text{True}=\text{CheckGapL6950Speed10less}_35.\text{NumberOut True} + 1; \text{NEXT}(401$); \\
2524$ \text{ASSIGN: } \text{CheckGapL6950Speed10less}_35.\text{NumberOut} \\
\text{False}=\text{CheckGapL6950Speed10less}_35.\text{NumberOut False} + 1; \text{NEXT}(402$); \\
401$ \text{TRANSPORT: LAGV, RStation7050; } \\
402$ \text{TRANSPORT: LAGV, LStation7050; } \\
388$ \text{BRANCH, 1:} \\
\quad \text{If,} \\
\quad \text{NZ(RL4, 4)} \&\& \text{NZ(RL4, 5)} \&\& \text{NZ(RL4, 6)} \&\& \text{NZ(RL4, 7)} \&\& \text{NZ(RL4, 8)} \\
\&\& \text{NZ(RL4, 9)} \&\& \text{NZ(RL4, 10)} \&\& \text{NZ(RL5, 1)} \&\& \text{NZ(RL5, 2)} == 0, \\
2525$, Yes:
Else, 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, Yes:
Else, 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) == 0,398$, Yes:
   Else,399$, 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#),361$, Yes:
   Else,362$, 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,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$,Yes;
2545$  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
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;

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;

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$);
TRANSPORT: LAGVT,RStation7050;

TRANSPORT: LAGVT,LStation7050;

; Model statements for module: Decide 923
;
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$);

TRANSPORT: LAGVT,RStation7050;

TRANSPORT: LAGVT,LStation7050;

; Model statements for module: Decide 924
;
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$);

TRANSPORT: LAGVT,RStation7050;

TRANSPORT: LAGVT,LStation7050;

; Model statements for module: Station 88
;
270

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$,Yes;
452$  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$);
ASSIGN: CheckGapL7050Speed10less.NumberOut
False=CheckGapL7050Speed10less.NumberOut False + 1:NEXT(470$);

TRANSPORT: LAGV,RStation7150;

TRANSPORT: LAGV,LStation7150;

: ; Model statements for module: Decide 936
;

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;
  ASSIGN: CheckGapL7050Speed75less.NumberOut True=CheckGapL7050Speed75less.NumberOut True + 1:NEXT(457$);

ASSIGN: CheckGapL7050Speed75less.NumberOut False=CheckGapL7050Speed75less.NumberOut False + 1:NEXT(458$);

TRANSPORT: LAGV,RStation7150;

TRANSPORT: LAGV,LStation7150;

: ; Model statements for module: Decide 937
;

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;
  ASSIGN: CheckGapL7050Speed55_75less.NumberOut True=CheckGapL7050Speed55_75less.NumberOut True + 1
  :NEXT(460$);

ASSIGN: CheckGapL7050Speed55_75less.NumberOut False=CheckGapL7050Speed55_75less.NumberOut False + 1
  :NEXT(461$);

TRANSPORT: LAGV,RStation7150;

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,Speed atRStation6950 >= 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, Yes:
    If, SpeedatRStation6950 > 55 && SpeedatRStation6950 <= 75, Yes:
    If, SpeedatRStation6950 > 35 && SpeedatRStation6950 <= 55, Yes:
    If, SpeedatRStation6950 > 10 && SpeedatRStation6950 <= 35, Yes:
    Else, 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, Yes:
    Else, 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,
      Yes:
    Else, 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

;
276

473 S 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 S, Yes:
Else, 2599 S, Yes;
2598 S ASSIGN: CheckGapL7050Speed55_75t.NumberOut
True=CheckGapL7050Speed55_75t.NumberOut True + 1:NEXT(436$);

2599 S ASSIGN: CheckGapL7050Speed55_75t.NumberOut
False=CheckGapL7050Speed55_75t.NumberOut False + 1:NEXT(435$);

436 S TRANSPORT: LAGVT, RStation7150;

435 S TRANSPORT: LAGVT, LStation7150;

; ;
; Model statements for module: Decide 944
;
474 S 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 S,
Yes:
Else, 2601 S, Yes;
2600 S ASSIGN: CheckGapL7050Speed35_55t.NumberOut
True=CheckGapL7050Speed35_55t.NumberOut True + 1:NEXT(437$);

2601 S ASSIGN: CheckGapL7050Speed35_55t.NumberOut
False=CheckGapL7050Speed35_55t.NumberOut False + 1:NEXT(438$);

437 S TRANSPORT: LAGVT, RStation7150;

438 S TRANSPORT: LAGVT, LStation7150;

; ;
; Model statements for module: Decide 945
;
475 S BRANCH, 1:
If,
NZ(RL6,1) && NZ(RL6,2) && NZ(RL6,3) && NZ(RL6,4) && NZ(RL7,1) &&
NZ(RL5,4) == 0.2602 S, Yes:
Else, 2603 S, Yes;
2602 S ASSIGN: CheckGapL7050Speed10_35t.NumberOut
True=CheckGapL7050Speed10_35t.NumberOut True + 1:NEXT(439$);

2603 S ASSIGN: CheckGapL7050Speed10_35t.NumberOut
False=CheckGapL7050Speed10_35t.NumberOut False + 1:NEXT(440$);

439 S TRANSPORT: LAGVT, RStation7150;
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$,Yes:
Else,482$,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: LAGVT,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

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;
ASSIGN: CheckGapL7050Speed55_75lesst.NumberOut
True=CheckGapL7050Speed55_75lesst.NumberOut True + 1
:NEXT(485$);
ASSIGN: CheckGapL7050Speed55_75lesst.NumberOut
False=CheckGapL7050Speed55_75lesst.NumberOut False + 1
:NEXT(486$);
TRANSPORT: LAGVT,RStation7150;
TRANSPORT: LAGVT,LStation7150;

Model statements for module: Decide 950

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;
ASSIGN: CheckGapL7050Speed35_55lesst.NumberOut
True=CheckGapL7050Speed35_55lesst.NumberOut True + 1
:NEXT(487$);
ASSIGN: CheckGapL7050Speed35_55lesst.NumberOut
False=CheckGapL7050Speed35_55lesst.NumberOut False + 1
:NEXT(488$);
TRANSPORT: LAGVT,RStation7150;
TRANSPORT: LAGVT,LStation7150;

Model statements for module: Decide 951

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;
ASSIGN: CheckGapL7050Speed10_35lesst.NumberOut True=CheckGapL7050Speed10_35lesst.NumberOut True + 1 :NEXT(489$);

ASSIGN: CheckGapL7050Speed10_35lesst.NumberOut False=CheckGapL7050Speed10_35lesst.NumberOut False + 1 :NEXT(490$);

TRANSPORT: LAGVT,RStation7150;

TRANSPORT: LAGVT,LStation7150;

; ; Model statements for module: Station 89
;

STATION, LStation7150;

DELAY: 0.0,,VA:NEXT(494$);

; ; Model statements for module: Decide 953
;

BRANCH, 1:
    If,Entity.Type==LCar,2619$,Yes:
    Else,2620$,Yes;

ASSIGN: checkvehicletype7150.NumberOut True=checkvehicletype7150.NumberOut True + 1:NEXT(495$);

ASSIGN: checkvehicletype7150.NumberOut False=checkvehicletype7150.NumberOut False + 1:NEXT(496$);

; ; Model statements for module: Decide 954
;

BRANCH, 1:
    If,SpeedatRStation7050 >= VTU(LAGV,LCar#),2621$,Yes:
    Else,2622$,Yes;

ASSIGN: Decide 954.NumberOut True=Decide 954.NumberOut True + 1:NEXT(497$);

ASSIGN: Decide 954.NumberOut False=Decide 954.NumberOut False + 1:NEXT(523$);

; ; Model statements for module: Decide 956
;

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:
281

`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$\quad$BRANCH, 1:
  \begin{align*}
  &\text{If, } \text{SpeedatRStation7050} > 75, \text{Yes;} \\
  &\text{If, } \text{SpeedatRStation7050} > 55 && \text{SpeedatRStation7050} \leq 75, \text{Yes;} \\
  &\text{If, } \text{SpeedatRStation7050} > 35 && \text{SpeedatRStation7050} \leq 55, \text{Yes;} \\
  &\text{If, } \text{SpeedatRStation7050} > 10 && \text{SpeedatRStation7050} \leq 35, \text{Yes;} \\
  &\text{Else, } \text{Yes}.
  \end{align*}

; ; ;

; Model statements for module: Decide 967

; ; ;

536$\quad$BRANCH, 1:
  \begin{align*}
  &\text{If, } \text{NZ(RL6,3)} && \text{NZ(RL6,4)} && \text{NZ(RL7,1)} = 0, \text{Yes;} \\
  &\text{Else, } \text{Yes}.
  \end{align*}

2637$\quad$ASSIGN: CheckGapL7150Speed10less.NumberOut True = CheckGapL7150Speed10less.NumberOut True + 1; \text{NEXT(537$)};

2638$\quad$ASSIGN: CheckGapL7150Speed10less.NumberOut False = CheckGapL7150Speed10less.NumberOut False + 1; \text{NEXT(538$)};

537$\quad$TRANSPORT: LAGV,RStation7250;

538$\quad$TRANSPORT: LAGV,LStation7250;

; ; ;

; Model statements for module: Decide 963

; ; ;

524$\quad$BRANCH, 1:
  \begin{align*}
  &\text{If, } \\
  &\text{NZ(RL5,2)} && \text{NZ(RL5,3)} && \text{NZ(RL5,4)} && \text{NZ(RL6,1)} \\
  && \text{NZ(RL6,2)} && \text{NZ(RL6,3)} && \text{NZ(RL6,4)} && \text{NZ(RL7,1)} && \text{NZ(RL7,2)} = 0, \\
  &\text{2639$}, \text{Yes;} \\
  &\text{Else, } \text{2640$}, \text{Yes}.
  \end{align*}

2639$\quad$ASSIGN: CheckGapL7150Speed75less.NumberOut True = CheckGapL7150Speed75less.NumberOut True + 1; \text{NEXT(525$)};

2640$\quad$ASSIGN: CheckGapL7150Speed75less.NumberOut False = CheckGapL7150Speed75less.NumberOut False + 1; \text{NEXT(526$)};

525$\quad$TRANSPORT: LAGV,RStation7250;

526$\quad$TRANSPORT: LAGV,LStation7250;

; ; ;

; Model statements for module: Decide 964

; ; ;

527$\quad$BRANCH, 1:
  \begin{align*}
  &\text{If, } \text{NZ(RL5,3)} && \text{NZ(RL5,4)} && \text{NZ(RL6,1)} \\
  && \text{NZ(RL6,2)} && \text{NZ(RL6,3)} && \text{NZ(RL6,4)} && \text{NZ(RL7,1)} = 0, \text{2641$}, \text{Yes}:
  \end{align*}
 Else,2642$, 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;

550$ ASSIGN: CheckGapL7150Speed10lesst.NumberOut True=CheckGapL7150Speed10lesst.NumberOut True + 1:NEXT(559$);

559$ TRANSPORT: LAGVT,RStation7250;

560$ TRANSPORT: LAGVT,LStation7250;

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$, Yes;
588$ ASSIGN: CheckGapL7250Speed10.NumberOut True = CheckGapL7250Speed10.NumberOut True + 1: NEXT (567$);

Model statements for module: Decide 988

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(RL7,4) == 0, 2682$, Yes:
    Else, 2683$, Yes;
  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;

ASSIGN: CheckGapL7250Speed10_35.NumberOut True = CheckGapL7250Speed10_35.NumberOut True + 1; NEXT(586$);

ASSIGN: CheckGapL7250Speed10_35.NumberOut False = CheckGapL7250Speed10_35.NumberOut False + 1; NEXT(587$);

TRANSPORT: LAGV, RStation7350;

TRANSPORT: LAGV, LStation7350;

; ; Model statements for module: Decide 989

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

BRANCH, 1:
If, NZ(RL7,3) && NZ(RL7,4) && NZ(RL8,1) == 0, 2694$, Yes:
Else, 2695$, Yes;

ASSIGN: CheckGapL7250Speed10less.NumberOut True = CheckGapL7250Speed10less.NumberOut True + 1; NEXT(605$);

ASSIGN: CheckGapL7250Speed10less.NumberOut False = CheckGapL7250Speed10less.NumberOut False + 1; NEXT(606$);

TRANSPORT: LAGV, RStation7350;

TRANSPORT: LAGV, LStation7350;

; ; Model statements for module: Decide 990

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.NumberOutTrue=CheckGapL7250Speed35_55t.NumberOutTrue + 1:NEXT(573$);
2715$ ASSIGN: CheckGapL7250Speed35_55t.NumberOutFalse=CheckGapL7250Speed35_55t.NumberOutFalse + 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.NumberOutTrue=CheckGapL7250Speed10_35t.NumberOutTrue + 1:NEXT(575$);
2717$ ASSIGN: CheckGapL7250Speed10_35t.NumberOutFalse=CheckGapL7250Speed10_35t.NumberOutFalse + 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$);
TRANSPORT: LAGVT,RStation7350;
TRANSPORT: LAGVT,LStation7350;

; Model statements for module: Decide 1004
; 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$);
TRANSPORT: LAGVT,RStation7350;
TRANSPORT: LAGVT,LStation7350;

; Model statements for module: Decide 1005
; 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$);
TRANSPORT: LAGVT,RStation7350;
TRANSPORT: LAGVT,LStation7350;

; Model statements for module: Station 91
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$);
Model statements for module: Decide 1011

\[
\begin{align*}
\text{BRANCH}, & 1: \\
\text{If,} & \NZ(RL9,1) \&\& \NZ(RL9,2) \&\& \NZ(RL9,3) \&\& \NZ(RL9,4) \&\& \NZ(RL10,1) \\
& \&\& \NZ(RL10,2) \&\& \NZ(RL11,1) \&\& \NZ(RL11,2) \&\& \NZ(RL8,4) = 0, \\
& \text{Yes:} \\
& \text{Else,} \\
& \text{Yes:} \\
\end{align*}
\]

\[
\begin{align*}
& \text{ASSIGN:} \text{CheckGapL7350Speed75.NumberOut} \\
& \text{True=} \text{CheckGapL7350Speed75.NumberOut True} + 1; \\
& \text{NEXT(635$)}; \\
\end{align*}
\]

\[
\begin{align*}
& \text{ASSIGN:} \text{CheckGapL7350Speed75.NumberOut} \\
& \text{False=} \text{CheckGapL7350Speed75.NumberOut False} + 1; \\
& \text{NEXT(636$)}; \\
\end{align*}
\]

Model statements for module: Decide 1012

\[
\begin{align*}
\text{BRANCH}, & 1: \\
\text{If,} & \NZ(RL9,1) \&\& \NZ(RL9,2) \&\& \NZ(RL9,3) \&\& \NZ(RL9,4) \&\& \NZ(RL10,1) \\
& \&\& \NZ(RL10,2) \&\& \NZ(RL8,4) = 0.2743$, \\
& \text{Yes:} \\
& \text{Else,} \\
& \text{Yes:} \\
\end{align*}
\]

\[
\begin{align*}
& \text{ASSIGN:} \text{CheckGapL7350Speed55_75.NumberOut} \\
& \text{True=} \text{CheckGapL7350Speed55_75.NumberOut True} + 1; \\
& \text{NEXT(648$)}; \\
\end{align*}
\]

\[
\begin{align*}
& \text{ASSIGN:} \text{CheckGapL7350Speed55_75.NumberOut} \\
& \text{False=} \text{CheckGapL7350Speed55_75.NumberOut False} + 1; \\
& \text{NEXT(649$)}; \\
\end{align*}
\]

Model statements for module: Decide 1013

\[
\begin{align*}
\text{BRANCH}, & 1: \\
\end{align*}
\]
If, NZ(RL9,1) && NZ(RL9,2) && NZ(RL9,3) && NZ(RL9,4) && NZ(RL8,4) == 0, 0.2745$, Yes:  
Else, 0.2746$, 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;

653$  
654$ TRANSPORT: LAGV, RStation7450;
655$ TRANSPORT: LAGV, LStation7450;

659$  
662$  
672$  
2751$ ASSIGN: CheckGapL7350Speed10less.NumberOut
True = CheckGapL7350Speed10less.NumberOut True + 1:NEXT(673$);
\[ \text{ASSIGN: CheckGapL7350Speed10less.NumberOut False=CheckGapL7350Speed10less.NumberOut False + 1:NEXT(674)}; \]

\[ \text{TRANSPORT: LAGV,RStation7450;} \]

\[ \text{TRANSPORT: LAGV,LStation7450;} \]

\[ \text{Model statements for module: Decide 1017} \]

\[ \text{BRANCH, 1:} \]
\[ \text{If,} \]
\[ \text{NZ(RL7,2) && NZ(RL7,3) && NZ(RL7,4) && NZ(RL8,1)} \]
\[ && \text{NZ(RL8,2) && NZ(RL8,3) && NZ(RL8,4) && NZ(RL9,1) && NZ(RL9,2) == 0,} \]
\[ \text{2753$, Yes:} \]
\[ \text{Else, 2754$, Yes;} \]
\[ \text{ASSIGN: CheckGapL7350Speed75less.NumberOut True=CheckGapL7350Speed75less.NumberOut True + 1:NEXT(661)}; \]

\[ \text{ASSIGN: CheckGapL7350Speed75less.NumberOut False=CheckGapL7350Speed75less.NumberOut False + 1:NEXT(662)}; \]

\[ \text{TRANSPORT: LAGV,RStation7450;} \]

\[ \text{TRANSPORT: LAGV,LStation7450;} \]

\[ \text{Model statements for module: Decide 1018} \]

\[ \text{BRANCH, 1:} \]
\[ \text{If,} \]
\[ \text{NZ(RL7,3) && NZ(RL7,4) && NZ(RL8,1)} \]
\[ && \text{NZ(RL8,2) && NZ(RL8,3) && NZ(RL8,4) && NZ(RL9,1) == 0, 2755$, Yes:} \]
\[ \text{Else, 2756$, Yes;} \]
\[ \text{ASSIGN: CheckGapL7350Speed55_75less.NumberOut True=CheckGapL7350Speed55_75less.NumberOut True + 1 :NEXT(664)}; \]

\[ \text{ASSIGN: CheckGapL7350Speed55_75less.NumberOut False=CheckGapL7350Speed55_75less.NumberOut False + 1 :NEXT(665)}; \]

\[ \text{TRANSPORT: LAGV,RStation7450;} \]

\[ \text{TRANSPORT: LAGV,LStation7450;} \]

\[ \text{Model statements for module: Decide 1019} \]
If, NZ(RL7,4) && NZ(RL8,1) && NZ(RL8,2) && NZ(RL8,3) && NZ(RL8,4) && NZ(RL9,1) == 0, Yes;
Else, Yes;

ASSIGN: CheckGapL7350Speed35_55less.NumberOut True=CheckGapL7350Speed35_55less.NumberOut True + 1;
:NEXT(668$);

ASSIGN: CheckGapL7350Speed35_55less.NumberOut False=CheckGapL7350Speed35_55less.NumberOut False + 1;
:NEXT(668$);

TRANSPORT: LAGV, RStation7450;

TRANSPORT: LAGV, LStation7450;

If, NZ(RL8,2) && NZ(RL8,3) && NZ(RL8,4) && NZ(RL9,1) == 0, Yes;
Else, Yes;

ASSIGN: CheckGapL7350Speed10_35less.NumberOut True=CheckGapL7350Speed10_35less.NumberOut True + 1;
:NEXT(670$);

ASSIGN: CheckGapL7350Speed10_35less.NumberOut False=CheckGapL7350Speed10_35less.NumberOut False + 1;
:NEXT(671$);

TRANSPORT: LAGV, RStation7450;

TRANSPORT: LAGV, LStation7450;

If, SpeedatRStation7250 >= VTU(LAGVT, LTruck#), Yes;
Else, Yes;

ASSIGN: Decide 1009.NumberOut True=Decide 1009.NumberOut True + 1;
:NEXT(675$);

ASSIGN: Decide 1009.NumberOut False=Decide 1009.NumberOut False + 1;
:NEXT(681$);
; Model statements for module: Decide 1022
;
675$ BRANCH, 1:
   If, SpeedatRStation7250 > 75, Yes:
   If, SpeedatRStation7250 > 55 && SpeedatRStation7250 <= 75, Yes:
   If, SpeedatRStation7250 > 35 && SpeedatRStation7250 <= 55, Yes:
   If, SpeedatRStation7250 > 10 && SpeedatRStation7250 <= 35, Yes:
   Else, 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, Yes:
   Else, 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,
   Yes:
   Else, 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
;
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
; ;
677$ 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$,
2771$ 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;
TRANSPORT: LAGVT,LStation7450;

Model statements for module: Decide 1028

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;

ASSIGN: CheckGapL7350Speed10lesst.NumberOut True=CheckGapL7350Speed10lesst.NumberOut True + 1:NEXT(695$);

ASSIGN: CheckGapL7350Speed75lesst.NumberOut False=CheckGapL7350Speed75lesst.NumberOut False + 1:NEXT(688$);

TRANSPORT: LAGVT,RStation7450;

TRANSPORT: LAGVT,LStation7450;

Model statements for module: Decide 1033

BRANCH, 1:
If,NZ(RL8,1) &&NZ(RL8,2)&&NZ(RL8 ,3)&&NZ(RL8,4) &&NZ(RL9,1)==0,2777$,Yes:
Else,2778$,Yes;

ASSIGN: CheckGapL7350Speed10lesst.NumberOut True=CheckGapL7350Speed10lesst.NumberOut True + 1:NEXT(695$);

ASSIGN: CheckGapL7350Speed10lesst.NumberOut False=CheckGapL7350Speed10lesst.NumberOut False + 1:NEXT(696$);

TRANSPORT: LAGVT,RStation7450;

TRANSPORT: LAGVT,LStation7450;

Model statements for module: Decide 1029

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;

ASSIGN: CheckGapL7350Speed75lesst.NumberOut True=CheckGapL7350Speed75lesst.NumberOut True + 1:NEXT(687$);

ASSIGN: CheckGapL7350Speed75lesst.NumberOut False=CheckGapL7350Speed75lesst.NumberOut False + 1:NEXT(688$);

TRANSPORT: LAGVT,RStation7450;

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;
ASSIGN: CheckGapL7350Speed10_35lesst.NumberOut
True=CheckGapL7350Speed10_35lesst.NumberOut True + 1
:NEXT(693$);

ASSIGN: CheckGapL7350Speed10_35lesst.NumberOut
False=CheckGapL7350Speed10_35lesst.NumberOut False + 1
:NEXT(694$);

TRANSPORT: LAGVT,RStation7450;

TRANSPORT: LAGVT,LStation7450;

STATION: RStation7450;

DELAY: 0.0,,VA:NEXT(735$);

BRANCH, 1:
    If,Entity.Type==RCar,731$,Yes:
    If,Entity.Type==RTruck,732$,Yes:
    If,Entity.Type==LCar,733$,Yes:
    Else,734$,Yes;

ASSIGN: SpeedatRStation7450=VTU(LAGVT,LTruck#):NEXT(701$);

BRANCH, 1:
    If,Entity.Type==RCar,2792$,Yes:
    Else,2793$,Yes;

ASSIGN: Decide 1277.NumberOut True=Decide 1277.NumberOut True + 1:NEXT(700$);

ASSIGN: Decide 1277.NumberOut False=Decide 1277.NumberOut False + 1:NEXT(703$);

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$);
Model statements for module: Decide 1282

711$ BRANCH, 1:
   If, Entity.Type==LCar, 709$, Yes:
   Else, 710$, Yes;

709$ TRANSPORT: RAGVT, RStation7550;

709$ ;

711$ ;

711$ Model statements for module: Decide 1282

711$ ;

711$ ;

711$ ;

712$ ASSIGN: Decide 1282.NumberOut True = Decide 1282.NumberOut True + 1: NEXT(712$);

713$ ASSIGN: Decide 1282.NumberOut False = Decide 1282.NumberOut False + 1: NEXT(713$);

712$ TRANSPORT: LAGV, RStation7550;

713$ TRANSPORT: LAGVT, RStation7550;

713$ ;

713$ ;

713$ ;

713$ ;

713$ ;

721$ ASSIGN: SpeedatRStation7500 = VTU(RAGV, RCar#): NEXT(708$);

722$ ASSIGN: SpeedatRStation7500 = VTU(RAGVT, RTruck#): NEXT(708$);

723$ ASSIGN: SpeedatRStation7500 = VTU(LAGV, LCar#): NEXT(708$);

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$);
ASSIGN: Decide 1285.NumberOut False=Decide 1285.NumberOut False + 1:NEXT(720$);

TRANSPORT: LAGV,RStation7600;

TRANSPORT: LAGVT,RStation7600;

ASSIGN: SpeedatRStation7550=VTU(RAGV,RCar#):NEXT(715$);

ASSIGN: SpeedatRStation7550=VTU(RAGVT,RTruck#):NEXT(715$);

ASSIGN: SpeedatRStation7550=VTU(LAGV,LCar#):NEXT(715$);

STATION, LStation7450;
DELAY: 0.0.,VA:NEXT(737$);

BRANCH, 1:
  If,Entity.Type==LCar,2823$,Yes:
  Else,2824$,Yes:
ASSIGN: checkvehicletype7450.NumberOut True=checkvehicletype7450.NumberOut True + 1:NEXT(738$);

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;
ASSIGN: CheckGapL7450Speed75.NumberOut
True=CheckGapL7450Speed75.NumberOut True + 1:NEXT(742$);

ASSIGN: CheckGapL7450Speed75.NumberOut
False=CheckGapL7450Speed75.NumberOut False + 1:NEXT(743$);

TRANSPORT: LAGV,RStation7500,NORM(101,3);

TRANSPORT: LAGV,LStation7500,NORM(101,3);

; Model statements for module: Decide 1294
;

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,
2835$,Yes:
Else,2836$,Yes;

ASSIGN: CheckGapL7450Speed55_75.NumberOut
True=CheckGapL7450Speed55_75.NumberOut True + 1:NEXT(755$);

ASSIGN: CheckGapL7450Speed55_75.NumberOut
False=CheckGapL7450Speed55_75.NumberOut False + 1:NEXT(756$);

TRANSPORT: LAGV,RStation7500,NORM(101,3);

TRANSPORT: LAGV,LStation7500,NORM(101,3);

; Model statements for module: Decide 1295
;

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;

ASSIGN: CheckGapL7450Speed35_55.NumberOut
True=CheckGapL7450Speed35_55.NumberOut True + 1:NEXT(758$);

ASSIGN: CheckGapL7450Speed35_55.NumberOut
False=CheckGapL7450Speed35_55.NumberOut False + 1:NEXT(759$);

TRANSPORT: LAGV,RStation7500,NORM(101,3);

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, Yes:
Else, 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);

766$ BRANCH, 1:
If, SpeedatRStation7350 > 75, Yes:
If, SpeedatRStation7350 > 55 && SpeedatRStation7350 <= 75, Yes:
If, SpeedatRStation7350 > 35 && SpeedatRStation7350 <= 55, Yes:
If, SpeedatRStation7350 > 10 && SpeedatRStation7350 <= 35, Yes:
Else, Yes;
779$ BRANCH, 1:
If, NZ(RL9,3) && NZ(RL9,4) && NZ(RL10,1) == 0, Yes:
Else, 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);

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;
ASSIGN: CheckGapL7450Speed10t.NumberOut
True=CheckGapL7450Speed10t.NumberOut True + 1:NEXT(752$);

ASSIGN: CheckGapL7450Speed10t.NumberOut
False=CheckGapL7450Speed10t.NumberOut False + 1:NEXT(753$);

TRANSPORT: LAGVT,RStation7500,NORM(101,3);

TRANSPORT: LAGVT,LStation7500,NORM(101,3);

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;

ASSIGN: CheckGapL7450Speed75t.NumberOut
True=CheckGapL7450Speed75t.NumberOut True + 1:NEXT(744$);

ASSIGN: CheckGapL7450Speed75t.NumberOut
False=CheckGapL7450Speed75t.NumberOut False + 1:NEXT(745$);

TRANSPORT: LAGVT,RStation7500,NORM(101,3);

TRANSPORT: LAGVT,LStation7500,NORM(101,3);

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,
2859$,Yes:
Else,2860$,Yes;

ASSIGN: CheckGapL7450Speed55_75t.NumberOut
True=CheckGapL7450Speed55_75t.NumberOut True + 1:NEXT(747$);

ASSIGN: CheckGapL7450Speed55_75t.NumberOut
False=CheckGapL7450Speed55_75t.NumberOut False + 1:NEXT(746$);

TRANSPORT: LAGVT,RStation7500,NORM(101,3);

TRANSPORT: LAGVT,LStation7500,NORM(101,3);
319

; 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$,Yes;
    ;
    ; Model statements for module: Decide 1315
0.2867$, 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(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(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
321

: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(RL8,3) &&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
322

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
324

; 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;
ASSIGN: CheckGapL7500Speed10less.NumberOut True = CheckGapL7500Speed10less.NumberOut True + 1: NEXT(848$);

ASSIGN: CheckGapL7500Speed10less.NumberOut False = CheckGapL7500Speed10less.NumberOut False + 1: NEXT(849$);

TRANSPORT: LAGV, RStation7550;

TRANSPORT: LAGV, LStation7550;

BRANCH, 1:
If,
NZ(RL8,4) && NZ(RL9,1) && NZ(RL9,2) && NZ(RL9,3) && NZ(RL10,1) && NZ(RL10,2) && NZ(RL11,1) && NZ(RL11,2) == 0,
ASSIGN: CheckGapL7500Speed75less.NumberOut True = CheckGapL7500Speed75less.NumberOut True + 1: NEXT(836$);
Else,
ASSIGN: CheckGapL7500Speed55_75less.NumberOut True = CheckGapL7500Speed55_75less.NumberOut True + 1: NEXT(839$);

TRANSPORT: LAGV, RStation7550;

TRANSPORT: LAGV, LStation7550;

BRANCH, 1:
If,
NZ(RL9,1) && NZ(RL9,2) && NZ(RL9,3) && NZ(RL10,1) && NZ(RL10,2) && NZ(RL11,1) == 0,
ASSIGN: CheckGapL7500Speed55_75less.NumberOut True = CheckGapL7500Speed55_75less.NumberOut True + 1: NEXT(839$);
Else,
ASSIGN: CheckGapL7500Speed55_75less.NumberOut False = CheckGapL7500Speed55_75less.NumberOut False + 1: NEXT(840$);

TRANSPORT: LAGV, RStation7550;

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$, Yes:
     If,SpeedatRStation740 > 55 && SpeedatRStation7450 <= 75,852$, 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$);
Model statements for module: Decide 1337

\[
\text{If, Speed at } R\text{Station7450} > 75, \text{Yes:}
\]

\[
\text{If, Speed at } R\text{Station7450} > 55 \&\& \text{ Speed at } R\text{Station7450} <= 75, \text{Yes:}
\]

\[
\text{If, Speed at } R\text{Station7450} > 35 \&\& \text{ Speed at } R\text{Station7450} <= 55, \text{Yes:}
\]

\[
\text{If, Speed at } R\text{Station7450} > 10 \&\& \text{ Speed at } R\text{Station7450} <= 35, \text{Yes:}
\]

\[
\text{Else, Yes:}
\]

Model statements for module: Decide 1342

\[
\text{If, } \text{NZ(RL9,3) \&\& NZ(RL9,4) \&\& NZ(RL10,1) \&\& NZ(RL10,2) \&\& NZ(RL11,1) == 0, Yes:}
\]

\[
\text{Else, Yes:}
\]

ASSIGN: CheckGapL7500Speed10lesst.NumberOut True = CheckGapL7500Speed10lesst.NumberOut True + 1:NEXT(870$);

ASSIGN: CheckGapL7500Speed10lesst.NumberOut False = CheckGapL7500Speed10lesst.NumberOut False + 1:NEXT(871$);

TRANSPORT: LAGVT,RStation7550;

TRANSPORT: LAGVT,LStation7550;

Model statements for module: Decide 1338

\[
\text{If,}
\]

\[
\text{NZ(RL8,3) \&\& NZ(RL8,4) \&\& NZ(RL9,1) \&\& NZ(RL9,2) \&\& NZ(RL9,3) \&\& NZ(RL9,4) \&\& NZ(RL10,1) \&\& NZ(RL10,2) \&\& NZ(RL11,1) \&\& NZ(RL11,2) == 0, Yes:}
\]

\[
\text{Else, Yes:}
\]

ASSIGN: CheckGapL7500Speed75lesst.NumberOut True = CheckGapL7500Speed75lesst.NumberOut True + 1:NEXT(862$);

ASSIGN: CheckGapL7500Speed75lesst.NumberOut False = CheckGapL7500Speed75lesst.NumberOut False + 1:NEXT(863$);

TRANSPORT: LAGVT,RStation7550;
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: CheckGapL7500.Speed10_35lesst.NumberOut
True = CheckGapL7500.Speed10_35lesst.NumberOut True + 1
:NEXT(868$);
2933$ ASSIGN: CheckGapL7500.Speed10_35lesst.NumberOut
False = CheckGapL7500.Speed10_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;

  ASSIGN: CheckGapL7550Speed10.NumberOut True = CheckGapL7550Speed10.NumberOut True + 1:NEXT(900$);

  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;

  ASSIGN: CheckGapL7550Speed75.NumberOut True = CheckGapL7550Speed75.NumberOut True + 1:NEXT(878$);

  ASSIGN: CheckGapL7550Speed75.NumberOut False = CheckGapL7550Speed75.NumberOut False + 1:NEXT(879$);

878$  TRANSPORT: LAGV,RStation7600;

879$  TRANSPORT: LAGV,LStation7600;

Model statements for module: Decide 1348
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,\(2952$,Yes;\)

\[
2951$ ASSIGN: CheckGapL7550Speed10_35.NumberOut True=CheckGapL7550Speed10_35.NumberOut True + 1:NEXT(897$);
\]

\[
2952$ ASSIGN: CheckGapL7550Speed10_35.NumberOut False=CheckGapL7550Speed10_35.NumberOut False + 1:NEXT(898$);
\]

\[
897$ TRANSPORT: LAGV,RStation7600;
\]

\[
898$ TRANSPORT: LAGV,LStation7600;
\]
334

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$,Yes:
        Else,912$,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,
        904$,Yes:
        Else,2958$,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;

909$ BRANCH, 1:
   If,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;

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;

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$, Yes;
929$ ASSIGN: CheckGapL7550Speed10lesst.NumberOut True=CheckGapL7550Speed10lesst.NumberOut True + 1:NEXT(938$);
938$ TRANSPORT: LAGVT,RStation7600;
939$ TRANSPORT: LAGVT,LStation7600;
Model statements for module: Decide 1365

BRANCH, 1:
If,
\[\text{NZ}(RL9,1) \land \text{NZ}(RL9,2) \land \text{NZ}(RL9,3) \land \text{NZ}(RL9,4) \land \text{NZ}(RL10,1) \land \text{NZ}(RL10,2) \land \text{NZ}(RL11,1) \land \text{NZ}(RL11,2) \land \text{NZ}(RL12,1) \land \text{NZ}(RL12,2) = 0,\]
\[\text{Yes} = 1;\]
Else,\[\text{Yes} = 1;\]
\[\text{ASSIGN:} \quad \text{CheckGapL7550Speed75lesst.NumberOut True = CheckGapL7550Speed75lesst.NumberOut True + 1; \text{NEXT} (930$);}\]
\[\text{ASSIGN:} \quad \text{CheckGapL7550Speed75lesst.NumberOut False = CheckGapL7550Speed75lesst.NumberOut False + 1; \text{NEXT} (931$);}\]
\[\text{TRANSPORT:} \quad \text{LAGVT, RStation7600};\]
\[\text{TRANSPORT:} \quad \text{LAGVT, LStation7600};\]

Model statements for module: Decide 1366

BRANCH, 1:
If,
\[\text{NZ}(RL9,2) \land \text{NZ}(RL9,3) \land \text{NZ}(RL9,4) \land \text{NZ}(RL10,1) \land \text{NZ}(RL10,2) \land \text{NZ}(RL11,1) \land \text{NZ}(RL11,2) \land \text{NZ}(RL12,1) = 0,\]
\[\text{Yes} = 1;\]
Else,\[\text{Yes} = 1;\]
\[\text{ASSIGN:} \quad \text{CheckGapL7550Speed55_75lesst.NumberOut True = CheckGapL7550Speed55_75lesst.NumberOut True + 1; \text{NEXT} (932$);}\]
\[\text{ASSIGN:} \quad \text{CheckGapL7550Speed55_75lesst.NumberOut False = CheckGapL7550Speed55_75lesst.NumberOut False + 1; \text{NEXT} (933$);}\]
\[\text{TRANSPORT:} \quad \text{LAGVT, RStation7600};\]
\[\text{TRANSPORT:} \quad \text{LAGVT, LStation7600};\]

Model statements for module: Decide 1367

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$);
TRANSPORT: LAGV, RStation7650;

TRANSPORT: LAGVT, RStation7650;

Model statements for module: Assign 159

ASSIGN: SpeedatRStation7600=VTU(RAGV, RCar#); NEXT(943$);

Model statements for module: Assign 160

ASSIGN: SpeedatRStation7600=VTU(RAGVT, RTruck#); NEXT(943$);

Model statements for module: Assign 161

ASSIGN: SpeedatRStation7600=VTU(LAGV, LCar#); NEXT(943$);

Model statements for module: Station 108

STATION, RStation7650;
DELAY: 0.0, VA; NEXT(965$);

Model statements for module: Decide 1377

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

ASSIGN: SpeedatRStation7650=VTU(LAGVT, LTruck#); NEXT(950$);
950$ 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;

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;

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;

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$, Yes:
   If, SpeedatRStation7550 > 55 && SpeedatRStation7550 <= 75, 972$, Yes:
   If, SpeedatRStation7550 > 35 && SpeedatRStation7550 <= 55, 973$, Yes:
   Else, 974$, Yes;

Model statements for module: Decide 1386

993$ BRANCH, 1:
   If, NZ(RL13,1) && NZ(RL13,2) && NZ(RL14,1) && NZ(RL14,2) == 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(RL14,2) &&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;
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$, Yes;
1017$ 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

BRANCH, 1:
If,NZ(RL13,1) && NZ(RL13,2) && NZ(RL14,1) && NZ(RL14,2) && NZ(RL12,2) == 0, Yes:
Else, Yes;

ASSIGN: CheckGapL7600Speed10_35t.NumberOut True = CheckGapL7600Speed10_35t.NumberOut True + 1; NEXT(980$);

ASSIGN: CheckGapL7600Speed10_35t.NumberOut False = CheckGapL7600Speed10_35t.NumberOut False + 1; NEXT(981$);

TRANSPORT: LAGVT, RStation7650;

TRANSPORT: LAGVT, LStation7650;

Model statements for module: Decide 1399

BRANCH, 1:
If, SpeedatRStation7550 > 75, Yes:
If, SpeedatRStation7550 > 55 && SpeedatRStation7550 <= 75, Yes:
If, SpeedatRStation7550 > 35 && SpeedatRStation7550 <= 55, Yes:
If, SpeedatRStation7550 > 10 && SpeedatRStation7550 <= 35, Yes:
Else, Yes;

ASSIGN: CheckGapL7600Speed10less.NumberOut True = CheckGapL7600Speed10less.NumberOut True + 1; NEXT(1032$);

ASSIGN: CheckGapL7600Speed10less.NumberOut False = CheckGapL7600Speed10less.NumberOut False + 1; NEXT(1033$);

TRANSPORT: LAGVT, RStation7650;

TRANSPORT: LAGVT, LStation7650;

Model statements for module: Decide 1404

BRANCH, 1:
If, NZ(RL11,1) && NZ(RL11,2) && NZ(RL12,1) && NZ(RL12,2) && NZ(RL13,1) == 0, Yes:
Else, Yes;

ASSIGN: CheckGapL7600Speed10lesst.NumberOut True = CheckGapL7600Speed10lesst.NumberOut True + 1; NEXT(1032$);

ASSIGN: CheckGapL7600Speed10lesst.NumberOut False = CheckGapL7600Speed10lesst.NumberOut False + 1; NEXT(1033$);

TRANSPORT: LAGVT, RStation7650;

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,
  Yes:
  Else, Yes;
  ASSIGNED: CheckGapL7600Speed75lesst.NumberOut
  True=CheckGapL7600Speed75lesst.NumberOut True + 1:NEXT(1024$);
  Assign:
  False=CheckGapL7600Speed75lesst.NumberOut False + 1:NEXT(1025$);
  TRANSPORT: LAGVT,RStation7650;
  TRANSPORT: LAGVT,LStation7650;

; 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,
  Yes:
  Else, Yes;
  ASSIGNED: CheckGapL7600Speed55_75lesst.NumberOut
  True=CheckGapL7600Speed55_75lesst.NumberOut True + 1:NEXT(1026$);
  Assign:
  False=CheckGapL7600Speed55_75lesst.NumberOut False + 1:NEXT(1027$);
  TRANSPORT: LAGVT,RStation7650;
  TRANSPORT: LAGVT,LStation7650;

; 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;
  ASSIGNED: CheckGapL7600Speed35_55lesst.NumberOut
  True=CheckGapL7600Speed35_55lesst.NumberOut True + 1
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==L.Car,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, Speed at RStation7600 >= 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, Speed at RStation7600 > 75, 1039$, Yes:
   If, Speed at RStation7600 > 55 && Speed at RStation7600 <= 75, 1052$, Yes:
   If, Speed at RStation7600 > 35 && Speed at RStation7600 <= 55, 1055$, Yes:
   If, Speed at RStation7600 > 10 && Speed at RStation7600 <= 35, 1058$, Yes:
   Else, 1061$, Yes;

1061$ TRANSPORT: LAGV, RStation7700;

Model statements for module: Decide 1413

1061$ BRANCH, 1:
   If, NZ(RL14,1) && NZ(RL14,2) && NL(RL15) && NL(RL16) && NL(RL17) && 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;
ASSIGN: CheckGapL7650Speed75.NumberOut
True=CheckGapL7650Speed75.NumberOut True + 1:NEXT(1040$);

ASSIGN: CheckGapL7650Speed75.NumberOut
False=CheckGapL7650Speed75.NumberOut False + 1:NEXT(1041$);

TRANSPORT: LAGV,RStation7700;

TRANSPORT: LAGV,LStation7700;

\[
\begin{align*}
3083$ & \text{ASSIGN: CheckGapL7650Speed}5_5_75\text{.NumberOut} \\
& \text{True} = \text{CheckGapL7650Speed}_5_5_75\text{.NumberOut True} + 1\text{:NEXT(1053$)}; \\
3084$ & \text{ASSIGN: CheckGapL7650Speed}3_5_5_75\text{.NumberOut} \\
& \text{False} = \text{CheckGapL7650Speed}3_5_5_75\text{.NumberOut False} + 1\text{:NEXT(1054$)}; \\
1053$ & \text{TRANSPORT: LAGV,RStation}7700; \\
1054$ & \text{TRANSPORT: LAGV,LStation}7700; \\
\end{align*}
\]
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$, Yes;
1085$ BRANCH, 1:
    If, NZ(RL14,1) && NZ(RL14,2) && NL(RL15) && NZ(RL13,2) == 0, 3105$, Yes:
Else, \(3106\), Yes;
3105$ ASSIGN: \text{CheckGapL7650Speed10t.NumberOut}
True=CheckGapL7650Speed10t.NumberOut True + 1:NEXT(1050$);

3106$ ASSIGN: \text{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: \text{CheckGapL7650Speed75t.NumberOut}
True=CheckGapL7650Speed75t.NumberOut True + 1:NEXT(1042$);

3108$ ASSIGN: \text{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: \text{CheckGapL7650Speed55_75t.NumberOut}
True=CheckGapL7650Speed55_75t.NumberOut True + 1:NEXT(1045$);

3110$ ASSIGN: \text{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
Model statements for module: Decide 1424

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 1426

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 1431

1086$ BRANCH, 1:
   If,SspeedatRStation7600>75,1087$,Yes:
      If,SspeedatRStation7600 > 55 && SspeedatRStation7600 <= 75,1088$,Yes:
         If,SspeedatRStation7600 > 35 && SspeedatRStation7600 <= 55,1089$,Yes:
            If,SspeedatRStation7600 > 10 && SspeedatRStation7600 <= 35,1090$,Yes:
               Else,1091$,Yes;
1091$ BRANCH, 1:
   If,NZ(RL12,1) &&NZ(RL12,2) &&NZ(RL13,1) &&NZ(RL13,2) &&NZ(RL14,1) == 0,3117$,Yes:
      Else,3118$,Yes;
ASSIGN: CheckGapL7650Speed10lesst.NumberOut
True = CheckGapL7650Speed10lesst.NumberOut True + 1:NEXT(1100$);

ASSIGN: CheckGapL7650Speed10lesst.NumberOut
False = CheckGapL7650Speed10lesst.NumberOut False + 1:NEXT(1101$);

TRANSPORT: LAGVT,RStation7700;

TRANSPORT: LAGVT,LStation7700;

ASSIGN: CheckGapL7650Speed75lesst.NumberOut
True = CheckGapL7650Speed75lesst.NumberOut True + 1:NEXT(1092$);

ASSIGN: CheckGapL7650Speed75lesst.NumberOut
False = CheckGapL7650Speed75lesst.NumberOut False + 1:NEXT(1093$);

TRANSPORT: LAGVT,RStation7700;

TRANSPORT: LAGVT,LStation7700;

ASSIGN: CheckGapL7650Speed55_75lesst.NumberOut
True = CheckGapL7650Speed55_75lesst.NumberOut True + 1:NEXT(1094$);

ASSIGN: CheckGapL7650Speed55_75lesst.NumberOut
False = CheckGapL7650Speed55_75lesst.NumberOut False + 1:NEXT(1095$);

TRANSPORT: LAGVT,RStation7700;

TRANSPORT: LAGVT,LStation7700;
Model statements for module: Decide 1429

1089$ BRANCH, 1:
   If,NZ(RL11,1)&amp;NZ(RL11,2)
   &amp;NZ(RL12,1)&amp;NZ(RL12,2)&amp;NZ(RL13,1)&amp;NZ(RL13,2)&amp;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(RL1,2)
   &amp;NZ(RL1,1)&amp;NZ(RL12,1)&amp;NZ(RL13,1)&amp;NZ(RL13,2)&amp;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

114$   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

113$   ASSIGN: SpeedatRStation7700=VTU(LAGVT,LTruck#):NEXT(1104$);

Model statements for module: Decide 1432

104$   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

106$   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;
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;
ASSIGN: checkvehicletype7700.NumberOut True=checkvehicletype7700.NumberOut True + 1:NEXT(1117$);

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
366

; 1120$ BRANCH, 1:
    If, NL(RL15) && NL(RL16) && NL(RL17) && NL(RL18) && NZ(RL14, 2) == 0, Yes:
    Else, 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;

; 1133$ BRANCH, 1:
    If, NL(RL15) && NL(RL16) && NL(RL17) && NZ(RL14, 2) == 0, Yes:
    Else, Yes;
3151$ ASSIGN: CheckGapL7700Speed35_55.NumberOut
    True = CheckGapL7700Speed35_55.NumberOut True + 1; NEXT(1134$);
3152$ ASSIGN: CheckGapL7700Speed35_55.NumberOut
    False = CheckGapL7700Speed35_55.NumberOut False + 1; NEXT(1135$);
1134$ TRANSPORT: LAGV, RStation7750;
1135$ TRANSPORT: LAGV, LStation7750;

; 1166$ BRANCH, 1:
    If, NL(RL15) && NL(RL16) && NZ(RL14, 2) == 0, Yes:
    Else, 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;

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 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
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, Yes:

If SpeedatRStation7650 > 35 && SpeedatRStation7650 <= 55, Yes:

If SpeedatRStation7650 > 10 && SpeedatRStation7650 <= 35, Yes:

Else, Yes;

Model statements for module: Decide 1456

BRANCH, 1:

If NL(RL15) && NL(RL16) && NZ(RL14, 2) == 0, Yes:

Else, Yes;

ASSIGN: CheckGapL7700Speed10t.NumberOut
True = CheckGapL7700Speed10t.NumberOut True + 1: NEXT(1131$);

ASSIGN: CheckGapL7700Speed10t.NumberOut
False = CheckGapL7700Speed10t.NumberOut False + 1: NEXT(1132$);

TRANSPORT: LAGVT, RStation7750;

TRANSPORT: LAGVT, LStation7750;

Model statements for module: Decide 1452

BRANCH, 1:

If NL(RL15) && NL(RL16) && NL(RL17) && NL(RL18) && NL(RL19) && NZ(RL14, 2) == 0, Yes:

Else, Yes;

ASSIGN: CheckGapL7700Speed75t.NumberOut
True = CheckGapL7700Speed75t.NumberOut True + 1: NEXT(1123$);

ASSIGN: CheckGapL7700Speed75t.NumberOut
False = CheckGapL7700Speed75t.NumberOut False + 1: NEXT(1124$);

TRANSPORT: LAGVT, RStation7750;

TRANSPORT: LAGVT, LStation7750;

Model statements for module: Decide 1453

BRANCH, 1:

If NL(RL15) && NL(RL16) && NL(RL17) && NL(RL18) && NZ(RL14, 2) ==

0.3175$, Yes:

Else, Yes;

ASSIGN: CheckGapL7700Speed55_75t.NumberOut
True = CheckGapL7700Speed55_75t.NumberOut True + 1: NEXT(1126$);
371

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

BRANCH, 1:
If,NL(RL15) && NL(RL16) && NL(RL17) && NZ(RL14,2) == 0,3179$, Yes:
Else,3180$, Yes;

ASSIGN: CheckGapL7700Speed35_55t.NumberOut
True=CheckGapL7700Speed35_55t.NumberOut True + 1:NEXT(1127$);

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

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(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$, Yes;
ASSIGN: CheckGapL7700Speed10_35lesst.NumberOut
True=CheckGapL7700Speed10_35lesst.NumberOut True + 1
:NEXT(1179$);

ASSIGN: CheckGapL7700Speed10_35lesst.NumberOut
False=CheckGapL7700Speed10_35lesst.NumberOut False + 1
:NEXT(1180$);

TRANSPORT: LAGVT,RStation7750;

TRANSPORT: LAGVT,LStation7750;

Model statements for module: Station 113

STATION, RStation7750;

DELAY: 0.0, VA:NEXT(1216$);

Model statements for module: Decide 1475

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

ASSIGN: SpeedatRStation7750=VTU(LAGVT,LTruck#):NEXT(1188$);

Model statements for module: Decide 1463

BRANCH, 1:
If, Entity.Type==RCar, 3200$, Yes:
Else, 3201$, Yes;

ASSIGN: Decide 1463.NumberOut True=Decide 1463.NumberOut True + 1:NEXT(1187$);

ASSIGN: Decide 1463.NumberOut False=Decide 1463.NumberOut False + 1:NEXT(1190$);

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;

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;

1217$ ASSIGN: Speed at RStation7800 = VTU(RAGV,RCar#):NEXT(1195$);

1218$ ASSIGN: Speed at RStation7800 = VTU(RAGVT,RTruck#):NEXT(1195$);

1219$ ASSIGN: Speed at RStation7800 = VT(LAGV):NEXT(1195$);

Model statements for module: Station 115

Model statements for module: Decide 1468

Model statements for module: Assign 175

Model statements for module: Assign 176

Model statements for module: Assign 177
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
Model statements for module: Decide 1480

1234$ BRANCH, 1:
If, SpeedatRStation7700 >= VTU(LAGV, LCar#), Yes;
Else, 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, Yes;
If, SpeedatRStation7700 > 55 && SpeedatRStation7700 <= 75, Yes;
If, SpeedatRStation7700 > 35 && SpeedatRStation7700 <= 55, Yes;
If, SpeedatRStation7700 > 10 && SpeedatRStation7700 <= 35, Yes;
Else, Yes;

Model statements for module: Decide 1487

1259$ BRANCH, 1:
If, NL(RL16) && NZ(RL17, 1) && NZ(RL15, 10) == 0, Yes;
Else, 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, 3248$, Yes;
   Else, 3249$, 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$);
TRANSPORT: LAGV,RStation7800;

TRANSPORT: LAGV,LStation7800;

Model statements for module: Decide 1489

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,

ASSIGN: CheckGapL7750Speed75less.NumberOut True = CheckGapL7750Speed75less.NumberOut True + 1
:NEXT(1264$);

ASSIGN: CheckGapL7750Speed55_75less.NumberOut False = CheckGapL7750Speed55_75less.NumberOut False + 1
:NEXT(1265$);

TRANSPORT: LAGV,RStation7800;

TRANSPORT: LAGV,LStation7800;

Model statements for module: Decide 1490

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,

ASSIGN: CheckGapL7750Speed55_75less.NumberOut True = CheckGapL7750Speed55_75less.NumberOut True + 1
:NEXT(1267$);

ASSIGN: CheckGapL7750Speed55_75less.NumberOut False = CheckGapL7750Speed55_75less.NumberOut False + 1
:NEXT(1268$);

TRANSPORT: LAGV,RStation7800;

TRANSPORT: LAGV,LStation7800;

Model statements for module: Decide 1491

;
1269$ BRANCH, 1:
If,NZ(RL13,2)&&&NZ(RL14,1)&&&NL(RL15)&&&NZ(RL16,1)&&&NZ(RL16,2)&&&NZ(RL16,3) == 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, Speed at R Station 7700 > 75, 1279$, Yes:
    If, Speed at R Station 7700 > 55 && Speed at R Station 7700 <= 75, 1280$, Yes:
    If, Speed at R Station 7700 > 35 && Speed at R Station 7700 <= 55, 1281$, Yes:
    If, Speed at R Station 7700 > 10 && Speed at R Station 7700 <= 35, 1282$, 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, R Station 7800;

1249$ TRANSPORT: LAGVT, L Station 7800;

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, R Station 7800;

1241$ TRANSPORT: LAGVT, L Station 7800;
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$);
388

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$);
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,
        Yes:
            ASSIGN: CheckGapL7750Speed55_75lesst.NumberOut
                True=CheckGapL7750Speed55_75lesst.NumberOut True + 1
                : NEXT(1292$);
        Else, Yes:
            ASSIGN: CheckGapL7750Speed55_75lesst.NumberOut
                False=CheckGapL7750Speed55_75lesst.NumberOut False + 1
                : NEXT(1293$);
    Else, Yes:
        ASSIGN: CheckGapL7750Speed35_55lesst.NumberOut
            True=CheckGapL7750Speed35_55lesst.NumberOut True + 1
            : NEXT(1294$);
        Else, Yes:
            ASSIGN: CheckGapL7750Speed35_55lesst.NumberOut
                False=CheckGapL7750Speed35_55lesst.NumberOut False + 1
                : NEXT(1295$);

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,
        Yes:
            ASSIGN: CheckGapL7750Speed35_55lesst.NumberOut
                True=CheckGapL7750Speed35_55lesst.NumberOut True + 1
                : NEXT(1294$);
        Else, Yes:
            ASSIGN: CheckGapL7750Speed35_55lesst.NumberOut
                False=CheckGapL7750Speed35_55lesst.NumberOut False + 1
                : NEXT(1295$);
    Else, Yes:
        ASSIGN: CheckGapL7750Speed35_55lesst.NumberOut
            True=CheckGapL7750Speed35_55lesst.NumberOut True + 1
            : NEXT(1294$);
        Else, Yes:
            ASSIGN: CheckGapL7750Speed35_55lesst.NumberOut
                False=CheckGapL7750Speed35_55lesst.NumberOut False + 1
                : NEXT(1295$);
Model statements for module: Decide 1504

1. If, NZ(RL13,2) && NZ(RL14,1) && NZ(RL14,2) && NL(RL15) && NZ(RL16,1) && NZ(RL16,2) && NZ(RL16,3) == 0, Yes; Else, Yes;

2. ASSIGN: CheckGapL7750Speed10_35lesst.NumberOut True = CheckGapL7750Speed10_35lesst.NumberOut True + 1
   : NEXT(1296$);

3. ASSIGN: CheckGapL7750Speed10_35lesst.NumberOut False = CheckGapL7750Speed10_35lesst.NumberOut False + 1
   : NEXT(1297$);

TRANSPORT: LAGVT,RStation7800;

TRANSPORT: LAGVT,LStation7800;

Model statements for module: Station 118

STATION, LStation7800;

DELAY: 0.0,, VA:NEXT(1301$);

Model statements for module: Decide 1506

BRANCH, 1:
  If, Entity.Type == LCar, Yes;
  Else, Yes;

ASSIGN: checkvehicletype7800.NumberOut True = checkvehicletype7800.NumberOut True + 1
  : NEXT(1302$);

ASSIGN: checkvehicletype7800.NumberOut False = checkvehicletype7800.NumberOut False + 1
  : NEXT(1303$);

Model statements for module: Decide 1507

BRANCH, 1:
  If, SpeedatRStation7750 >= VTU(LAGV, LCar#), Yes;
  Else, Yes;

ASSIGN: Decide 1507.NumberOut True = Decide 1507.NumberOut True + 1
  : NEXT(1304$);
Model statements for module: Decide 1507

391

ASSIGN: Decide 1507.NumberOut False = Decide 1507.NumberOut False + 1; NEXT(1330$);

1304$ BRANCH, 1:
   If, SpeedatRStation7750 > 75, 1305$, Yes;
   If, SpeedatRStation7750 > 55 && SpeedatRStation7750 <= 75, 1318$, Yes;
   If, SpeedatRStation7750 > 35 && SpeedatRStation7750 <= 55, 1321$, Yes;
   Else, 1327$, Yes;

Model statements for module: Decide 1509

1327$ BRANCH, 1:
   If, NL(RL17) && NL(RL18) && NL(RL19) && NZ(RL20, 1) && NZ(RL20, 2)
   && NZ(RL20, 3) && NZ(RL20, 4) && NZ(RL16, 1) && NZ(RL16, 6) && NZ(RL16, 7) && NZ(RL16, 8)
   && NZ(RL16, 10) == 0, 3301$, Yes;
   Else, 3302$, Yes;

3296$ 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, 1) && NZ(RL16, 6) && NZ(RL16, 7) && NZ(RL16, 8)
   && 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

BRANCH, 1:
If, NL(RL17) && NL(RL18) && NL(RL19) && NZ(RL16,7) && NZ(RL16,9) && NZ(RL16,10) == 0, Yes:
Else, Yes;
ASSIGN: CheckGapL7800Speed55_75.NumberOut
True = CheckGapL7800Speed55_75.NumberOut True + 1; NEXT(1319$);
ASSIGN: CheckGapL7800Speed55_75.NumberOut
False = CheckGapL7800Speed55_75.NumberOut False + 1; NEXT(1320$);
TRANSPORT: LAGV, RStation7850;
TRANSPORT: LAGV, LStation7850;

Model statements for module: Decide 1512

BRANCH, 1:
If, NL(RL18) && NL(RL19) && NZ(RL16,7) && NZ(RL16,9) && NZ(RL16,10) == 0, Yes:
Else, Yes;
ASSIGN: CheckGapL7800Speed35_55.NumberOut
True = CheckGapL7800Speed35_55.NumberOut True + 1; NEXT(1322$);
ASSIGN: CheckGapL7800Speed35_55.NumberOut
False = CheckGapL7800Speed35_55.NumberOut False + 1; NEXT(1323$);
TRANSPORT: LAGV, RStation7850;
TRANSPORT: LAGV, LStation7850;

Model statements for module: Decide 1513

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$, Yes:
Else, Yes;
ASSIGN: CheckGapL7800Speed10_35.NumberOut
True = CheckGapL7800Speed10_35.NumberOut True + 1; NEXT(1325$);
ASSIGN: CheckGapL7800Speed10_35.NumberOut
False=CheckGapL7800Speed10_35.NumberOut False + 1:NEXT(1326$);

TRANSPORT: LAGV,RStation7850;

TRANSPORT: LAGV,LStation7850;

Model statements for module: Decide 1515

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

BRANCH, 1:
If,NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)== 0,3311$,Yes:
Else,3312$,Yes;

ASSIGN: CheckGapL7800Speed10less.NumberOut
True=CheckGapL7800Speed10less.NumberOut True + 1:NEXT(1344$);

ASSIGN: CheckGapL7800Speed10less.NumberOut
False=CheckGapL7800Speed10less.NumberOut False + 1:NEXT(1345$);

TRANSPORT: LAGV,RStation7850;

TRANSPORT: LAGV,LStation7850;

Model statements for module: Decide 1516

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;

ASSIGN: CheckGapL7800Speed75less.NumberOut
True=CheckGapL7800Speed75less.NumberOut True + 1:NEXT(1332$);
ASSIGN: CheckGapL7800Speed75less.NumberOut
False=CheckGapL7800Speed75less.NumberOut False + 1:NEXT(1333$);

TRANSPORT: LAGV,RStation7850;

TRANSPORT: LAGV,LStation7850;

Model statements for module: Decide 1517

BRANCH, 1:

If,NZ(RL14,1)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)== 0,3315$;

Else,3316$,Yes:

ASSIGN: CheckGapL7800Speed55_75less.NumberOut
True=CheckGapL7800Speed55_75less.NumberOut True + 1
:NEXT(1335$);

ASSIGN: CheckGapL7800Speed55_75less.NumberOut
False=CheckGapL7800Speed55_75less.NumberOut False + 1
:NEXT(1336$);

TRANSPORT: LAGV,RStation7850;

TRANSPORT: LAGV,LStation7850;

Model statements for module: Decide 1518

BRANCH, 1:

If,NZ(RL14,2)&&NL(RL15)&&NL(RL16)&&NZ(RL17,1)&&NZ(RL17,2)&&NZ(RL17,3)== 0,3317$;

Else,3318$,Yes:

ASSIGN: CheckGapL7800Speed35_55less.NumberOut
True=CheckGapL7800Speed35_55less.NumberOut True + 1
:NEXT(1338$);

ASSIGN: CheckGapL7800Speed35_55less.NumberOut
False=CheckGapL7800Speed35_55less.NumberOut False + 1
:NEXT(1339$);

TRANSPORT: LAGV,RStation7850;

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;
1351$
$BRANCH, 1:
  If, NL(RL17)\&\&NL(RL18)\&\&NZ(RL16,8) \&\& NZ(RL16,9) \&\& NZ(RL16,10) == 0, 3325$, Yes:
  Else, 3326$, Yes;
ASSIGN: CheckGapL7800Speed10t.NumberOut
True=CheckGapL7800Speed10t.NumberOut True + 1:NEXT(1316$);

ASSIGN: CheckGapL7800Speed10t.NumberOut
False=CheckGapL7800Speed10t.NumberOut False + 1:NEXT(1317$);

TRANSPORT: LAGVT,RStation7850;

TRANSPORT: LAGVT,LStation7850;

Model statements for module: Decide 1522

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;

ASSIGN: CheckGapL7800Speed75t.NumberOut
True=CheckGapL7800Speed75t.NumberOut True + 1:NEXT(1308$);

ASSIGN: CheckGapL7800Speed75t.NumberOut
False=CheckGapL7800Speed75t.NumberOut False + 1:NEXT(1309$);

TRANSPORT: LAGVT,RStation7850;

TRANSPORT: LAGVT,LStation7850;

Model statements for module: Decide 1523

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(RL16.8)\&\&NZ(RL16.9)\&\&NZ(RL16.10) == 0,
3329$,Yes:
Else,3330$,Yes;

ASSIGN: CheckGapL7800Speed55_75t.NumberOut
True=CheckGapL7800Speed55_75t.NumberOut True + 1:NEXT(1311$);

ASSIGN: CheckGapL7800Speed55_75t.NumberOut
False=CheckGapL7800Speed55_75t.NumberOut False + 1:NEXT(1310$);

TRANSPORT: LAGVT,RStation7850;
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, Yes:
        Else, 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(RL17,3) == 0,
        Yes:
        Else, 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(RL17,3) == 0,
        Yes:
        Else, Yes;

3341$, Yes:
        Else, Yes;
3341$ 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)&&(NL(RL15)&&(NL(RL16)&&(NL(RL17,1)&&(NL(RL17,2)&&(NL(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)&&(NL(RL17,1)&&(NL(RL17,2)&&(NL(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;
TRANSPORT: LAGVT, LStation7850;

Model statements for module: Station 119

STATION, LStation7850;
DELAY: 0.0, VA:NEXT(1369$);

Model statements for module: Decide 1533

BRANCH, 1:
  If, Entity.Type == LCar, 3350$, Yes:
  Else, 3351$, Yes;
ASSIGN: checkvehicletype7850.NumberOut True = checkvehicletype7850.NumberOut True + 1:NEXT(1370$);
ASSIGN: checkvehicletype7850.NumberOut False = checkvehicletype7850.NumberOut False + 1:NEXT(1371$);

Model statements for module: Decide 1534

BRANCH, 1:
  If, SpeedatRStation7800 >= VTU(LAGV, LCar#), 3352$, Yes:
  Else, 3353$, Yes;
ASSIGN: Decide 1534.NumberOut True = Decide 1534.NumberOut True + 1:NEXT(1372$);
ASSIGN: Decide 1534.NumberOut False = Decide 1534.NumberOut False + 1:NEXT(1398$);

Model statements for module: Decide 1536

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$);
ASSIGN: CheckGapL7850Speed55_75.NumberOut
False=CheckGapL7850Speed55_75.NumberOut False + 1:NEXT(1388$);

TRANSPORT: LAGV,RStation7900;

TRANSPORT: LAGV,LStation7900;

; ; Model statements for module: Decide 1539 ; ;

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;

ASSIGN: CheckGapL7850Speed35_55.NumberOut
True=CheckGapL7850Speed35_55.NumberOut True + 1:NEXT(1390$);

ASSIGN: CheckGapL7850Speed35_55.NumberOut
False=CheckGapL7850Speed35_55.NumberOut False + 1:NEXT(1391$);

TRANSPORT: LAGV,RStation7900;

TRANSPORT: LAGV,LStation7900;

; ; Model statements for module: Decide 1540 ; ;

BRANCH, 1:
If,
NL(RL18) && NZ(RL19,1) && NZ(RL19,2) && NZ(RL19,3) && NZ(RL19,4)
&& NZ(RL19,5) && NZ(RL19,6) && NZ(RL19,7) && NZ(RL17,10) == 0,
3364$,Yes:
Else,3365$,Yes;

ASSIGN: CheckGapL7850Speed10_35.NumberOut
True=CheckGapL7850Speed10_35.NumberOut True + 1:NEXT(1393$);

ASSIGN: CheckGapL7850Speed10_35.NumberOut
False=CheckGapL7850Speed10_35.NumberOut False + 1:NEXT(1394$);

TRANSPORT: LAGV,RStation7900;

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) \&\& NL(RL18,1) \&\& NZ(RL18,2) \&\& NZ(RL18,3) == 0,

3376$ Yes:

Else, 3377$ Yes;
ASSIGN:  CheckGapL7850Speed10_35less.NumberOut
True=CheckGapL7850Speed10_35less.NumberOut True + 1
:NEXT(1409$);

ASSIGN:  CheckGapL7850Speed10_35less.NumberOut
False=CheckGapL7850Speed10_35less.NumberOut False + 1
:NEXT(1410$);

TRANSPORT:  LAGV,RStation7900;

TRANSPORT:  LAGV,LStation7900;

;  ;  ; Model statements for module:  Decide 1535 ;  ;

BRANCH,  1:
  If,SpeedatRStation7800 >= VTU(LAGVT,LTruck#),3378$,Yes:
  Else,3379$,Yes;

ASSIGN:  Decide 1535.NumberOut True=Decide 1535.NumberOut True + 1:NEXT(1414$);

ASSIGN:  Decide 1535.NumberOut False=Decide 1535.NumberOut False + 1:NEXT(1420$);

;  ;  ; Model statements for module:  Decide 1548 ;  ;

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 ;  ;

BRANCH,  1:
  If,NL(RL18)&&NL(RL19)&&NZ(RL17,9) && NZ(RL17,10) == 0,3382$,Yes:
  Else,3383$,Yes;

ASSIGN:  CheckGapL7850Speed10t.NumberOut
True=CheckGapL7850Speed10t.NumberOut True + 1:NEXT(1384$);

ASSIGN:  CheckGapL7850Speed10t.NumberOut
False=CheckGapL7850Speed10t.NumberOut False + 1:NEXT(1385$);

TRANSPORT:  LAGVT,RStation7900;
TRANSFORM: 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,
    ASSIGN: CheckGapL7850Speed75t.NumberOut True = CheckGapL7850Speed75t.NumberOut True + 1:NEXT(1376$);
    Else, ASSIGN: CheckGapL7850Speed75t.NumberOut False = CheckGapL7850Speed75t.NumberOut False + 1:NEXT(1377$);
    1376$ TRANSPORT: LAGVT,RStation7900;
    1377$ TRANSPORT: LAGVT,LStation7900;

; Model statements for module: Decide 1550
;
1416$ BRANCH, 1:
    If,
    NL(RL18) && NL(RL19) && NL(RL20) && NZ(RL17,6) && NZ(RL17,7) && NZ(RL17,8) && NZ(RL17,9) && NZ(RL17,10) == 0,
    ASSIGN: CheckGapL7850Speed55_75t.NumberOut True = CheckGapL7850Speed55_75t.NumberOut True + 1:NEXT(1379$);
    Else, ASSIGN: CheckGapL7850Speed55_75t.NumberOut False = CheckGapL7850Speed55_75t.NumberOut False + 1:NEXT(1378$);
    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, Speed at RStation7800 > 75, 1421$, Yes:
If, Speed at RStation7800 > 55 && Speed at RStation7800 <= 75, 1422$, Yes:
If, Speed at RStation7800 > 35 && Speed at RStation7800 <= 55, 1423$, Yes:
If, Speed at RStation7800 > 10 && Speed at RStation7800 <= 35, 1424$, 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)&amp;&amp;NL(RL16)&amp;&amp;NL(RL17)&amp;&amp;NZ(RL18,1)&amp;&amp;NZ(RL18,2)&amp;&amp;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)&amp;&amp;NZ(RL15,7)&amp;&amp;NZ(RL15,8)&amp;&amp;NZ(RL15,9)&amp;&amp;NZ(RL15,10)&amp;&amp;NL(RL16)&amp;&amp;NL(RL17)&amp;&amp;NZ(RL18,1)&amp;&amp;NZ(RL18,2)&amp;&amp;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  
;
Model statements for module: Decide 1560

BRANCH, 1:
  If, Entity.Type==LCar, Yes:
  Else, Yes;

ASSIGN: checkvehicletype7900.NumberOut True = checkvehicletype7900.NumberOut True + 1
         NEXT(1438$);

ASSIGN: checkvehicletype7900.NumberOut False = checkvehicletype7900.NumberOut False + 1
         NEXT(1439$);

Model statements for module: Decide 1561

BRANCH, 1:
  If, SpeedatRStation7850 >= VTU(LAGV, LCar#), Yes:
  Else, Yes;

ASSIGN: Decide 1561.NumberOut True = Decide 1561.NumberOut True + 1
         NEXT(1440$);

ASSIGN: Decide 1561.NumberOut False = Decide 1561.NumberOut False + 1
         NEXT(1466$);

Model statements for module: Decide 1563

BRANCH, 1:
  If, SpeedatRStation7850 > 75, Yes:
  If, SpeedatRStation7850 > 55 && SpeedatRStation7850 <= 75, Yes:
  If, SpeedatRStation7850 > 35 && SpeedatRStation7850 <= 55, Yes:
  If, SpeedatRStation7850 > 10 && SpeedatRStation7850 <= 35, Yes:
  Else, Yes;

ASSIGN: CheckGapL7900Speed10.NumberOut True = CheckGapL7900Speed10.NumberOut True + 1
         NEXT(1464$);
ASSIGN: CheckGapL7900Speed10.NumberOut
False=CheckGapL7900Speed10.NumberOut False + 1:NEXT(1465$);

TRANSPORT: LAGV,RStation7950;

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$);

TRANSPORT: LAGV,RStation7950;

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$);

TRANSPORT: LAGV,RStation7950;

TRANSPORT: LAGV,LStation7950;

; ; Model statements for module: Decide 1566
2: 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, 3: Yes;
Else, 4: Yes;
3: ASSIGN: CheckGapL7900Speed35_55.NumberOut True = CheckGapL7900Speed35_55.NumberOut True + 1: NEXT(1458$);
4: ASSIGN: CheckGapL7900Speed35_55.NumberOut False = CheckGapL7900Speed35_55.NumberOut False + 1: NEXT(1459$);
1458$ TRANSPORT: LAGV, RStation7950;
1459$ TRANSPORT: LAGV, LStation7950;

; ; Model statements for module: Decide 1567
; 1: BRANCH, 1:
If, NL(RL19) && NZ(RL20,1) && NZ(RL20,2) && NZ(RL20,3) && NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) == 0, 2: Yes;
Else, 3: Yes;
2: ASSIGN: CheckGapL7900Speed10_35.NumberOut True = CheckGapL7900Speed10_35.NumberOut True + 1: NEXT(1461$);
3: ASSIGN: CheckGapL7900Speed10_35.NumberOut False = CheckGapL7900Speed10_35.NumberOut False + 1: NEXT(1462$);
1461$ TRANSPORT: LAGV, RStation7950;
1462$ TRANSPORT: LAGV, LStation7950;

; ; Model statements for module: Decide 1569
; 1: BRANCH, 1:
If, SpeedatRStation7850 > 75, 2: Yes:
If, SpeedatRStation7850 > 55 && SpeedatRStation7850 <= 75, 3: Yes:
If, SpeedatRStation7850 > 35 && SpeedatRStation7850 <= 55, 4: Yes:
If, SpeedatRStation7850 > 10 && SpeedatRStation7850 <= 35, 5: Yes:
Else, 6: Yes;
2: ;
3: ;
4: ;
5: ;
6: ; Model statements for module: Decide 1574
;
1479$ \text{BRANCH, 1:} \\
\quad \text{If,} \text{NL(RL18) && NZ(RL19,1) && NZ(RL19,2) && NZ(RL19,3) == 0, 3425$, Yes:} \\
\quad \text{Else, 3426$, Yes;}} \\
3425$ \text{ASSIGN: CheckGapL7900Speed10less.NumberOut} \\
\quad \text{True = CheckGapL7900Speed10less.NumberOut True + 1; NEXT(1480$);} \\
3426$ \text{ASSIGN: CheckGapL7900Speed10less.NumberOut} \\
\quad \text{False = CheckGapL7900Speed10less.NumberOut False + 1; NEXT(1481$);} \\
1480$ \text{TRANSPORT: LAGV, RStation7950;} \\
1481$ \text{TRANSPORT: LAGV, LStation7950;} \\
\; \\
\; \\
\; \text{Model statements for module: Decide 1570} \\
\; \\
1467$ \text{BRANCH, 1:} \\
\quad \text{If,} \text{NL(RL15) && NL(RL16) && NL(RL17) && NL(RL18) && NZ(RL19,1) && NZ(RL19,2) && NZ(RL19,3) ==} \\
\quad \text{0, 3427$, Yes:} \\
\quad \text{Else, 3428$, Yes;}} \\
3427$ \text{ASSIGN: CheckGapL7900Speed75less.NumberOut} \\
\quad \text{True = CheckGapL7900Speed75less.NumberOut True + 1; NEXT(1468$);} \\
3428$ \text{ASSIGN: CheckGapL7900Speed75less.NumberOut} \\
\quad \text{False = CheckGapL7900Speed75less.NumberOut False + 1; NEXT(1469$);} \\
1468$ \text{TRANSPORT: LAGV, RStation7950;} \\
1469$ \text{TRANSPORT: LAGV, LStation7950;} \\
\; \\
\; \\
\; \text{Model statements for module: Decide 1571} \\
\; \\
1470$ \text{BRANCH, 1:} \\
\quad \text{If,} \text{NL(RL16) && NL(RL17) && NL(RL18) && NZ(RL19,1) && NZ(RL19,2) && NZ(RL19,3) ==} \\
\quad \text{0, 3429$, Yes:} \\
\quad \text{Else, 3430$, Yes;}} \\
3429$ \text{ASSIGN: CheckGapL7900Speed55_75less.NumberOut} \\
\quad \text{True = CheckGapL7900Speed55_75less.NumberOut True + 1; NEXT(1471$);} \\
3430$ \text{ASSIGN: CheckGapL7900Speed55_75less.NumberOut} \\
\quad \text{False = CheckGapL7900Speed55_75less.NumberOut False + 1; NEXT(1472$);} \\
1471$ \text{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$ \text{BRANCH, 1:} \\
\text{If, SpeedatRStation7850} \geq \text{VTU(LAGVT,LT ruck#)}, 3435$, Yes; \\
\text{Else, 3436$, Yes;} \\
3435$ \text{ASSIGN: } \text{Decide 1562.NumberOut True=Decide 1562.NumberOut True + 1; NEXT(1482$);} \\
3436$ \text{ASSIGN: } \text{Decide 1562.NumberOut False=Decide 1562.NumberOut False + 1; NEXT(1488$);} \\

Model statements for module: Decide 1575

1482$ \text{BRANCH, 1:} \\
\text{If, SpeedatRStation7850 > 75, 1483$, Yes;} \\
\text{If, SpeedatRStation7850 > 55 && SpeedatRStation7850} \leq 75, 1484$, Yes; \\
\text{If, SpeedatRStation7850 > 35 && SpeedatRStation7850} \leq 55, 1485$, Yes; \\
\text{If, SpeedatRStation7850 > 10 && SpeedatRStation7850} \leq 35, 1486$, Yes; \\
\text{Else, 1487$, Yes;} \\
1487$ \text{BRANCH, 1:} \\
\text{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$, Yes;} \\
\text{Else, 3440$, Yes;} \\
3439$ \text{ASSIGN: CheckGapL7900Speed10t.NumberOut True=CheckGapL7900Speed10t.NumberOut True + 1; NEXT(1452$);} \\
3440$ \text{ASSIGN: CheckGapL7900Speed10t.NumberOut False=CheckGapL7900Speed10t.NumberOut False + 1; NEXT(1453$);} \\
1452$ \text{TRANSPORT: LAGVT,RStation7950;} \\
1453$ \text{TRANSPORT: LAGVT,LStation7950;} \\

Model statements for module: Decide 1580

1487$ \text{BRANCH, 1:} \\
\text{If, NL(RL19) && NL(RL20) && NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) == 0, 3441$, Yes;} \\
\text{Else, 3442$, Yes;} \\

Model statements for module: Decide 1576

1483$ \text{BRANCH, 1:} \\
\text{If, NL(RL19) && NL(RL20) && NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) == 0, 3441$, Yes;} \\
\text{Else, 3442$, Yes;}
416

ASSIGN: CheckGapL7900Speed75t.NumberOut
True=CheckGapL7900Speed75t.NumberOut True + 1:NEXT(1444$);

ASSIGN: CheckGapL7900Speed75t.NumberOut
False=CheckGapL7900Speed75t.NumberOut False + 1:NEXT(1445$);

TRANSPORT: LAGVT,RStation7950;

TRANSPORT: LAGVT,LStation7950;

; ; Model statements for module: Decide 1577 ;

BRANCH, 1:
If,NL(RL19)&&NL(RL20)&&NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) == 0,3443$,Yes:
Else,3444$,Yes;

ASSIGN: CheckGapL7900Speed55_75t.NumberOut
True=CheckGapL7900Speed55_75t.NumberOut True + 1:NEXT(1447$);

ASSIGN: CheckGapL7900Speed55_75t.NumberOut
False=CheckGapL7900Speed55_75t.NumberOut False + 1:NEXT(1446$);

TRANSPORT: LAGVT,RStation7950;

TRANSPORT: LAGVT,LStation7950;

; ; Model statements for module: Decide 1578 ;

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;

ASSIGN: CheckGapL7900Speed35_55t.NumberOut
True=CheckGapL7900Speed35_55t.NumberOut True + 1:NEXT(1448$);

ASSIGN: CheckGapL7900Speed35_55t.NumberOut
False=CheckGapL7900Speed35_55t.NumberOut False + 1:NEXT(1449$);

TRANSPORT: LAGVT,RStation7950;

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$,Yes;
1493$  ASSIGN:  CheckGapL7900Speed10lesst.NumberOut
         True=CheckGapL7900Speed10lesst.NumberOut True + 1:NEXT(1502$);
3451$  ASSIGN:  CheckGapL7900Speed10lesst.NumberOut
         False=CheckGapL7900Speed10lesst.NumberOut False + 1:NEXT(1503$);
1502$  TRANSPORT:  LAGVT,RStation7950;
1503$  TRANSPORT:  LAGVT,LStation7950;

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,
      Yes:
      3453$, Yes;
      Else, 3454$, Yes;
   3453$  ASSIGN:  CheckGapL7900Speed75less.NumberOut
   True=CheckGapL7900Speed75less.NumberOut True + 1; NEXT(1494$);
   3454$  ASSIGN:  CheckGapL7900Speed75less.NumberOut
   False=CheckGapL7900Speed75less.NumberOut False + 1; NEXT(1495$);
   1494$  TRANSPORT:  LAGVT,RStation7950;
   1495$  TRANSPORT:  LAGVT,LStation7950;

1490$  BRANCH,  1:
   If,
      NL(RL15) && NL(RL16) && NL(RL17) && NL(RL18) && NZ(RL19,1) && NZ(RL19,2) && NZ(RL19,3) == 0,
      Yes:
      3455$, Yes;
      Else, 3456$, Yes;
   3455$  ASSIGN:  CheckGapL7900Speed55_75less.NumberOut
   True=CheckGapL7900Speed55_75less.NumberOut True + 1;
   : NEXT(1496$);
   3456$  ASSIGN:  CheckGapL7900Speed55_75less.NumberOut
   False=CheckGapL7900Speed55_75less.NumberOut False + 1; NEXT(1497$);
   1496$  TRANSPORT:  LAGVT,RStation7950;
   1497$  TRANSPORT:  LAGVT,LStation7950;

1491$  BRANCH,  1:
   If,
      NL(RL16) && NL(RL17) && NL(RL18) && NZ(RL19,1) && NZ(RL19,2) && NZ(RL19,3) == 0,
      Yes:
      3457$, Yes;
      Else, 3458$, Yes;
   3457$  ASSIGN:  CheckGapL7900Speed35_55less.NumberOut
   True=CheckGapL7900Speed35_55less.NumberOut True + 1;
   : NEXT(1498$);
ASSIGN: CheckGapL7900Speed35_55lesst.NumberOut False=CheckGapL7900Speed35_55lesst.NumberOut False + 1
:NEXT(1499$);

TRANSPORT: LAGVT,RStation7950;

TRANSPORT: LAGVT,LStation7950;

...

Model statements for module: Decide 1585

BRANCH, 1:
If,
NZ(RL16,6)&&NZ(RL16,7)&&NZ(RL16,8)&&NZ(RL16,9)&&NL(RL17)&&NL(RL18 )&&NZ(RL19,1)&&NZ(RL19,2)&&NZ(RL19,3)== 0,

ASSIGN: CheckGapL7900Speed10_35lesst.NumberOut True=CheckGapL7900Speed10_35lesst.NumberOut True + 1
:NEXT(1500$);

ASSIGN: CheckGapL7900Speed10_35lesst.NumberOut False=CheckGapL7900Speed10_35lesst.NumberOut False + 1
:NEXT(1501$);

TRANSPORT: LAGVT,RStation7950;

TRANSPORT: LAGVT,LStation7950;

...

Model statements for module: Station 121

STATION, LStation7950;

DELAY: 0.0,VA:NEXT(1505$);

...

Model statements for module: Decide 1587

BRANCH, 1:
If,Entity.Type==LCar,3464$,Yes:
Else,3465$,Yes;

ASSIGN: checkvehicletype7950.NumberOut True=checkvehicletype7950.NumberOut True + 1:NEXT(1512$);
Model statements for module: Decide 1615

BRANCH, 1:
  If, NQ(QueueatTaperT) + NQ(QueueatTaperC) <= 3, Yes:
  If, NQ(QueueatTaperT) + NQ(QueueatTaperC) > 3 && NQ(QueueatTaperT) + NQ(QueueatTaperC) <= 5, Yes:
  If, NQ(QueueatTaperT) + NQ(QueueatTaperC) > 5 && NQ(QueueatTaperT) + NQ(QueueatTaperC) <= 10, Yes:
  Else, Yes;

Model statements for module: Decide 1641

BRANCH, 1:
  If, SpeedatRStation7900 > 75, Yes:
  If, SpeedatRStation7900 > 55 && SpeedatRStation7900 <= 75, Yes:
  If, SpeedatRStation7900 > 35 && SpeedatRStation7900 <= 55, Yes:
  If, SpeedatRStation7900 > 10 && SpeedatRStation7900 <= 35, Yes:
  If, SpeedatRStation7900 <= 10, Yes:
  Else, Yes;

Model statements for module: Hold 47

QUEUE, QueueatTaperC;
SCAN: NZ(RL20,1) == 0: NEXT(1598$);

TRANSPORT: LAGV, RStation8100;

Model statements for module: Decide 1642

BRANCH, 1:
  If, NL(RL18) && NL(RL19) && NZ(RL20,1) == 0, Yes:
  Else, Yes;

ASSIGN: Decide 1642.NumberOut True = Decide 1642.NumberOut True + 1: NEXT(1542$);

ASSIGN: Decide 1642.NumberOut False = Decide 1642.NumberOut False + 1: NEXT(1600$);
TRANSPORT: LAGV,RStation8100;

Model statements for module: Decide 1643

BRANCH, 1:
If,NZ(RL18,8) &&NZ(RL18,9) &&NZ(RL18,10) && NL(RL19)&&NZ(RL20,1)== 0, Yes:
Else, Yes;

ASSIGN: Decide 1643.NumberOut True=Decide 1643.NumberOut True + 1:NEXT(1543$);

ASSIGN: Decide 1643.NumberOut False=Decide 1643.NumberOut False + 1:NEXT(1600$);

TRANSPORT: LAGV,RStation8100;

Model statements for module: Decide 1644

BRANCH, 1:
If,NZ(RL18,10) && NL(RL19)&&NZ(RL20,1)== 0, Yes:
Else, Yes;

ASSIGN: Decide 1644.NumberOut True=Decide 1644.NumberOut True + 1:NEXT(1544$);

ASSIGN: Decide 1644.NumberOut False=Decide 1644.NumberOut False + 1:NEXT(1600$);

TRANSPORT: LAGV,RStation8100;

Model statements for module: Decide 1645

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, Yes:
Else, Yes;

ASSIGN: Decide 1645.NumberOut True=Decide 1645.NumberOut True + 1:NEXT(1545$);

ASSIGN: Decide 1645.NumberOut False=Decide 1645.NumberOut False + 1:NEXT(1600$);
TRANSPORT: LAGV,RStation8100;

Model statements for module: Decide 1646

BRANCH, 1:
  If,NZ(RL19,7) && NZ(RL19,8) && NZ(RL19,9) && NZ(RL19,10) && NZ(RL20,1) == 0, Yes:
  Else, Yes;

ASSIGN: Decide 1646.NumberOut True = Decide 1646.NumberOut True + 1:NEXT(1546$);

ASSIGN: Decide 1646.NumberOut False = Decide 1646.NumberOut False + 1:NEXT(1600$);

TRANSPORT: LAGV,RStation8100;

Model statements for module: Decide 1590

BRANCH, 1:
  If, SpeedatRStation7900 > 75, Yes:
  If, SpeedatRStation7900 > 55 && SpeedatRStation7900 <= 75, Yes:
  If, SpeedatRStation7900 > 35 && SpeedatRStation7900 <= 55, Yes:
  If, SpeedatRStation7900 > 10 && SpeedatRStation7900 <= 35, Yes:
  If, SpeedatRStation7900 <= 10, Yes:
  Else, Yes;

ASSIGN: Decide 1624.NumberOut True = Decide 1624.NumberOut True + 1:NEXT(1507$);

ASSIGN: Decide 1624.NumberOut False = Decide 1624.NumberOut False + 1:NEXT(1600$);

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;

1525$ 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;

1526$ BRANCH, 1:
  If, NZ(RL18,6) && NZ(RL18,7) && NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) && NZ(RL19) && NZ(RL20,1) && NZ(RL20,2) == 0,
  3496$, Yes:
  Else, 3497$, Yes;
\[\text{ASSIGN: } \text{Decide 1631.NumberOut True} = \text{Decide 1631.NumberOut True} + 1: \text{NEXT(1521\$)};\]

\[\text{ASSIGN: } \text{Decide 1631.NumberOut False} = \text{Decide 1631.NumberOut False} + 1: \text{NEXT(1600\$)};\]

\[\text{TRANSPORT: } \text{LAGV,RStation8100};\]

; Model statements for module: Decide 1632

\[\text{BRANCH, 1:}\]
\[\quad \text{If,} \text{NZ(RL18,8) && NZ(RL18,9) && NZ(RL18,10) && NL(RL19) && NZ(RL20,1) && NZ(RL20,2)} == 0, 3498\$, \text{Yes};\]
\[\quad \text{Else,} 3499\$, \text{Yes};\]

\[\text{ASSIGN: } \text{Decide 1632.NumberOut True} = \text{Decide 1632.NumberOut True} + 1: \text{NEXT(1522\$)};\]

\[\text{ASSIGN: } \text{Decide 1632.NumberOut False} = \text{Decide 1632.NumberOut False} + 1: \text{NEXT(1600\$)};\]

\[\text{TRANSPORT: } \text{LAGV,RStation8100};\]

; Model statements for module: Decide 1633

\[\text{BRANCH, 1:}\]
\[\quad \text{If,} \text{NZ(RL18,5) && NL(RL19) && NZ(RL20,1)} == 0, 3500\$, \text{Yes};\]
\[\quad \text{Else,} 3501\$, \text{Yes};\]

\[\text{ASSIGN: } \text{Decide 1633.NumberOut True} = \text{Decide 1633.NumberOut True} + 1: \text{NEXT(1523\$)};\]

\[\text{ASSIGN: } \text{Decide 1633.NumberOut False} = \text{Decide 1633.NumberOut False} + 1: \text{NEXT(1600\$)};\]

\[\text{TRANSPORT: } \text{LAGV,RStation8100};\]

; Model statements for module: Decide 1634

\[\text{BRANCH, 1:}\]
\[\quad \text{If,} \text{NZ(RL19,4) && NZ(RL19,5) && NZ(RL19,6) && NZ(RL19,7) && NZ(RL19,8) && NZ(RL19,9) && NZ(RL19,10) && NZ(RL20,1)} == 0, 3502\$, \text{Yes};\]
\[\quad \text{Else,} 3503\$, \text{Yes};\]
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, Yes:
    If, NQ(QueueatTaperT) + NQ(QueueatTaperC) > 3 && NQ(QueueatTaperT) + NQ(QueueatTaperC) <= 5, Yes:
    If, NQ(QueueatTaperT) + NQ(QueueatTaperC) > 5 && NQ(QueueatTaperT) + NQ(QueueatTaperC) <= 10, Yes:
    Else, Yes;

1563$  BRANCH,  1:
    If, SpeedatRStation7900 > 75, Yes:
    If, SpeedatRStation7900 > 55 && SpeedatRStation7900 <= 75, Yes:
    If, SpeedatRStation7900 > 35 && SpeedatRStation7900 <= 55, Yes:
    If, SpeedatRStation7900 > 10 && SpeedatRStation7900 <= 35, Yes:
    If, SpeedatRStation7900 <= 10, Yes:
    Else, Yes;

1596$  QUEUE,  QueueatTaperT:
    SCAN:  NZ(RL 20, 1) == 0:NEXT(1599$);

1599$  TRANSPORT:  LAGVT,RStation8100;

1569$  BRANCH,  1:
    If, NL(RL17) && NL(RL18) && NL(RL19) && NZ(RL20, 1) == 0, Yes:
    Else, 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;
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(RL19) && 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$);
1:NEXT(1596$);
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;
1553$ TRANSPORT: LAGVT,RStation8100;

; ;
; ; 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)== 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:
    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: LAGVT,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:
  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$, Yes:
Else, Yes;
3561$ 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, Yes:
    Else, 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, Yes:
    Else, 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(Queueat TaperC) + NQ(Queueat TaperT):
       CumuWaitTime = CumuWaitTime + TAVG(Queueat TaperC.WaitingTime) +
        TAVG(Queueat TaperT.WaitingTime):
        NEXT(1632$);

Model statements for module: ReadWrite 1

1632$  WRITE,  Statistics:
        IDENT;
        Entity.Type,
        Entity.StartTime,
        Entity.TranTime,
        Entity.WaitTime,
        NQ(Queueat TaperC) + NQ(Queueat TaperT),
        TAVG(Queueat TaperC.WaitingTime) +
        TAVG(Queueat TaperT.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==L.Car,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
tcheckvehicletype7500.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
tcheckvehicletype7750.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
CheckGalL7600Speed75lesst.NumberOut True
Decide 1283.NumberOut False
CheckGapL7050Speed35_55.NumberOut True
R.Vehicle Enters930_945.NumberOut
CheckGapL7650Speed75.NumberOut False
CheckGapL7500Speed55_75lesst.NumberOut True
tcheckvehicletype7250.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 False
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
CheckGapL6450Speed10less.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_75less.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 Enters 1745_1800. NumberOut
CheckGapL7550Speed10less. NumberOut True
CheckGapL7500Speed55_75less. NumberOut True
CheckGapL7450Speed75less. NumberOut True
DecideVehicleTypeRight0. NumberOut True
Decide 659. NumberOut False
Decide 1480. NumberOut False
CheckGapL7700Speed35_55t. NumberOut True
Decide 1638. NumberOut True
CheckGapL7500Speed75t. NumberOut False
CheckGapL6450Speed75less. NumberOut True
Decide 1638. NumberOut False
CheckGapL7550Speed55_75t. NumberOut True
CheckGapL7250Speed10_35less. NumberOut True
CheckGapL7150Speed10_35. NumberOut False
CheckGapL7600Speed35_55. NumberOut True
Decide 1699. NumberOut True
Decide 1627. NumberOut False
CheckGapL7900Speed35_55less. NumberOut False
CheckGapL7350Speed10less. NumberOut True
CheckGapL6700Speed55_75less. NumberOut True
L. Vehicle Entry 1145_1200. NumberOut
CheckGapL7900Speed10t. NumberOut False
CheckGapL7550Speed35_55less. 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 False
CheckGapL7600Speed75t. NumberOut True
Decide4VehicleTypeR6700. NumberOut False
R. Vehicle Enters 715_730. NumberOut
R. Vehicle Enters 1430_1445. NumberOut
CheckGapL7600Speed10. NumberOut True
Decide2VehicleTypeR6700. NumberOut False
Decide 1009. NumberOut False
CheckGapL7150Speed10less. NumberOut True
R. Vehicle Enters 1100_1115. NumberOut
Decide 1467. NumberOut True
CheckGapL7650Speed75less. NumberOut False
CheckGapL7350Speed55_75. NumberOut True
CheckGapL7150Speed75t. NumberOut True
R. Vehicle Enters 1815_1830. NumberOut
CheckGapL7500Speed75less. NumberOut False
CheckGapL7250Speed35_55. NumberOut False
Decide 928.NumberOut True
CheckGapL6950.Speed10_35less.NumberOut False
CheckGapL7900.Speed10less.NumberOut True
CheckGapL7850.Speed55_75t.NumberOut True
CheckGapL7700.Speed75t.NumberOut True
Decide 1344.NumberOut True
CheckGapL7050.Speed35_55less.NumberOut True
CheckGapL6700.Speed10_35less.NumberOut False
CheckGapL7550.Speed10_35less.NumberOut False
CheckGapL7900.Speed55_75t.NumberOut False
Decide 1721.NumberOut False
L.Vehicle Entry1215_1230.NumberOut
CheckGapL7450.Speed10_35t.NumberOut False
CheckGapL6450.Speed75less.NumberOut True
Decide 1710.NumberOut False
CheckGapL7050.Speed35_55.NumberOut False
CheckGapL6950.Speed10less.NumberOut False
CheckGapL7150.Speed35_55t.NumberOut False
Decide 748.NumberOut True
CheckGapL7800.Speed75t.NumberOut True
Decide 1437.NumberOut False
Decide 1780.NumberOut False
L.Vehicle Entry2000_2015.NumberOut
CheckGapL7050.Speed55_75.NumberOut False
CheckGapL6450.Speed75.NumberOut True
CheckGapL7800.Speed55_75less.NumberOut False
CheckGapL7500.Speed10_35less.NumberOut True
CheckGapL7050.Speed10less.NumberOut False
Decide 1745.NumberOut True
Decide 1723.NumberOut True
Decide 1703.NumberOut False
Decide 1701.NumberOut True
CheckGapL7900.Speed55_75less.NumberOut True
CheckGapL7850.Speed35_55less.NumberOut True
checkvehicletype7800.NumberOut True
CheckGapL6450.Speed10_35less.NumberOut True
L.Vehicle Entry1545_1600.NumberOut
Decide 1776.NumberOut True
Decide 1754.NumberOut True
Decide 1732.NumberOut True
Decide 1644.NumberOut True
CheckGapL7450.Speed10less.NumberOut False
CheckGapL6950.Speed35_55.NumberOut True
Decide 546.NumberOut True
L.Vehicle Entry815_830.NumberOut
CheckGapL7550.Speed10_35less.NumberOut False
Decide 1785.NumberOut True
Decide 1762.NumberOut False
checkvehicletype7600.NumberOut False
Decide 1375.NumberOut False
CheckGapL6700.Speed35_55.NumberOut False
Decide 1631.NumberOut True
CheckGapL7450.Speed75.NumberOut True
CheckGapL7250Speed35_55less.NumberOut False
Decide 1751.NumberOut False
R.Vehicle Enters900_915.NumberOut
Decide 1697.NumberOut True
SpeedatRStation7550
CheckGapL6950Speed751lesst.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_35lesst.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
CheckGapL7650.Speed75t.NumberOut False
CheckGapL7600.Speed55_75less.NumberOut False
CheckGapL7550.Speed10.NumberOut True
CheckGapL7500.Speed35_55less.NumberOut True
Decide 1317.NumberOut True
CheckGapL6450.Speed35_55less.NumberOut False
CheckGapL7050.Speed10_35less.NumberOut True
CheckGapL6950.Speed10_35t.NumberOut False
CheckGapL6450.Speed10t.NumberOut False
Decide 1719.NumberOut False
checkvehicletype7900.NumberOut False
CheckGapL7350.Speed10less.NumberOut False
CheckGapL6950.Speed75t.NumberOut True
Decide 1664.NumberOut False
CheckGapL7900.Speed35_55less.NumberOut False
CheckGapL7850.Speed35_55t.NumberOut True
CheckGapL7600.Speed55_75.NumberOut False
DecideVehicleTypeLeft0.NumberOut True
CheckGapL7850.Speed10_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
CheckGapL7550.Speed75less.NumberOut True
Decide 1778.NumberOut False
CheckGapL7800.Speed75less.NumberOut True
Decide 552.NumberOut True
CheckGapL6700.Speed10.NumberOut False
R.VehicleEnters1900_1915.NumberOut
Decide 1791.NumberOut True
Decide 1668.NumberOut False
Decide 1278.NumberOut True
CheckGapL7250.Speed10less.NumberOut True
Decide 1657.NumberOut False
Decide 1433.NumberOut False
CheckGapL7250.Speed55_75less.NumberOut False
Decide 955.NumberOut False
DecideVehicleTypeLeft0.NumberOut False
CheckGapL7650.Speed35_55t.NumberOut True
CheckGapL7250.Speed75t.NumberOut True
L.VehicleEntry1630_1645.NumberOut
CheckGapL7150.Speed10t.NumberOut False
Decide 1690.NumberOut True
CheckGapL7250.Speed55_75t.NumberOut False
L.VehicleEntry1300_1315.NumberOut
CheckGapL7350.Speed75less.NumberOut True
Decide 1727.NumberOut True
Decide 1705.NumberOut True
CheckGapL6700.Speed75less.NumberOut False
CheckGapL7650Speed55_75t.NumberOut True
CheckGapL7900Speed10_35t.lesst.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
CheckGapL7150Speed75lesst.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_75lesst.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
CheckGapL6950Speed10lesst. NumberOut True
CheckGapL7650Speed55_75.NumberOut True
CheckGapL7450Speed10lesst.NumberOut False
CheckGapL7350Speed75.NumberOut True
CheckGapL7050Speed35_55less. NumberOut False
Decide 1740.NumberOut False
CheckGapL7900Speed10lesst.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
CheckGapL7600 Speed 10lesst. NumberOut False
Decide 1749. NumberOut False
Decide 1507. NumberOut True
Decide 1648. NumberOut True
CheckGapL7900 Speed 35_55. NumberOut True
CheckGapL6950 Speed 10_35lesst. NumberOut True
Decide 1470. NumberOut False
CheckGapL7450 Speed 35_55t. NumberOut True
CheckGapL6450 Speed 10lesst. NumberOut True
R. Vehicle Enters 1330_1345. NumberOut
Decide 1694. NumberOut False
CheckGapL7800 Speed 55_75lesst. NumberOut True
CheckGapL7750 Speed 10_35less. NumberOut True
Decide 1628. NumberOut False
CheckGapL7800 Speed 10lesst. NumberOut False
Decide 981. NumberOut False
R. Vehicle Enters 1000_1015. NumberOut
Decide 1406. NumberOut True
Decide 539. NumberOut False
Decide 874. NumberOut True
R. Vehicle Enters 1715_1730. NumberOut
CheckGapL7750 Speed 35_55lesst. NumberOut False
Decide 1437. NumberOut True
Decide 1290. NumberOut True
SpeedatRStation 7800
CheckGapL7850 Speed 55_75less. NumberOut False
Decide 1507. NumberOut False
CheckGapL7650 Speed 75lesst. NumberOut True
CheckGapL7800 Speed 35_55lesst. NumberOut False
Decide 1468. NumberOut True
CheckGapL7500 Speed 75lesst. NumberOut True
CheckGapL7250 Speed 35_55t. NumberOut True
CheckGapL7150 Speed 10_35less. NumberOut False
L. Vehicle Entry 1115_1130. NumberOut
CheckGapL7800 Speed 10_35less. NumberOut True
CheckGapL7250 Speed 75. NumberOut True
L. Vehicle Entry 1530_1545. NumberOut
CheckGapL7600 Speed 55_75t. NumberOut False
CheckGapL6700 Speed 35_55. NumberOut True
Decide 1345. NumberOut True
CheckGapL7150 Speed 10_35t. NumberOut False
L. Vehicle Entry 1200_1215. NumberOut
CheckGapL7650 Speed 10lesst. NumberOut False
CheckGapL7600 Speed 55_75less. NumberOut True
CheckGapL7150 Speed 55_75lesst. NumberOut False
Decide 1669. NumberOut False
CheckGapL7550 Speed 35_55less. NumberOut True
CheckGapL7500 Speed 35_55. NumberOut True
CheckGapL7050 Speed 35_55t. NumberOut True
CheckGapL6950 Speed 10lesst. NumberOut True
checkvehiclestype 7850. NumberOut False
CheckGapL7850 Speed 10lesst. 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
Decide 1733.NumberOut True
Decide 1372.NumberOut False
CheckGapL7550Speed10_35less.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 Entry 1515_1530. NumberOut
CheckGapL7700Speed75less. NumberOut True
L. Vehicle Entry 1930_1945. NumberOut
Decide 1631. NumberOut False
Decide 1468. NumberOut False
L. Vehicle Entry 1600_1615. NumberOut
L. Vehicle Entry 645_700. NumberOut
CheckGapL7850Speed75t. NumberOut True
CheckGapL7750Speed10less. NumberOut False
CheckGapL7050Speed10. NumberOut False
CheckGapL7750Speed35_55. NumberOut False
CheckGapL7600Speed10_35. NumberOut True
CheckGapL7550Speed75less. NumberOut False
Decide 1745. NumberOut False
L. Vehicle Entry 915_930. NumberOut
CheckGapL7800Speed35_55t. NumberOut True
Decide 1734. NumberOut False
Decide 1690. NumberOut False
CheckGapL7800Speed10less. 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 Entry 1845_1900. NumberOut
CheckGapL7800Speed10_35lesst. NumberOut True
CheckGapL7800Speed55_75t. 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
CheckGapL7850Speed75t. 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
SpeedRStation5750
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_35less.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
CheckGapL6700Speed75less.NumberOut True
CheckGapL7250Speed35_55.NumberOut True
R.Vehicle Enters1230_1245.NumberOut
Decide 1728.NumberOut True
Decide 1706.NumberOut True
Decide 1728.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 False
L.Vehicle Entry1015_1030.NumberOut
CheckGapL7350Speed10_35lesst.NumberOut True
L.Vehicle Entry945_1000.NumberOut
Decide 1535.NumberOut True
CheckGapL7700Speed10_35t.NumberOut True
CheckGapL7450Speed55_75.NumberOut False
Decide 981.NumberOut False
CheckGapL7050Speed35_55less.NumberOut False
CheckGapL7700Speed35_55lesst.NumberOut False
CheckGapL7050Speed10_35.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
CheckGapL7850Speed75less.NumberOut True
CheckGapL7350Speed10_35.NumberOut True
CheckGapL7700Speed75less.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
CheckGapL6700Speed75less.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
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
ccheckvehicletype7750.NumberOut True
Decide 1682.NumberOut True
Decide 1643.NumberOut False
Decide 1643.NumberOut False
CheckGapL7650Speed10less.NumberOut True
Decide 1632.NumberOut False
CheckGapL7850Speed10_35t.NumberOut False
CheckGapL7150Speed35_55t.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
CheckGapL7050Speed75less.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
CheckGapL7150.Speed35_55less.NumberOut False
Decide 954.NumberOut True
Decide 1370.NumberOut True
CheckGapL7500.Speed10_35less.NumberOut True
R.Vehicle Enters1700_1715.NumberOut
CheckGapL6700.Speed10less.NumberOut True
Decide 1739.NumberOut False
CheckGapL7900.Speed10_35less.NumberOut False
Decide 1407.NumberOut True
Decide 4.Vehicle TypeR6700.NumberOut True
Decide 1695.NumberOut False
CheckGapL7850.Speed55_75less.NumberOut True
CheckGapL7750.Speed75less.NumberOut False
CheckGapL7600.Speed35_55less.NumberOut False
Decide 982.NumberOut False
CheckGapL7900.Speed10.NumberOut True
Decide 1438.NumberOut True
Decide 1291.NumberOut True
CheckGapL6450.Speed10t.NumberOut False
L.Vehicle Entry1430_1445.NumberOut
CheckGapL7700.Speed10less.NumberOut False
CheckGapL7700.Speed55_75less.NumberOut False
CheckGapL7600.Speed75less.NumberOut False
CheckGapL7250.Speed35_55less.NumberOut True
CheckGapL7550.Speed10.NumberOut False
CheckGapL7150.Speed10_35less.NumberOut True
L.Vehicle Entry1100_1115.NumberOut
Decide 1469.NumberOut True
CheckGapL7500.Speed55_75less.NumberOut False
checkvehicletype6950.NumberOut True
Decide 3.Vehicle TypeR6700.NumberOut False
L.Vehicle Entry1815_1830.NumberOut
CheckGapL6450.Speed55_75less.NumberOut False
Decide 1688.NumberOut False
CumWaitTime
CheckGapL7250.Speed10.NumberOut True
Dispose 9.NumberOut
CheckGapL7900.Speed35_55t.NumberOut False
CheckGapL7150.Speed10_35lesst.NumberOut False
Decide 1279.NumberOut False
Decide 660.NumberOut True
CheckGapL7650.Speed10.NumberOut False
Decide 1780.NumberOut True
L.Vehicle Entry1345_1400.NumberOut
Decide 550.NumberOut True
checkvehicletype7950.NumberOut False
CheckGapL7900.Speed35_55.NumberOut False
CheckGapL7850.Speed10_35t.NumberOut True
CheckGapL7550.Speed35_55.NumberOut True
CheckGapL7650.Speed55_75lesst.NumberOut False
CheckGapL6950Speed55_75.NumberOut False
CheckGapL7150Speed75less.NumberOut False
CheckGapL7900Speed55_75.NumberOut False
CheckGapL7750Speed10.NumberOut False
Decide 1380.NumberOut False
CheckGapL7050Speed75less.NumberOut True
ccheckvehicletype.NumberOut False
Decide 1747.NumberOut True
Decide 1725.NumberOut True
Decide 1703.NumberOut True
Decide 1701.NumberOut False
CheckGapL6950Speed35_55less.NumberOut True
CheckGapL7450Speed75t.NumberOut False
Decide 658.NumberOut True
CheckGapL7700Speed35_55.NumberOut False
CheckGapL7350Speed75less.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_55less.NumberOut False
CheckGapL7750Speed75.NumberOut True
Decide 1373.NumberOut False
CheckGapL6700Speed55_75less.NumberOut False
SpeedatRStation7850
Decide 1655.NumberOut True
Decide 1633.NumberOut False
CheckGapL7850Speed10.NumberOut False
CheckGapL7700Speed55_75.NumberOut False
Decide 1699.NumberOut True
Decide 660.NumberOut False
CheckGapL7600Speed35_55less.NumberOut False
L.Vehicle Entry1830_1845.NumberOut
CheckGapL7450Speed10_35less.NumberOut False
CheckGapL6450Speed35_55t.NumberOut True
L.Vehicle Entry1500_1515.NumberOut
ccheckvehicletype7600.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
CheckGapL7750 Speed35_55t. NumberOut True
CheckGapL7550 Speed75less. NumberOut True
CheckGapL7500 Speed55_75. NumberOut False
CheckGapL7050 Speed75. NumberOut True
Decide 1344. NumberOut False
CheckGapL7500 Speed10lesst. NumberOut True
CheckGapL7350 Speed55_75less. NumberOut True
CheckGapL7900 Speed10_35. NumberOut True
Decide 1469. NumberOut False
CheckGapL7450 Speed75less. NumberOut True
CheckGapL6700 Speed10_35less. NumberOut True
R. Vehicle Enters 830_845. NumberOut
L. Vehicle Entry 1745_1800. NumberOut
CheckGapL7650 Speed55_75t. NumberOut False
CheckGapL7600 Speed10t. NumberOut True
CheckGapL7750 Speed55_75t. NumberOut True
CheckGapL6950 Speed75less. NumberOut False
CheckGapL7650 Speed10_35. NumberOut True
Decide 1374. NumberOut True
CheckGapL7500 Speed55_75t. NumberOut False
CheckGapL7050 Speed10_35t. NumberOut False
CheckGapL6950 Speed10t. NumberOut True
Decide 1735. NumberOut False
CheckGapL7150 Speed10t. NumberOut True
CheckGapL7700 Speed35_55less. NumberOut True
CheckGapL7550 Speed35_55t. NumberOut True
CheckGapL6450 Speed75. NumberOut False
Decide 1724. NumberOut False
Decide 1680. NumberOut False
CheckGapL7250 Speed75less. NumberOut False
CheckGapL7350 Speed10t. NumberOut False
CheckGapL7250 Speed75less. NumberOut True
CheckGapL7700 Speed10t. NumberOut True
CheckGapL7500 Speed75t. NumberOut False
Decide 1282. NumberOut False
checkvehicletype7350. NumberOut True
Decide 1794. NumberOut False
CheckGapL7850 Speed10_35lesst. NumberOut True
CheckGapL7850 Speed55_75. NumberOut True
CheckGapL7750 Speed35_55lesst. NumberOut True
CheckGapL7550 Speed55_75t. NumberOut True
Decide 1282. NumberOut True
Decide 901. NumberOut False
L. Vehicle Entry 745_800. NumberOut
CheckGapL7850 Speed10. NumberOut True
Decide 1783. NumberOut False
Decide 1728. NumberOut False
CheckGapL7700 Speed10_35less. NumberOut False
CheckGapL7550 Speed35_55lesst. NumberOut True
CheckGapL7450 Speed75lesst. NumberOut False
CheckGapL7750 Speed35_55less. NumberOut False
CheckGapL7250 Speed10t. 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
ccheckvehicletype7500.NumberOut False
CheckGapL7050Speed10less.NumberOut False
CheckGapL7050Speed75lesst.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
ccheckvehicletype.NumberOut True
Decide 1738.NumberOut True
Decide 1716.NumberOut True
Decide 1648.NumberOut False
Decide 1628.NumberOut True
CheckGapL7.750.Speed55_75.less.NumberOut True
CheckGapL7.050.Speed35_55less.NumberOut False
R.Vehicle Enters1045_1100.NumberOut
Decide 1637.NumberOut False
CheckGapL7.600.Speed10.less.NumberOut True
Decide 1769.NumberOut True
CheckGapL7.550.Speed10.less.NumberOut False
CheckGapL7.450.Speed75.less.NumberOut False
Decide 548.NumberOut False
Decide 1637.NumberOut True
Decide 539.NumberOut True
CheckGapL6.700.Speed75.NumberOut True
Dispose 10.NumberOut
CheckGapL7.900.Speed55_75less.NumberOut True
CheckGapL7.650.Speed10_35t.NumberOut True
CheckGapL7.500.Speed75.NumberOut False
Decide 1668.NumberOut True
Decide 1696.NumberOut True
CheckGapL7.900.Speed10_35t.NumberOut False
CheckGapL7.850.Speed35_55less.NumberOut False
CheckGapL7.150.Speed75t.NumberOut False
CheckGapL7.750.Speed35_55t.NumberOut False
Decide 1406.NumberOut False
CheckGapL7.550.Speed75.less.NumberOut False
Decide 982.NumberOut True
CheckGapL7.050.Speed75.NumberOut False
Decide 928.NumberOut False
check vehicletype6.700.NumberOut False
CheckGapL7.650.Speed75.NumberOut True
CheckGapL7.800.Speed55_75less.NumberOut True
CheckGapL7.600.Speed35_55t.NumberOut False
check vehicletype7.800.NumberOut False
CheckGapL7.700.Speed75.NumberOut True
Decide 1008.NumberOut False
Decide 873.NumberOut False
CheckGapL6.450.Speed10_35less.NumberOut False
L.Vehicle Entry1245_1300.NumberOut
Decide Vehicle TypeRight0.NumberOut False
CheckGapL7.500.Speed35_55.NumberOut False
CheckGapL7.450.Speed10_35t.NumberOut True
CheckGapL7.250.Speed10_35less.NumberOut False
CheckGapL7.600.Speed75.NumberOut False
Speedat R Station 7500
Decide 1465.NumberOut False
CheckGapL7.550.Speed10_35less.NumberOut True
CheckGapL7.050.Speed10_35less.NumberOut False
CheckGapL7.050.Speed55_75.NumberOut True
Decide 1689.NumberOut False
CheckGapL7.800.Speed10t.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
CheckGapL6700Speed10less.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
Decide 1744.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
CheckGapL7450.Speed 55_75lesst.NumberOut True
CheckGapL6450.Speed 10_35less.NumberOut False
CheckGapL6450.Speed 10_35.NumberOut True
Decide 1643.NumberOut True
CheckGapL7650.Speed 75lesst.NumberOut False
Decide 545.NumberOut True
Decide 1651.NumberOut True
CheckGapL6450.Speed 10_35less.NumberOut True
CheckGapL6450.Speed 10_35.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
CheckGapL7850.Speed 75lesst.NumberOut False
CheckGapL7350.Speed 55_75t.NumberOut False
CheckGapL7150.Speed 35_55t.NumberOut True
CheckGapL7050.Speed 10.NumberOut True
CheckGapL6950.Speed 75less.NumberOut False
Decide 540.NumberOut False
CheckGapL6700.Speed 10t.NumberOut False
CheckGapL6700.Speed 10_35t.NumberOut True
Decide 1754.NumberOut False
CheckGapL6950.Speed 10_35lesst.NumberOut True
R.Vehicle Enters745_800.NumberOut
Decide 1644.NumberOut False
L.Vehicle Entry930_945.NumberOut
checkvehicletype7650.NumberOut False
R.Vehicle Enters1930_1945.NumberOut
Decide 1633.NumberOut False
CheckGapL7150.Speed 55_75t.NumberOut False
cHECKVEHICLETYPE7150.NumberOut False
CheckGapL6450.Speed 75t.NumberOut False
CheckGapL7700.Speed 75lesst.NumberOut False
Decide 544.NumberOut False
R.Vehicle Enters1600_1615.NumberOut
CheckGapL6450.Speed 10_35.NumberOut False
CheckGapL7450.Speed 10t.NumberOut True
Decide 1747.NumberOut False
Decide 658.NumberOut False
Decide 1481.NumberOut True
Decide 955.NumberOut True
Decide 1736.NumberOut False
CheckGapL7900.Speed 75lesst.NumberOut False
Decide 1692.NumberOut False
L.Vehicle Entry1330_1345.NumberOut
CheckGapL7900Speed10_35lesst.NumberOut True
CheckGapL7800Speed35_55lesst.NumberOut True
Decide 1371.NumberOut True
CheckGapL6450Speed35_55.NumberOut False
Decide 1626.NumberOut False
CheckGapL6950Speed35_55less.NumberOut True
L.Vehicle Entry1000_1015.NumberOut
CheckGapL7850Speed10lesst.NumberOut True
CheckGapL7750Speed75lessst.NumberOut True
CheckGapL7750Speed10_35.NumberOut False
Decide 1380.NumberOut True
R.Vehicle Enters1845_1900.NumberOut
L.Vehicle Entry1715_1730.NumberOut
CheckGapL7700Speed10lesst.NumberOut True
CheckGapL7700Speed55_75lessst.NumberOut True
CheckGapL6700Speed10_35less.NumberOut False
Decide 1795.NumberOut False
CheckGapL7650Speed35_55lessst.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_75lessst.NumberOut True
CheckGapL7250Speed10_35t.NumberOut False
Decide 1 VehicleTypeR6450.NumberOut True
WarmupRunRight.NumberOut
CheckGapL7750Speed35_55less.NumberOut True
CheckGapL7750Speed55_75less.NumberOut False
CheckGapL7550Speed75.NumberOut True
CheckGapL7800Speed10_35.NumberOut True
R.Vehicle Enters1945_2000.NumberOut
CheckGapL7800Speed75lessst.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
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