Title

User's Manual for Transit ITS Simulator (TRAN-ITS)

Authors
Dessouky, Maged
Zhang, Lei
Singh, Ajay
et al.

Publication Date
1999-09-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

User's Manual for Transit ITS Simulator (TRAN-ITS)

Maged Dessouky, Lei Zhang,
Ajay Singh, Randolph Hall
University of Southern California

California PATH Working Paper
UCB-ITS-PWP-99-13

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department Transportation, Federal Highway Administration.

Report for 345

September 1999

ISSN 1055-1417

USERS MANUAL FOR
TRANSIT ITS SIMULATOR
(TRAN-ITS)

June 1,1999

Maged Dessouky
Lei Zhang
Ajay Singh
Randolph Hall

Department of Industrial and Systems Engineering
University of Southern California
Los Angeles, CA 90089-0193

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This report describes a simulation model developed to evaluate the impact of using Intelligent Transportation Systems (ITS), such as Global Positioning Systems (GPS) for bus tracking, on controlling buses in wide-area transit networks. Control strategies with ITS will be compared against those without ITS (i.e., they do not rely on communication or tracking). The model is developed using a general-purpose simulation language, AweSim (Pritsker, **1997).** The simulation model is generic and independent of any dedicated transit network. The model **has high** flexibility and can be used to simulate different kinds of transit networks with varying numbers of bus lines and different travel patterns. The user has the flexibility to input the appropriate control strategy at each bus stop. With this approach an identical replica **of** an actual system can be simulated. This report documents the approach used in the development of the simulation model, the input and output files of the model, and the execution of the simulation model.

EXECUTIVE SUMMARY

Recently, bus transit service providers have begun to adopt Intelligent Transportation Systems (ITS) technologies such as Global Positioning Systems (GPS), Mobile Data Terminals, and Electronic Fare boxes. GPS systems are particularly useful for vehicle tracking and mobile data terminals may be used for passenger counting. These systems taken together have the potential to reduce the cost of providing transportation services through the execution of real-time control strategies, performance monitoring systems and data collection to support service realignment.

The objective of this project, "Efficient Transit Service Through the Application of ITS" (PATH MOU 280), is to investigate the application of ITS technologies to improve the overall efficiency and productivity of transit operations. The perspective is to minimize the cost of achieving a desired level of service or, alternatively, maximizing the service quality within a given budget. Metrics that are investigated include fleet size and service frequency, passenger waiting and travel times, driver hours-of-service, and fare-box collection. The investigation covers field evaluation of the impact of ITS on driver and fleet productivity (documented in a separate working paper) and simulation of transit networks with ITS capabilities.

As part of this project, bus control strategies using ITS are evaluated against those without ITS. Two levels of ITS are considered: (1) system with centralized tracking and (2) system with information on connecting passengers, as well as centralized tracking. By making use of real-time information such as vehicle location, it is expected that control strategies using ITS have the potential to improve connectivity between origins and destinations while reducing passenger waiting times.

This report documents the approach used in the development of the simulation model for analyzing the various control strategies, the input and output files of the model, and the execution of the simulation model. An accompanying report presents the results of the simulation analysis. The simulator developed for this study is an expansion of the simulation model developed by Hall et al **(1997)** to analyze the effectiveness of ITS on scheduling buses at timed transfer terminals. This new simulator considers connectivity of buses at traditional transfer stops as well as at timed transfer terminals. The model has the capability to simulate wide-area transit networks. With this feature, we can study the impact of holding a bus at a particular stop on the waiting time for passengers boarding on subsequent stops. Another major addition to the simulator is the inclusion of passenger entities in the model. In this manner, real-time control strategies taking into account passenger count and waiting times may be considered. Furthermore, including passenger entities permits the explicit modeling of the boarding and debarking process. Hence, the phenomenon of an initial delay in service causing deteriorating service farther down the line due to the increased accumulation of boarding passengers can be studied.

1. INTRODUCTION

Intelligent Transportation Systems (ITS) have been investigated as means to improve the quality of service for automobiles, trucks, buses and other modes. ITS also has the potential to reduce the cost of providing transportation services. With the use of real-time information and bus control, connectivity between origins and destinations may be improved while reducing passenger waiting times. This may enable the transit operator to maintain service levels with a reduced fleet size.

A simulation model of a wide-area transit network is developed to evaluate various real-time control strategies with ITS versus those without ITS. Sample control strategies without ITS include hold a bus at a transit stop until all connecting buses have arrived or never hold a bus past its scheduled departure time. A sample control strategy using ITS is to hold a bus at a transit center if a connecting bus is forecasted to arrive within a time window. The control strategies are evaluated based on several performance metrics, including average bus arrival and departure lateness, average passenger trip time, and average total passenger waiting time.

The purpose of this study is to evaluate the use of ITS technologies on real-time control of buses. Some real-time control strategies include decisions regarding whether to (1) increase or reduce the travel speed, (2) hold or release a bus at a transit center, and (3) insert a bus in the schedule. These decisions clearly depend on the planned schedule (e.g., bus headways and scheduled departure times at the bus stops) and real-time status of the transit system, such as information on current lateness of the buses, number of passengers on board buses and waiting at the various stops, and forecast arrival times of buses. This simulation model allows an investigator to determine if it is possible to relay this information in real-time to the controller with ITS.

This report documents the approach used in the development of the simulation model for analyzing ITS effectiveness on overall transit operations. The simulator developed for this study is an expansion of the simulation model developed by Hall et al(1997) to analyze the effectiveness of ITS on scheduling buses at timed transfer terminals. This new simulator considers connectivity of buses at traditional transfer stops as well as at timed transfer terminals. The model has the capability to simulate wide-area transit networks. With this feature, we can study the impact of holding a bus at a particular stop on the waiting time for passengers boarding on subsequent stops. Another major addition to the simulator is the inclusion of passenger entities in the model. In this manner, real-time control strategies taking into account passenger count and waiting times may be considered. Furthermore, including passenger entities permits the explicit modeling of the boarding and debarking process. Hence, the phenomenon of an initial delay in service causing deteriorating service farther down the line due to the increased accumulation of boarding passengers can be studied.

The model is developed using a general-purpose simulation language, AweSim (Pritsker, **1997).** The advantage of using a process-oriented language to model bus operations is that a small network model, which has the flexibility to test many different control strategies, can be used to represent detailed bus movement. The simulation model is generic and independent of any dedicated transit network. The model has a high flexibility and can be used to simulate different kinds of transit networks with varying number of bus lines and different travel patterns. The user has the flexibility to input the appropriate control strategy at each stop. With this approach an identical replica of an actual system can be simulated.

The remainder of this report is divided into three sections. We first present the methodologies applied in the model, which include description of the model, the real-time control strategies that are coded in the simulation model, and forecasting methods. The second section presents how to execute the simulation. Third, we present an example problem to illustrate the simulation process. The Appendices include all the documents referred by the report.

2. METHODOLOGY

2.1 System Description

The assumptions applied in the simulation model are listed below:

- For each bus line, headway is a constant
- For each bus line, variance of actual travel time in the segment is constant
- The initial schedule of each bus line (arrival, departure and travel time) is used for the total simulation length
- At each stop, the holding strategy is the same for all the connecting bus lines
- Passenger arrival rate is constant during the total simulation length

2.2 Network Description

This section documents the simulation model for wide-area transit networks. The AweSim software package is used to simulate the transit operations. AweSim is a general purpose simulation language distributed by Pritsker Corporation. AweSim was chosen as it has the facility for incorporating user written inserts. The simulation model developed is generic and independent of any dedicated transit network. The model has a high flexibility and can be used to simulate different kinds of transit networks with varying number of bus lines and different travel patterns. The user has the flexibility to input the appropriate control strategy at each stop. With this approach an identical replica of an actual system can be simulated.

2.3 Control Logic

Each bus in the transit network is controlled by some holding strategy when it arrives at a stop or checkpoint. The holding strategy depends on the nature of the stop (e.g., transit centers such as timed transfer stations or uncoordinated stops). To allow for different types of stops in the network, a different holding strategy is defined for each stop in the transit network.

Some of the strategies are only applicable to timed transfer stops. Transit systems have timed transfer terminals where a certain number of buses arrive within the same time window, to facilitate the transfer of passengers from one bus line to another. Control strategies evaluated in the absence of bus tracking are (1) no hold (2) all hold and **(3)** hold for a fixed period of time. However, all three strategies suffer fiom the lack of accurate up to date information on different bus lines. The static and prefixed schedules do not allow for dynamic decision making by the dispatchers. **As** a result, the efficiency of the system may be lower using control strategies that do not make use of ITS. The advent of ITS technology for tracking the buses has changed the scenario. In light of the up to date information, more dynamic decision making is feasible.

We next describe the control strategies that have been implemented as part of the simulation model. Each bus stop in the transit network has its own control strategy.

1. Do not hold any bus.
2. Hold a bus, if a connecting bus is scheduled to arrive within a time bucket.
3. Hold a bus for a fixed period of time, if a connecting bus is scheduled to arrive within a time bucket.
4. Forecast the arrival times for different bus lines approaching the stop. Hold if the following conditions are true:
   - Holding bus has a high headway
   - Forecast arrival time of an approaching bus is within a fixed holding time.
5. Forecast the arrival of different bus lines approaching the stop. Hold a bus if all the following conditions are true for any approaching bus:
   - Holding bus has a high headway
   - Forecast arrival time of an approaching bus is within a fixed holding time.
   - More than a fixed number of passengers on board the arriving bus who want to transfer to the waiting bus.
6. For the present bus stop where the holding decision is made, forecast the arrival of different bus lines approaching the stop. Hold a bus if all the following conditions are true for any approaching bus:
   - Holding bus has a high headway
   - Forecast arrival time of an approaching bus is within a fixed holding time.
   - More than a fixed number of passengers on board the arriving bus who want to transfer to the waiting bus.
   - The total waiting time of passengers on board the holding bus due to the extra hold, is less than the cumulative waiting time that the transferring passengers would face as a result of missed connection and having to wait for the subsequent run.
7. For the present stop and all the subsequent stops along the bus line, forecast the arrival of different bus lines approaching the stop. Hold a bus if all the following conditions are true for any approaching bus:
   - Holding bus has a high headway
   - Forecast arrival time of an approaching bus is within a fixed holding time.
   - More than a fixed number of passengers on board the arriving bus who want to transfer to the waiting bus.
   - The total waiting time of passengers on board the holding bus due to the extra hold, is less than the cumulative waiting time that the transferring passengers would face as a result of missed connection and having to wait for the subsequent run.

Strategies 1, 2 and 3 do not need the application of ITS technologies. **A No Hold** strategy favors the passengers on holding buses and an **All Hold** strategy favors passengers on connecting buses. In strategy 3, a predefined holding time is used to balance these two benefits. Strategies 4, 5, 6 and 7 require ITS technologies as well as some forecasting methods if implemented in actual practice. Comparing to 3, strategy 4 makes use of the forecasted arrival time of connecting buses to determine the dispatching time rather than to dispatch the bus upon the arrival of all connecting buses or at the end of the holding time window. It does not guarantee a less global passenger trip time. Strategy *5* requires not only forecasted bus arrival times, but also the number of transferring passengers on the connecting buses. In the last two strategies, dispatching time is locally optimized **as** to ensure that waiting time of passengers on the holding bus is less than the cumulative waiting time of the transferring passengers. The difference between strategies **6** and 7 is that the latter one considers the passenger waiting times at the current bus stop as well at all the subsequent stops since holding a bus a particular stop has the potential to increase the waiting times of passengers at subsequent stops. Although the last two strategies may be the most difficult to implement in practice due to the additional computation that is required, these strategies take into account the system wide passenger waiting time in determining the bus dispatching time. In addition, they do not require any presetting of the input parameters (e.g., maximum holding time) like strategies **3-5.** Strategies **6** and 7 determine the bus holding time based on the total waiting time criteria of those passengers currently at the stops and those forecasted to arrive before the bus is forecasted to arrive. Since the actual bus arrival times and number of passengers on board are random variables, the dispatching times determined by either strategy **6** or 7 are not guarantied to be optimal or even dominate the other strategies in minimizing the total passenger waiting time criteria.

To allow for flexibility in modeling different types of stops, a different control strategy can be used in each stop along the bus route. The selected control strategy depends on numerous factors including the headway of the buses and the nature of the transition matrix from one bus line to another. The model also allows the user to specify, if early departure is possible from a stop or not. **A** standard bus stop with no connections can be simply modeled **as** a **No Hold** (Strategy 1) with possibility for early departures.

2.4 Forecasting Method

Some of the holding strategies require a forecast on bus arrival times and the number of passengers on the bus. We next describe the forecasting methods used in the model when a stop uses a holding strategy that requires ITS. The mathematical expressions of the forecasting methods are presented in the ITS evaluation report (Hall et al, 1999).

Arrival Time Forecast

In the presence of a bus tracking system, arrival times can be forecasted to any stop. We assume that the forecast is updated each time the bus passes a scheduled stop. The bus sends its forecast to the subsequent stops once on arrival and again at the departure. Since the arrival and the departure times of a bus at a stop may not coincide, the revised forecast at the departure is expected to be different from the ones sent on arrival.

We assume a lognormal distribution for travel times. Forecast is made whenever the bus departs from a bus stop. For holding strategy 7, it is necessary to forecast the arrival/departure times for all the subsequent stops.

There are two forecasting mechanisms for two different scenarios, either a bus is allowed to depart early or not. When early departure is allowed, the forecasted arrival time is simply the actual departure time from the previous stop plus the expected travel time. Assuming negligible dwell time, then the forecasted departure time is equal to the forecasted arrival time.

When early departure is not allowed, even though a bus arrives at a stop ahead of schedule, it **has** to wait until the scheduled departure time to be dispatched. The actual departure time is the maximum of the actual arrival time and the scheduled departure time. Therefore, the forecasted departure time, which equals the expected actual departure time, does not equal to the forecasted arrival time. Actually, it should be greater by intuition. The actual arrival time at the stop adjacent to the one where the forecast is made still has a lognormal distribution. However, the departure time has a density function that is composed with a truncated lognormal with a peak at the scheduled departure time. For stops subsequent to the adjacent one, the actual arrival time has a more complicated distribution. We use an approximation approach to forecast the arrival and departures times at all the subsequent stops.

Passenger Forecast

We also forecast the number of passengers on board the bus when it arrives at each subsequent stop. The forecast is sent on arrival to a stop and again at the time of departure. The expected number of passengers on board depends on the number of connecting buses that the given bus might intersect on its way to a stop. Hence, the forecast on the number of passengers is dependent on the control logic at subsequent stops. For example in the case of a no hold strategy, it is likely that there are no transferring passengers, while in the case of an all hold strategy, there would certainly be transfers thereby increasing the number of passengers on board. A very conservative approach is to forecast the present number on the bus plus the expected number of originating passengers at subsequent stops. However, this approach does not result in accurate predictions, since it does not take into account the passengers who might get off the bus or the extra transferring passengers who might board the bus. Using the control logic at subsequent stops, a more sophisticated approach can be used to predict the expected number of passengers. A detailed description of the passenger forecasting approach is provided in the evaluation report (Hall et al, **1999).**

3. SIMULATION EXECUTION

3.1 Overview of Simulation Steps

Step 1: Initialization

In this discussion, we assume the user has already installed the simulation software package AweSim, which is a general purpose simulation language distributed by Pritsker Corporation.

a. Before starting the simulation, the user can modify the input data file (section **3.2)** to accommodate the experiment parameters of interest, such **as** headway, number of lines, number **of** stops, transit network layout, departure times, transaction matrix, etc. It is also important to input the index of the holding strategy being tested and the switches for statistics collection.
b. The user may use notepad or UserEditor in AweSim to reset the output files (output.dat, line_out.dat, stop_out.dat), clear all the prior experiment results, and save them.
c. The user can set the finish time of the simulation by changing the **INITIALZE** statement in AweSim. The simulation is currently set to run for 10000 minutes.

Step 2: Startina Simulation

In the AweSim menu, select Simulate-Run to start simulation. Wait for the simulator to finish the simulation to complete. After simulation, the output file "output.dat" is updated.

Step 3: Output Processing

After each simulation run, execute "C:\PROJECT\ITS\ouput_processing.exe" (by double clicking the file name in the windows file manager) to generate line and stop statistics. The result of this execution is to update the output files "line_out.dat" and "stop_out.dat". It is not necessary to clear output files between two runs in a sensitivity study. The simulator will append new results to the output files instead of replacing it. After all the simulation for a specific scenario is over, the user is encouraged to load the three output files into Excel or another data analysis software package to process the data.

3.2 Input Data

The network transit system that is simulated is defined and controlled by many user input parameters. For example, the routes are specified by inputting the scheduled departure times of the first run of each bus line at each stop. The scheduled departure times of subsequent runs of the bus are determined by the inputted headway. The model reads all input parameters from a data file. The following table summarizes the inputs to the simulation model. In the table, the Index I represents the bus line number and the Index J represents the jth stop in the route. In addition, the user is provided the flexibility to collect statistics for some specific stops or bus lines of interest. Matrices that contains the switches for these statistics are included at the end of the data file.

INPUTS	Description
NLINE	The number of bus lines in the transit network
T_NSTOP	Total number of bus stops on all lines
MAX–NSTOP	Maximum stop identification (ID) for all lines
NSTOP[I]	Number of stops for bus line I
AC_STOP[I][J]	The sequence of stop ID visited by each bus line.
HEADWAY[I]	Headway for bus line I (assumed constant)
ARRIV_T [I][J]	Scheduled arrival time for bus line I to stop J (J is the stop ID) (array stores values only for the first run of each bus line)
DEPAR_T [I][J]	Scheduled departure time for bus line I from stop J (array stores values only for the first run of each bus line)
TRIP_T [I][J]	Scheduled travel time for bus line I to stop J (array stores values only for the first run of each bus line)
TRANS[J][I][K]	Transition matrix of transferring passengers when bus line I arrives to stop J Index K represents the transferring bus line number. K varies from 1 to NLINE. Elements of the matrix vary from 0 and 1. Summation of elements over K for given J and I is less or equal to 1. Difference between the summation and 1 is the fraction of passengers being dropped off.
SLACK[I]	Maximum slack built into the schedule for bus line I. (ratio between average actual travel time to scheduled travel time)
SIGMA[I]	Standard deviation of travel time in segment for bus line I.
CAPACITY[I]	Bus capacity for bus line I
T_BOARD & T_DEBARK	Represents the average boarding and debarking time per passenger (Gamma distribution is used to generate the actual boarding and debarking times)
AWARE[I][J]	The fraction of people boarding a bus who are aware of the scheduled arrival time of bus line I at stop J

BOARDS[I][J]	The average number of boarding passengers at stop J for bus line I (Poisson Distribution is used to generate the number boarding)
I_LOAD[I]	Initial load of passengers on bus line I when it enters the transit network
STOP_D[J]	Control strategy at stop J (range from 1 to **7)**
STOP_ED[J]	Binary variable representing whether or not a bus is allowed to leave earlier than its scheduled departure time from stop J (0- not allowed to leave early, 1 – allowed leave early)
WINDOW	To coordinate movement of different bus lines whose scheduled arrival times to a stop do not differ by more than WINDOW
MAX_W	Maximum allowable holding time
	Threshold value for the number of transferring passengers to determine whether or not to hold a bus
FORE–P	Binary variable representing whether switch on technique used for forecasting the passenger count of a bus line to its subsequent stops
FORE_B	Binary variable representing whether switch on technique with integration used for forecasting bus arrival and departure time
H_HEADWAY	Bus lines with headway over H_HEADWAY represent the high headway lines
SWITCH_L[I]	Binary switches for collecting line statistics (0 =off, 1= on)
SWITCH_S[J]	Binary switches for collecting stop statistics (0 = off, 1= on)
SW_LS[I][J]	Binary switches for collecting statistics at stop J of line I (0 = off, 1= on)

These parameters of the simulator are stored in the input data file "dataset.txt" which is stored in the directory "PROJECT\ITS". Appendix B shows a sample input data file. The file is divided into several domains corresponding to the inputs mentioned above. There is a comment for each domain in a pair of asterisks. **Do** not remove the comments. A user may modify the content of each domain and then save the data file. The simulator will load the modified input parameters automatically when the simulation starts.

3.3 Output Data

<u>Performance Measurements</u>

The simulation model generates outputs related to bus lines and passengers. For each output performance measure, the average and variance are recorded. Passenger performance metrics in the simulation model include: (1) average passenger trip time, (2) average passenger delay, and **(3)** average passenger waiting time. Passenger delay and waiting times measure the inconvenience that the system imposes to passengers. The distinction between these two measures is made in the following definitions

a. Passenger Related Outputs:

- Global trip time per passenger for each bus line, which is defined to be the difference between the time passenger leaves the system, i.e. actual arrival time of the bus to the last stop on the trip for the passenger, from the scheduled boarding time of the passenger. We base the definition of the trip time on the scheduled bus arrival time instead of the actual arrival time of the passenger because the performance of the system will be unnecessarily penalized for early or late arriving passengers.
- Total waiting time among all passengers at a stop (or a set of stops of interest), where passenger waiting time is defined to be the difference between the time the bus actually departs the stop from the time a passenger arrives to the stop. For originating passengers, it is the time the bus actually departs minus the time the passenger arrives to the stop. For continuing passengers, it is the time the bus actually departs minus the time the bus actually arrives to the stop. For transferring passengers, it is the time the connecting bus actually departs minus the time that the bus that dropped the passenger arrives to the stop.
- Total delay among all passengers at a stop, where the definition of passenger delay is based on the passenger type. For passengers already on board, passenger delay is the difference between actual bus departure time and scheduled departure time, minus the bus lateness at the previous stop. The lateness is subtracted to insure the delay is not double counted for multiple stops. If this value is less than zero, then the passenger delay is zero. For passengers originating at the stop and transfer passengers, delay is defined to be the maximum of zero and the difference between the actual departure time and the scheduled departure time. For originating passengers, we use the passenger's scheduled departure time if they arrive either early or on-time. If the passenger arrives late, we use the scheduled departure time **of** the next bus to arrive. This ensures that we do not penalize the system for passengers missing their bus because of passenger late arrival. For transferring passengers, the scheduled departure time is for the bus line the passenger is coordinated to meet irrespective of the passenger's actual arrival time to the stop. In this case, the majority of the delay penalty for transferring passengers is for passengers missing their connection.

b. ***Bus Related Outputs:***

- Arrival and departure tardiness at different stops, where tardiness = max(actual - scheduled,0)
- Arrival and departure earliness at different stops, where earliness = max(scheduled - actual,0)
- Arrival and departure lateness at different stops, where lateness = actual - scheduled
- Dwell time at different stops, where dwell time is defined to be the difference between the actual arrival time from the actual departure time
- Boarding and disembarking times at different stops

- Holding time at different stops, where holding time = dwell time - boarding/disembarking time
- Waiting time at different stops, where waiting time = holding time - arrival earliness

Output Data Files

Three output files are updated after each simulation run: output.dat, line_out.dat, stop_out.dat. The first file records the overall performance measurements of the bus transit system. The second and third ones record the performance measurements of specific lines and stops respectively. See Appendix C for a sample output file.

4. EXAMPLE PROBLEM

4.1 Problem Description

In this example, we simulate a small bus transit network using the model presented earlier. There are 5 bus lines and 12 stops in this bus transit system. As a simplified illustration, the bus network has a nicely ordered structure (i.e., each bus line has all 12 stops along its route in the same sequence). All the bus lines have the same headway of 60 minutes. It is assumed that the scheduled travel time in each segment equals to 2.5 minutes. The standard deviation of actual travel time in the segment is 1.5 minutes. The ratio of the actual of travel time over the scheduled travel time is set to 1 (i.e., there is no slack in the schedule). Stop *6* is the transfer stop where we test different control strategies. We assume that the passengers only transfer at this stop. Half of the passengers will stay on the bus at the transfer stop. The other half have equal probability of transferring to any **of** the four connecting lines.

4.2 Input Data

The problem described above is converted to an input data file **as** shown in Appendix B. The definition of each field is quoted by a pair of asterisks. Because the stops are ordered sequentially, the maximum stop identification (ID) equals the total number of stops. To test different control strategies at stop *6*, the sixth element in the corresponding field is changed from 1 to **7.** There is one passenger transfer matrix for each stop. A row corresponds to the original bus line and a column corresponds to the connecting line. The element in the transfer matrix represents the fraction of passengers on the original bus. Switches for collecting statistics on all the lines and stops are set on.

4.3 Output Data

After running the simulation for a length of 10,000 minutes, the "output.dat" file has been updated **as** shown in Appendix C. As described before, by executing output_processing .exe, statistics for all the lines and stops are collected. "line_out.dat" and "stop_out.dat" files are also included in Appendix C.

5. REFERENCES

Hall, R. W., M. Dessouky, A. Nowroozi, and A. Singh (1997). "Evaluation of ITS Technology for **Bus** Timed Transfers," PATH Report UCB-ITS-PRR-97-37, Richmond, CA.

Hall, R. W., M. Dessouky, L. Zhang, and A. Singh (1999). "Real-Time Control of Bus Transit Systems Using ITS Technologies," PATH Report, Richmond, **CA.**

Pritsker, A. A. B, J. J. O'Reilly, and D. K. LaVal (1997). ***Simulation with Visual SLAM and AweSim,*** Wiley, **New** York, NY.

APPENDIX A

Internal Elements of the Simulation Network

Entities in the Simulation Network

There are two types of entities in the simulation model:

- Entities representing the buses in the system.
- Entities representing the passengers.

Entities representing different bus lines are entered into the network at time intervals equaling the headway for their specific bus line. Buses move on their specified sequence of stops until they reach the last stop in their route, at which instant they are terminated. The actual travel time on a bus segment (i.e., path between two adjacent stops) is generated fiom a lognormal distribution. The originating passengers for the different bus lines at different stops are **known** to be of two categories. The arrival times of aware passengers are generated from a lognormal distribution with mean being slightly before the scheduled arrival time of the bus. Unaware passengers arrive randomly and enter the network at a stop any time between two subsequent runs of a bus line. The number of originating passengers at a stop are generated from a Poisson distribution. The boarding and debarking times from/to the bus are generated using the gamma distribution. Once a passenger boards a bus, the entity representing the passenger is terminated and the bus attribute representing the number of passengers on board is incremented. Each entity in the simulation model has its unique set of attributes, which are collected in the LTRIB array and ATRIB array. The LTRIB array is used to store the integer values while the ATRIB array is used to store real value attributes.

Bus attributes are:

LTRIB #	Description
1	Line number of the bus
2	Index representing the number of stops covered in its sequence of M stops
3	Last stop number visited by the bus
4	Number of passengers on board
5	Run number of the specific line

ATRIB #	Description
1	Arrival time at the last stop visited

Passenger attributes are:

ATRIB #	Description	
	Originating	*Transferring*
1	Arrival time at the stop	Max of (Scheduled arrival time of the bus, Actual Arrival time of the

The simulation logic is conducted by user written event nodes. The arrival of originating passengers at a stop and that of the bus are simulated by user-written events. To capture the actual arrival time of the passenger, the creation of entities at a stop is begun one headway ahead of the scheduled arrival time of the bus for each stop in the network. The following sequence of events describe the logic used to simulate the movement of buses.

EVENT #	Description
1	Used to generate the buses for different bus lines. The event initializes the attribute values of the bus entity and reschedules itself to generate the subsequent run of different bus lines. It sends the generated buses to Event 2, after a time duration representing the actual travel time to reach their first stop.
2	Represents the arrival of a bus to a given stop. On arrival to a stop the bus sends it arrival forecast to the subsequent stops. It calculates the number of transferring passengers and the time it would take to disembark the passengers. Passengers wanting to transfer to another bus are generated and placed in appropriate queues. The attribute representing the number of passengers on board is decremented. The bus is scheduled for Event **3** after a time duration representing the time to disembark the passengers.
3	Having disembarked all the passengers, the bus goes into the control logic at the arrived stop. After the bus comes out of the control logic, it is scheduled for Event **4**, after a time lag of 0 or max(0, TNOW- Scheduled Departure) depending on whether the bus can leave early or not.
4	Having come out of the control logic, the bus loads the passengers that wanted to get on to the bus. Since the loading of all the passengers who arrive while the bus is disembarking its passengers and while it is waiting at the control logic, takes place during the same duration no extra wait is enforced for the

	above set of passengers. However, passengers who arrive at the instant the bus got out of the control logic or those who could not board the bus due to the long passenger boarding queue, force the bus to wait for an extra duration for boarding time. The bus is scheduled for Event **5** after a time duration representing the extra time that unboarded passengers might take to board the bus.
	The bus is finally ready to leave. It sends its latest forecasts to the subsequent stops and calculates the departure statistics. A check is made to see if the bus **has** reached its final stop or not. If yes, it is scheduled for Event ***6*** else it updates its attributes to represent the subsequent stop. Actual travel time for the leg is calculated and the bus is scheduled back to Event 2 after a time lag equaling the actual travel time
6	The arriving bus is terminated, representing the completion of a run for a bus line in the transit network.

The following Event nodes are used to generate the arrival of originating passengers to a stop.

EVENT #	**Description**
7	Dummy entities representing the originating passengers arrive one headway ahead of the scheduled arrival of buses **,** at its various stops. It generates the number of passengers that would board the bus at the given stop. Having generated the number **of** passengers, it splits the arriving passengers into aware and unaware category. It then assigns all the originating passengers an arrival time and schedules their arrivals to Event **8** at times representing their actual arrival times. The dummy entity reschedules itself to Event **7** after a time lag equalling the headway of the bus line for which it is generating the originating passengers.
8	The entities representing the originating passengers arrive to a stop. On their arrival, they go and wait in appropriate queues for their respective buses.

Flow charts showing the logic for bus movement and the different dispatching rules are shown in Appendix D.

APPENDIX B

Sample Input Data File

```
*number of bus lines* (NLINE)
5

*total number of bus stops* (T_NSTOP)
12

*maximum ID of a stop for all lines* (MAX_NSTOP)
12

*number of stops for each bus line* (NSTOP[I])
12 12 12 12 12

*bus stop ID for each line* (AC_STOP[I][J])
1 2 3 4 5 6 7 8 9 1 0 1 1 1 2
1 2 3 4 5 6 7 8 9 1 0 1 1 1 2
1 2 3 4 5 6 7 8 9 1 0 1 1 1 2
1 2 3 4 5 6 7 8 9 1 0 1 1 1 2
1 2 3 4 5 6 7 8 9 1 0 1 1 1 2

*headway for each line* (HEADWAY[I])

60 60 60 60 60


*initial scheduled arrival time at each stop* (ARRIV_T[I] [J])
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5

*initial scheduled departure time at each stop* (DEPAR_T[I][J])
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5
200 202.5 205 207.5 210 212.5 215 217.5 220 222.5 225 227.5


*scheduled travel time to each stop* (TRIP_T[I][J])
0 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5
0 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5
0 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5
0 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5
0 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5

*passenger transaction matrix at each stop for various lines*
(TRANS[J][I][K])
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
```

0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

0.5 0.125 0.125 0.125 0.125
0.125 0.5 0.125 0.125 0.125
0.125 0.125 0.5 0.125 0.125
0.125 0.125 0.125 0.5 0.125
0.125 0.125 0.125 0.125 0.5

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

ratio between actual travel time to scheduled travel time for each line (SLACK[I])
1 1 1 1 1

sigma for each line (SIGMA[I])
1.5 1.5 1.5 1.5 1.5

bus capacity for each line (CAPACITY[I])
10000 10000 10000 10000 10000

mean boarding time (T_BOARD)
0.02

mean debarking time (T-DEBARK)
0.02

fraction of aware passenger at each stop each line (AWARE[I][J])
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1

mean number of passengers that board the bus at each stop (BOARD[I][J])
2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2

initial load (I-LOAD[I])
0 0 0 0 0

holding strategy for each stop (STOP_D[J])
0 0 0 0 0 0 0 0 0 0 0 0

departing before schedule or not (STOP_ED[J])
0 0 0 0 0 0 0 0 0 0 0 0

constant window size under no forecast (WINDOW)
3

maximum waiting time under forecast (MAX-W)

1

minimum number of transferring passenger for a bus to be held (MAX_P)
1

forecasting switches (FORE_P & FORE_B)
1
0

switches for statistics collecting
1 1 1 1 1 (SWITCH_L[I])
1 1 1 1 1 1 1 1 1 1 1 1 (SWITCH_S[J])

1 1 1 1 1 1 1 1 1 1 1 1 (SW_LS[I][J])
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1

APPENDIX C

Sample Output Data Files

Output.dat

STATISTICS

CASE 1 :: NO HOLD
CASE 2 :: ALL HOLD
CASE 3 :: HOLD For A Fixed Period of Time
CASE 4 :: FORECAST AND HOLD IF ARRIVAL TIME WITHIN THE TRESHHOLD
CASE 5 :: FORECAST AND HOLD IF ARRIVAL TIME AND NO. OF TRANSFERS WITHIN THE TRESHHOLD
CASE 6 :: OPTIMIZE : ONLY FOR THE THE PRESENT STOP
CASE 7 :: OPTIMIZE : PRESENT AND AHEAD
No. of lines 5 No. of Stops 12

CASE 7

Passenger Related

Avg TRIP–T Pass 18.79
Avg Wait 1.07
Avg Delay 1.15
Avg Debark_T 0.00
Avg Board–T 0.00

Bus Related

Avg Hold–T 0.40
Avg Wait–T 0.20
Avg Dwell_T 0.40
Avg Dep_Tar 2.87
Avg Dep_Lat 2.87
Avg Dep_Ear 0.00
Avg Arr_Tar 2.67
Avg Am–Lat 2.47
Avg Am–Ear 0.20

Line_out.dat

Line 1 2 3 4 5

Pass_Wait 1.085 1.068 1.085 1.035 1.054
Pass_Delay 1.19 1.163 1.136 1.1 1.152
Pass–Debark 0 0 0 0 0
Pass–Board 0.001 0.001 0.001 0.001 0.001
bus–hold 0.4 0.399 0.414 0.417 0.382
bus–wait 0.194 0.205 0.215 0.21 0.192

bus–dwell 0.4 0.4 0.414 0.417 0.382
bus–dt 2.909 2.932 2.801 2.763 2.943
bus–dl 2.909 2.932 2.801 2.763 2.943
bus–de 0 0 0 0 0
bus–at 2.715 2.727 2.586 2.552 2.751
bus–a1 2.508 2.533 2.387 2.346 2.56
bus–ae 0.206 0.194 0.199 0.207 0.19

stop_out.dat

STRATEGY 7

S t o p 1 2 3 4 5 6 7 8 9 1 0 1 1 12

Pass–Wait 1.797 1.492 1.175 1.004 0.887 4.093 0.906 0.822 0.752 0.712 0.677 0.631
Pass–Delay 0 0.596 0.646 0.661 0.748 4.222 1.059 1.033 0.937 0.959 0.934 0.895
Pass–Debark 0 0 0 0 0 0.001 0 0 0 0 0 0
Pass–Board 0.001 0.001 0.001 0.001 0.001 0.002 0.001 0.001 0.001 0.001 0.001 0.001
bus–hold 0 0.531 0.418 0.371 0.295 2.729 0.037 0.063 0.065 0.094 0.109 0.104
bus_wait 0 0 0 0 0 2.43 0 0 0 0 0 0
bus–dwell 0 0.531 0.419 0.372 0.295 2.73 0.037 0.063 0.066 0.095 0.109 0.104
bus–dtO0.5560.881 1.159 1.4884.164.1974.2574.2524.3634.5264.645
bus–dl 0 0.556 0.881 1.159 1.488 4.16 4.197 4.257 4.252 4.363 4.526 4.645
bus_de 0 0 0 0 0 0 0 0 0 0 0 0
bus–at 0 0.556 0.881 1.158 1.488 1.729 4.197 4.257 4.251 4.362 4.526 4.645
bus–a1 0 0.025 0.462 0.787 1.193 1.429 4.16 4.194 4.186 4.268 4.417 4.541
bus–ae 0 0.531 0.418 0.371 0.295 0.299 0.037 0.063 0.065 0.094 0.109 0.104

APPENDIX D

Flow Charts of Model Logic

Figure 1. Flow Diagram for Movement of Buses Along Its Route

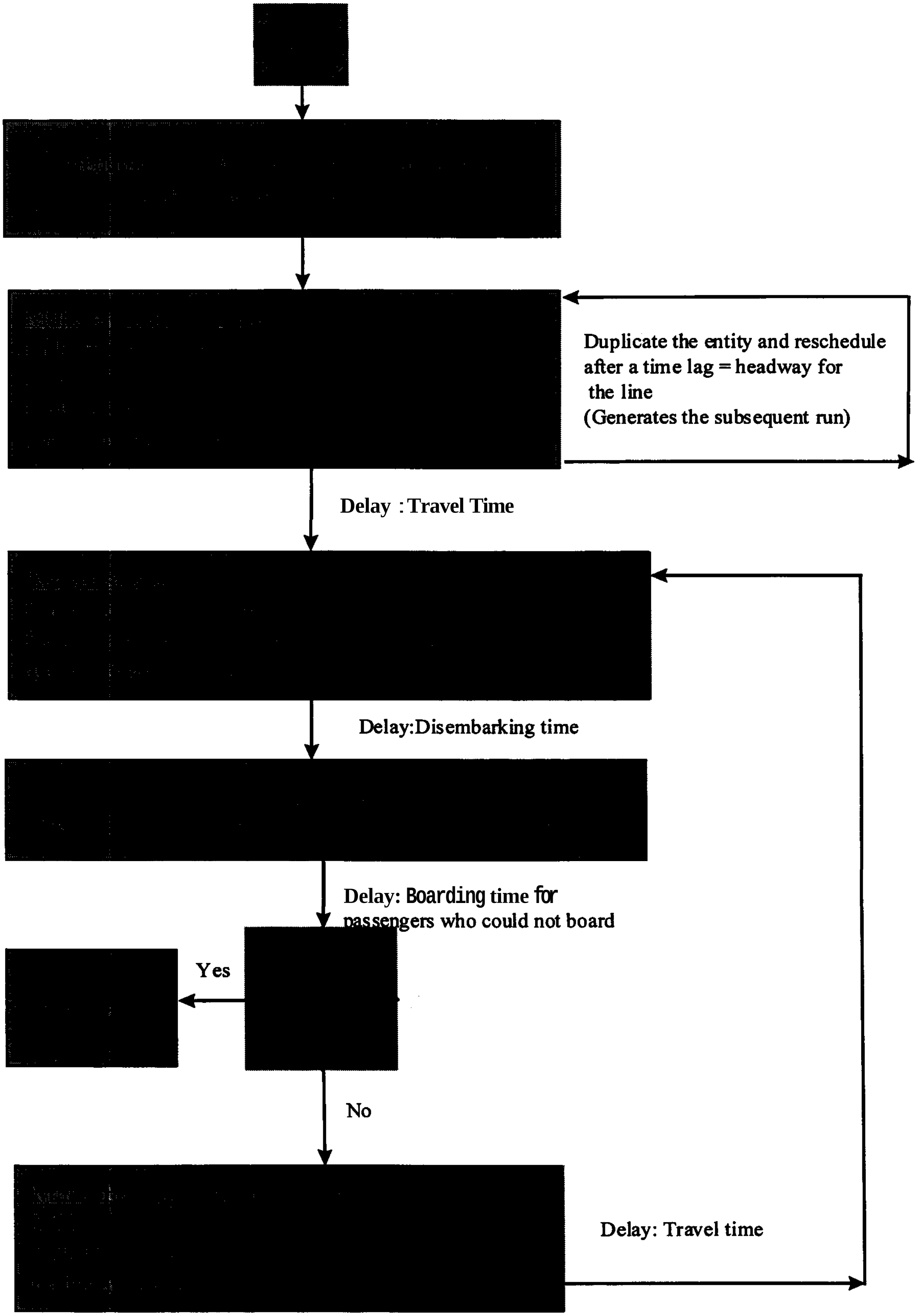


Figure 2. Control Logic: No Hold

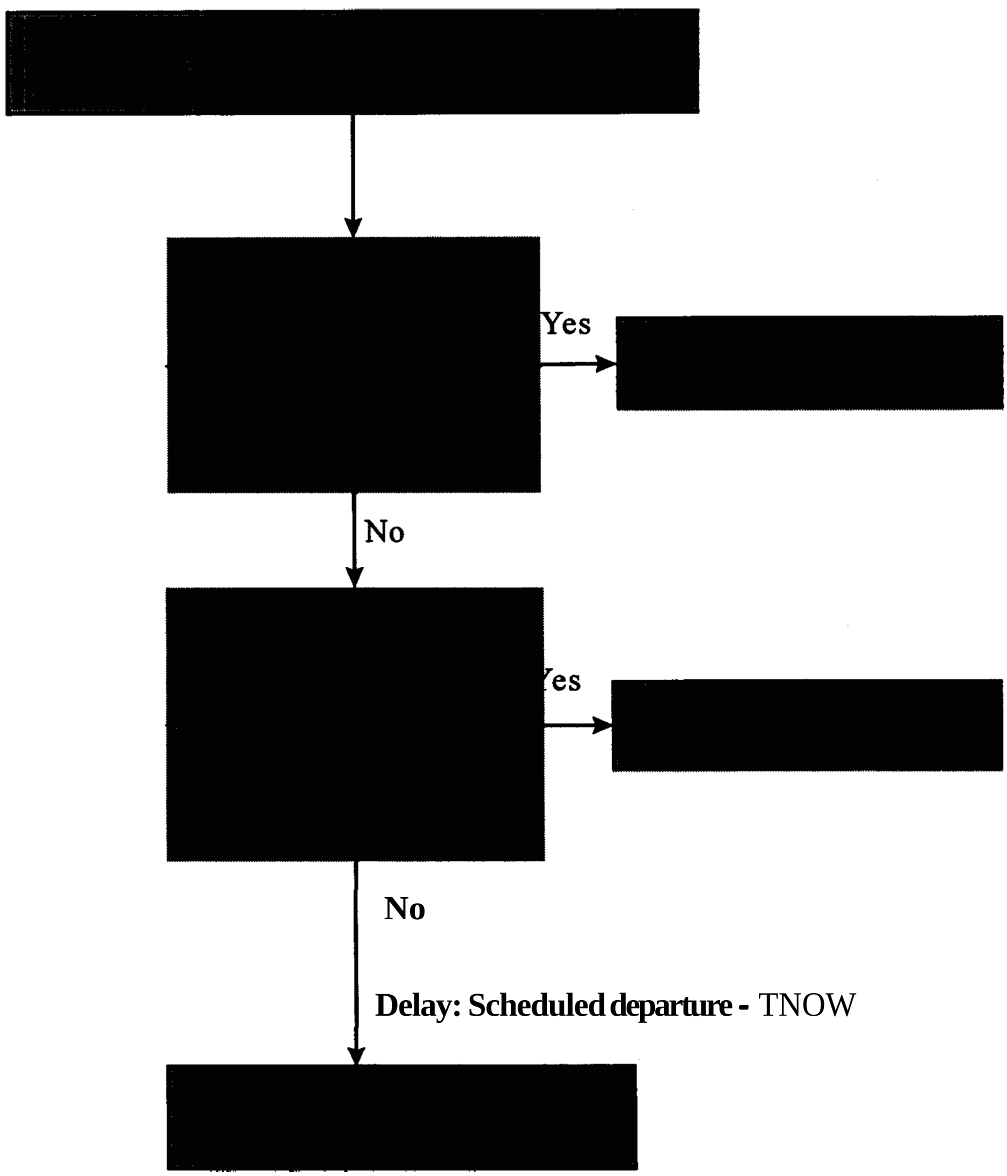


Figure 3. Control Lopic: All Hold

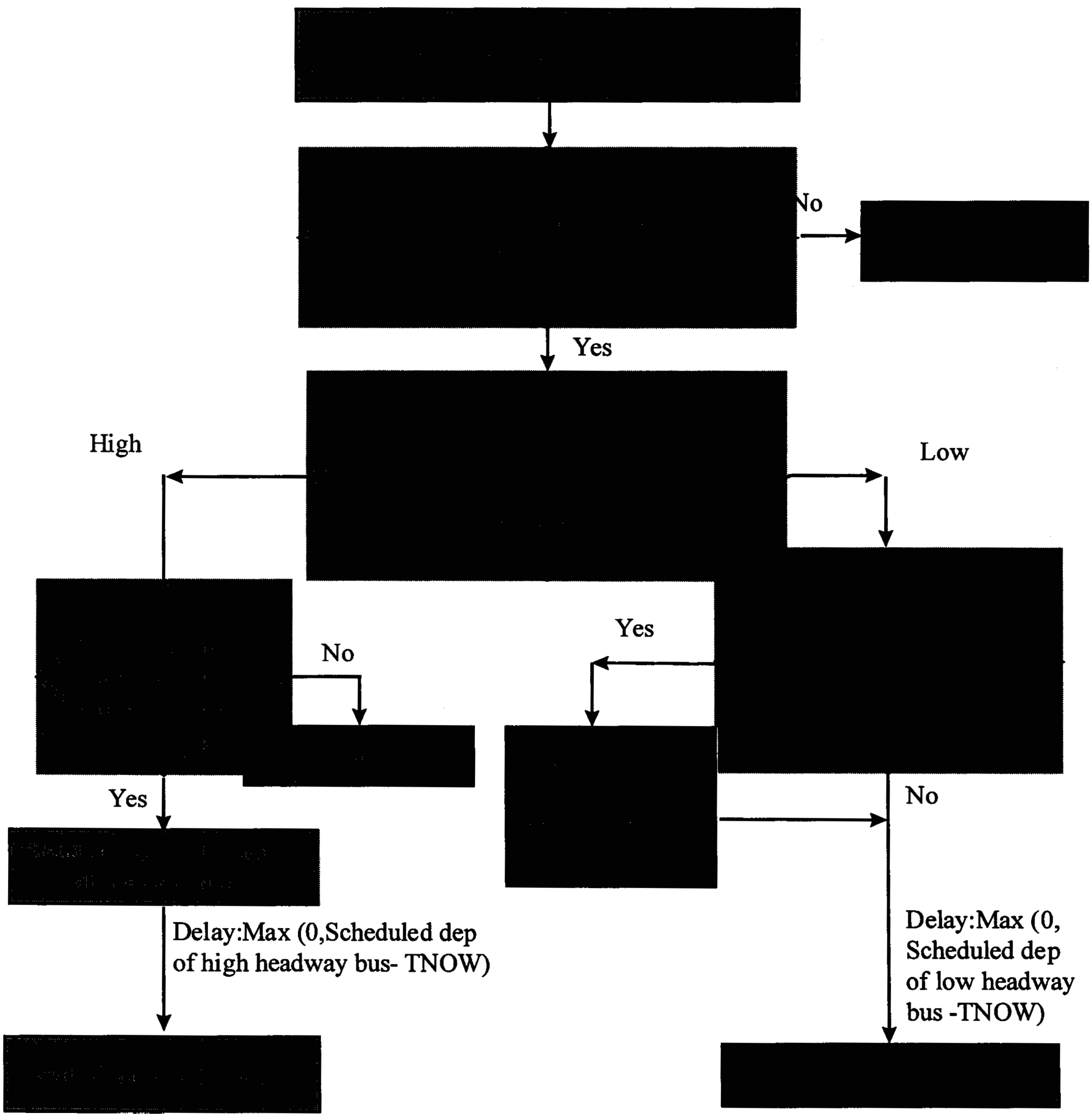
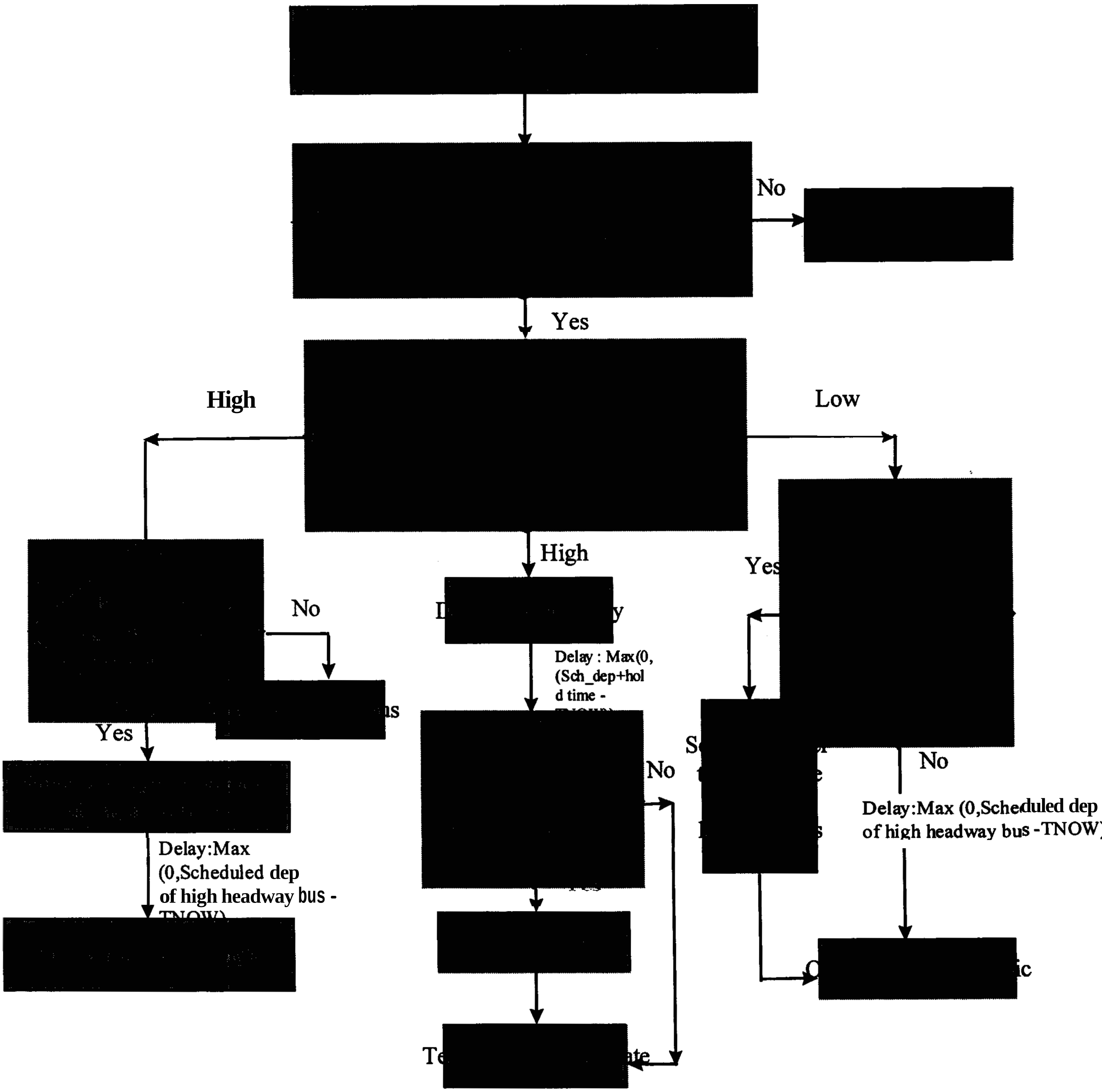


Figure 4. Control Logic: All Hold With a Maximum Waiting Period



<u>Fipure 5. Control Logic: Forecast and Hold if the Forecasted Arrival is Within the Holdinp Time</u>

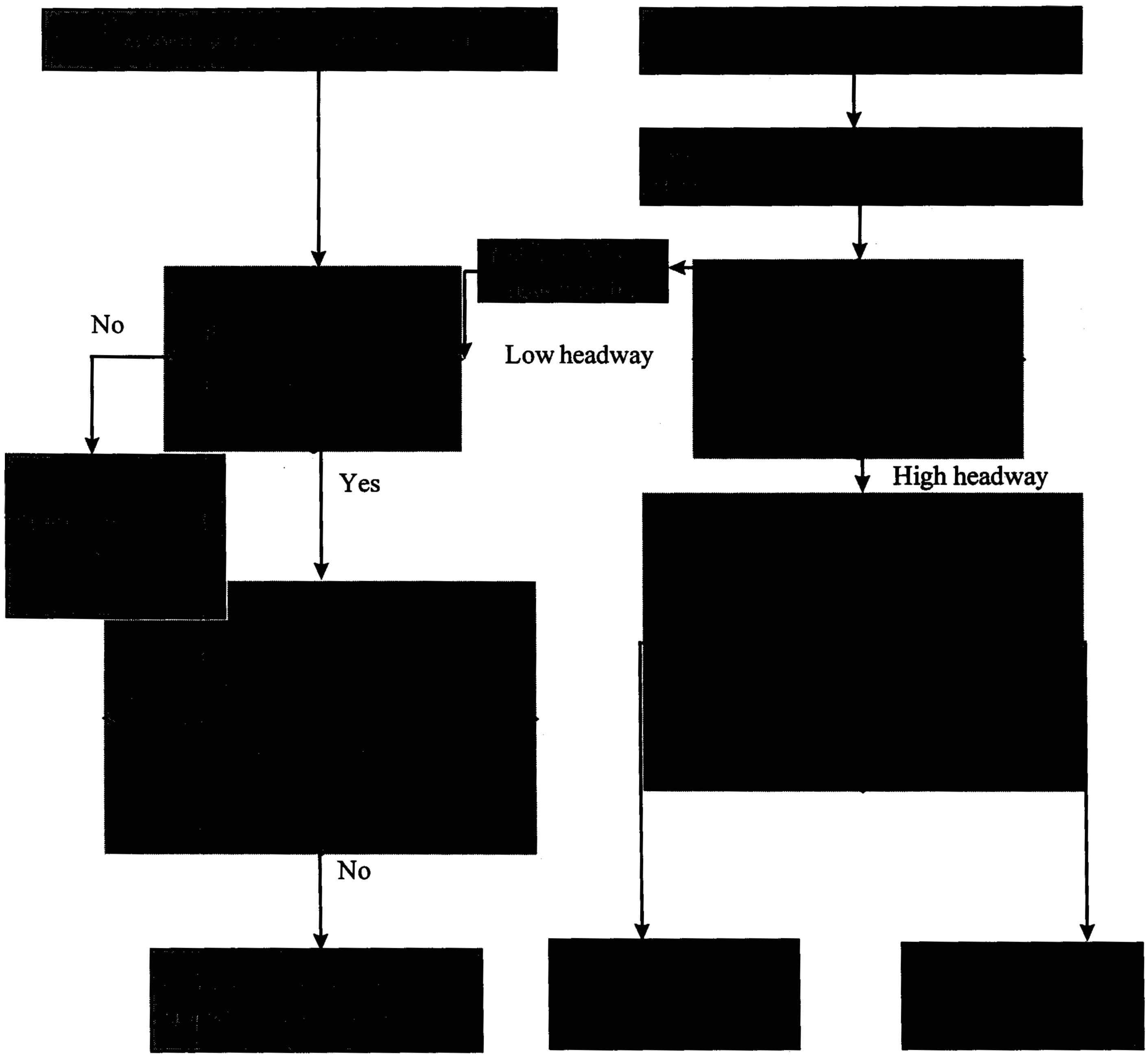


Figure 6. Control Logic: Forecast and Hold if the Forecasted Arrival Is Within the Holding Time and the Transferring Passengers Are Greater Than a Threshold Value