Real Time Dispatching of Bus To Improve Transfers

by

Qing Li

A thesis
Presented to the University of Ottawa
In fulfillment of the
thesis requirement for the degree
of Master of Science (Systems Science)

Ottawa, Ontario, 2001

I hereby declare that I am the sole author of this thesis.

I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Qing Li

I further authorize the University of Ottawa to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Qing Li

Abstract

The purpose of this research is to provide insight into the effectiveness of certain real time dispatching control actions on service reliability for transfer from a high frequency bus route to a low frequency bus route. A model of the operation of 2 connected bus routes was built based on data from real bus routes operated by OC Transpo in Ottawa, and a simulation program was designed and developed. This simulation program was used to conduct experiments to compare different control actions to improve transfer effectiveness, to test the impacts of the bus threshold on passenger waiting time and to evaluate the AVL detector location. We defined an objective function – cost/benefit function, to measure the system performance of the above cases. This objective function considered 2 important factors: number of passengers relevant to transfer and waiting time relevant to control.

The research and simulation experiments suggested that real time monitoring technologies can improve the transit bus transferring significantly without preplanned timed transfer. The detector location is determined by feeder line headway primarily and the connecting bus holding threshold secondarily. Without considering the hard requirements from outside system, the bus holding threshold should not be greater than the feeder line headway.

Acknowledgements

I wish to express my most sincere gratitude to Dr. Francois M. Julien, my thesis supervisor, for his advice, valuable guidance and support, encouragement in this research work and great effort in helping me to prepare this thesis.

I would like to give special thanks to Professor Jeffery Sidney for his appreciated suggestions, and valuable supports during the research work.

I really thank for valuable comments by the thesis examiners, Professor Wojtek Michalowski and Professor Jeffery Sidney.

Dedication

To my wife Tong, my parents Mr. & Mrs. Li, and my parents in law Mr. And Mrs. Guo.

Table of Content

Chapter 1, Introduction

The purpose of this chapter is to provide the background and motivation to our research.

In public transit systems, a single bus route does not always exist connecting every possible point of origin to every possible point of destination on the network. Passengers will normally have to transfer from one bus route to one or more bus routes during their trip. A significant proportion of passengers may transfer every day and missing a transfer may represent a very long wait for the next bus, or having to use alternate mode of transportation, such as a taxi which is more costly and may be inconvenient.

According to a survey in the United States [1], the number of transferring passengers is about 21% of the total number of public bus users. In urban areas, this percentage is about 23%.

Table 1.1 Nationwide Distribution of Linked Transit Trips by Transfers.

Transfers	Distribution (%)
Zero	79
One	18
Two or More	3

Table 1.2 Distribution of Linked Transit Trips by Number of Transfers and Area Density.

Area Density	Percent by Number of Transfers		
	0	1	2+
Rural	96	4	0
Small Town	94	4	0
Suburb	82	13	5
Second City	80	15	5
Urban	77	20	3

Over the years, the time and aggravation involved in transfers have been believed to be one of the primary problem areas for people using public transit over and above issues such as crime, transit time, accessibility and cost [1]. By providing a fast transferring to reduce the passengers waiting time at transfer stop may also help to reduce crime and reduce the total time spent in the public transit system. Therefore, achieving successful transfers is one of many concerns in designing reliable and effective service for the general public.

On the other hand, the wide range of information technologies now being incorporated into transportation systems has steadily increased the value of real-time monitoring and inspection of vehicles and infrastructure. Today, low-cost devices are available that make an extremely wide range of physical measurements and then are coupled to computer chips capable of generating warnings or adjusting operations. A virtually unlimited number of physical mechanisms and sensing concepts are potentially available, but devices of special importance to transportation will be identified and brought to fruition only to the degree that the transportation community makes these needs known and establishes their potential economic and operational value [2].

In Ottawa the urban public transit agency OCtranspo (Ottawa-Carleton Regional Transit Commission) is managing 130 routes daily, covering over 5000 bus stops. Route 95 is the backbone of the transit network. Route 95 crosses the Ottawa region from East to West and back, going through the downtown area. It operates on the East-West segment of OCTranspo's Rapid Transit way, which is a roadway dedicated to the exclusive use of OCTranspo buses. It has a short headway which is about 4 minutes, i.e. about 15 buses per hour. Route 120 is a local bus route, which connects with route 95 at the Place d'Orleans bus stop. Route 120 has the headway of about 30 minutes. So generally if a transfer passenger from route 95 just misses a route 120 bus, he or she will have to wait for about 30 minutes, or to choose alternative transportation.

With the installation and operation of monitoring systems such as AVL (Automated Vehicle Location System) in transit agencies, the important challenges of optimizing service reliability and minimizing the cost of operations of the systems arise. An AVL system has been under development since the late 1980's in OCTranspo and switched into use earlier in 1995. Some research work has been done in this area, such as single bus route real time control strategy. This thesis will work on the interaction between the 95 and 120 bus routes.

With this real time bus monitoring technology we may control the route 120 bus in the above example. The ideal scenario is that, we just hold the route 120 bus a little time, so as to let the next incoming route 95 bus make the transfer.

Our research work is to evaluate how modern monitoring technologies can be used to improve transfer reliability, and to identify simple dispatching rules and the conditions under which these rules are most effective.

Chapter 2, Research Environment

2.1 Background and Literature Review

A transfer takes place when a passenger has to move from one bus route to another during his or her trip. Basically, there are two streams of research related to this topic found in the literature review: static transfer planning and dynamic transfer dispatching.

Static transfer planning is the preplanning of the route or transit network configuration, the coordination of route schedule, or transfer strategies.

Mao-Chang Shih Hani S.Mahmassani, and M.Hadi Baaj [6] developed a solution methodology for the planning and design with timed transfer for a transit network, with a model composed of four major procedures: route generation, network analysis, transit center selection and network improvement.

Timed transfer is a scheduling strategy whereby particular trips are scheduled to meet at certain transfer points within given time windows. This time window added at the transfer point is to give more chance that the scheduled meeting occurs. But this addition of idle time implies higher operating costs and thus limits the number of transfer points in the transit network, normally less than 10 (Systan Inc., [4]).

Another way studied by James H. Bookbinder and Alain Desilets[5] was transfer optimization of a transit network, which did not aim at the "meetings" of related trips, instead, schedule coordination is done with the goal of minimizing an objective function measuring the overall inconvenience (e.g. average waiting time, quadratic waiting time or variance of the waiting time) of transfers in the network.

Mark Abkowitz et al, [1] developed a 2-route model simulation to investigate the effect of route characteristics on service performance for four kinds of different transfer strategies in the case of two-way transfers(i.e. passengers from both buses meeting at the transfer point could transfer to the other bus): (1) unscheduled, i.e. no coordination between routes; (2) scheduled, i.e. timed transfer but no holding for both routes; (3) single-holding, i.e. low frequency route bus will hold until the next arrival of the high frequency route bus; (4) double holding, i.e. whichever the bus arrives first waits for the next arriving bus from the other route. Their research showed that the route characteristics such as the compatibility of headways of 2 transfer routes, have a direct influence on the feasibility of timed transfer and transfer strategy. Of interest to our proposed research is that if the headways of 2 transfer routes are different (i.e. one bus is running at high frequency, the other is running at low frequency), either "single holding" or "scheduled" are the only good strategies. Abkowitz et al's work still dealt with static transfer planning: their strategies are determined before the transfer occurs and not changeable due to the particular real time situation of the transfer.

Dynamic transfer dispatching is the focus of our research.

As mentioned in chapter 1, the development and introduction into public transit system of real time monitoring technologies provided technical possibility and feasibility to apply dynamic control upon the transit buses.

Important factors to consider in dispatching the buses at transfer stop are strategic, such as choosing between a Global Positioning System or detector based AVL system, tactical, such as finding the best location for the AVL detector to locate the in-coming bus, and operational, such as computing the expected arrival time of the in-coming bus given its location and historical data regarding travel times, and the expected number of passengers transferring based on historical data and time of day.

Kurt Ker-Tsung Lee and Paul M. Schonfeld [2] developed a general numerical approach for real time control of transfer in a transit timed-transfer system. A mathematical total-cost function was built and this function took into account 2 kinds of costs: *connecting line delay cost* – due to the extra holding of the connecting line bus, and *feeder line misconnection cost* – due to misconnection of the feeder line buses if the transfer does not succeed. Each of these 2 costs is a function of the holding time of connecting line bus. Hence we can calculate the optimal holding time to make the total cost function least, based on the current position information provided by the advanced monitoring technologies.

2.2 Our Research Work

Out research work focused on dynamic transfer dispatching or real time bus dispatching using real time monitoring technologies. We extended the assumptions of the current research work and evaluated several real time control methods through simulation. Also we examined both conservative and optimistic assumptions about the quantity and type of real time information that can be obtained from the feeder bus, based on the current and expected real time monitoring technology development.

We studied the transfer from high frequency bus route to low frequency bus route without timed transfer implemented, explicitly split dispatching function and objective function, and introduced a new tool – threshold, which is used to limit the effect of the bus holding.

Real Time Dispatching vs Timed Transfer

Unlike other dynamic transfer control related research, our work will not focus on the timed transfer but on a specific non-timed-transfer situation: transfer is from a high frequency route to a low frequency route. In such a context, timed transfer will cause more limitation and work for route scheduling and designing. Timed transfer will limit the number of transfer stops in the transit network, (Systan Inc.[4]); meanwhile, high-frequency central business district-oriented bus routes are not viable candidates for timed transfer, as the potential reduction in average transfer time derived from timed transfer is not significant enough to substantially increase user satisfaction ([1]); The implementation of timed transfer in a transit network require significant effort in route planning or schedule modification. For example, if a new route is to be added into the current transit network where the timed transfer is implemented, we will not only consider this new route about its assignment and so on, but also need to think about the related routes to or from which the transfer will occur.

Moreover, it is possible to improve service by having a connecting bus wait for a feeder bus at a transfer stop even if a transfer between these two lines was not planned (statically preplanned in schedule coordination) at this stop at that time. Circumstances may be such that the feeder line could be late or early and it makes sense to consider this unplanned event in dispatching the connecting line. We know, that the purpose of implementing timed transfer is to ensure that 2 or more bus routes will meet at certain transfer stop. In our case of high-frequency to low-frequency transfer, if a time interval (could be the maximum waiting time of connecting bus) is well defined, then we also can ensure that there will be a new arrival of the feeder line bus in this interval with relatively high probability. That is to say, this interval should not cause great irregularity to the connecting line, and should be significant to the headway of feeder line. It is feasible due to the large difference between the headways of 2 routes.

A simple example for the applicability of real time dispatching and timed transfer is the following.

If a feeder bus is usually later than the departure of the connecting bus, then it may be a systemic problem which could be resolved by changing either the schedule (departure time) of the connecting line to a later time, or the schedule (arrival time) of the feeder line to an earlier time. It is a timed transfer implementation.

Suppose next that the feeder bus arrives on time generally, but sometimes later and some other times earlier due to the various traffic conditions. In this case, the schedule modification will not work. Instead, we will use real time information based strategies to dispatch the connecting bus, so as to make the transfer successful with greater probability.

By the way, another possibility is that, if the feeder bus is rarely late at the transfer stop, there may be no need to use real time control to dispatch the connecting line to make sure the transfer is successful. Since it may not be worthwhile to implement such a sophisticated and costly solution that would not yield important benefit.

Dispatching Function and Objective Function

Two functions are related to real time transfer control in our research: a dispatching function and an objective function. The dispatching function is used to dispatch the connecting line bus based on real time information, while the objective function is used to measure the result of the real time control. Ideally they are the same function. Sometimes the information needed to measure the efficiency of the control is not available in real time when we dispatch the bus. E.g. the number of passengers transferring is important for measuring the control method. Should we make a possibly faulty estimation of the number of passengers, or ignore it when we make a decision of holding or not, if we have no idea about how many people are on board of the feeder bus? In our simulation model, we will use different dispatching functions for real time control but only one objective function for measuring the effect of this control.

As to the objective function, we will use the format similar to Kurt Ker-Tsung Lee and Paul M. Schonfeld's work[2], but we will explicitly apply cost/benefit analysis.

The basic idea is that we will hold the connecting bus only if the expected benefit due to our control is greater than the expected extra cost due to our control. Here the benefit is a function of the time-saving of transfer passengers from the feeder line bus who might have missed their transfer if no holding had been applied, and the cost is a function of the extra holding time of passengers on board of the connecting bus and of those waiting downstream on the connecting line route. All the benefits and costs should result from the control action, and if no control action is applied, there is neither the benefit nor the cost. Since a control action applied to this system will create both a benefit (time-saving for other passengers) and a cost (longer waiting time for others), it is important to make the benefit outweigh the costs before implementing such an action.

A simple objective function could be the sum of the cost and the benefit. A simple cost measure could be the product of number of passengers on connecting bus and in downstream along the rest of the connecting line route, and the extra waiting time due to the control action. A simple benefit measure could be the product of number of passengers transferring on feeder bus and the saved time due the control action. We will show our definition about the objective function in the following chapter.

Dispatching functions should be simple and realistic: easy to implement and feasible for the real time information. In our model, we plan to examine several dispatching functions (i.e. control strategies) from simple to more complex. There is a description about the details of control actions in the following chapter.

Threshold

We introduce a maximum holding time for the connecting line bus—the *threshold*, which means the control action can never let the connecting line bus hold beyond the scheduled departure time (in the timetable) plus this threshold. Without this threshold, the holding time can be very long (although it is finite) and cause irregularities to the rest of the connecting line trip. Moreover if this connecting line is connected with another route to make transfer, then this irregularity may propagate to other routes. Also the passengers of the connecting line would not appreciate waiting indefinitely because of control as they expect the bus to leave on time and may be unsatisfied otherwise, particularly if they do not know why the bus is waiting.

The threshold should be long enough to allow for a good probability of successful transfer, but short enough so as to minimize adverse effects on the connecting line trip such as excessive lateness. It should be noted, however, that "slack time" is usually built in the schedule. The time between the departure of two consecutive trips is usually longer than the time needed to complete the trip, to allow for travel time variability and break time for the driver at the end of a trip. Hence, the threshold should be short enough to allow the bus not to be late for following trips given the slack time available.

Chapter 3, System Analysis and Modeling

This chapter will introduce details about system definitions, relevant terminologies, real time control actions, mathematical model and comprehensive model. We defined several control actions that can be used to dispatch buses, and we will discuss the procedural operations and other details. For the given problem, we first try to use mathematical tools to build up a simple model, and get some general conclusions from that. Lastly, we will build our comprehensive model in the sense that not only mathematical relationships exist but also we have some descriptive and administrative factors involved. In the following chapter, we will discuss in detail how to use this comprehensive model to develop and run simulation

3.1 System Introduction

Our research is aiming at how real time information can improve the bus transfer and how real time monitoring techniques can be used to make dispatching decision.

3.1.1 System Components

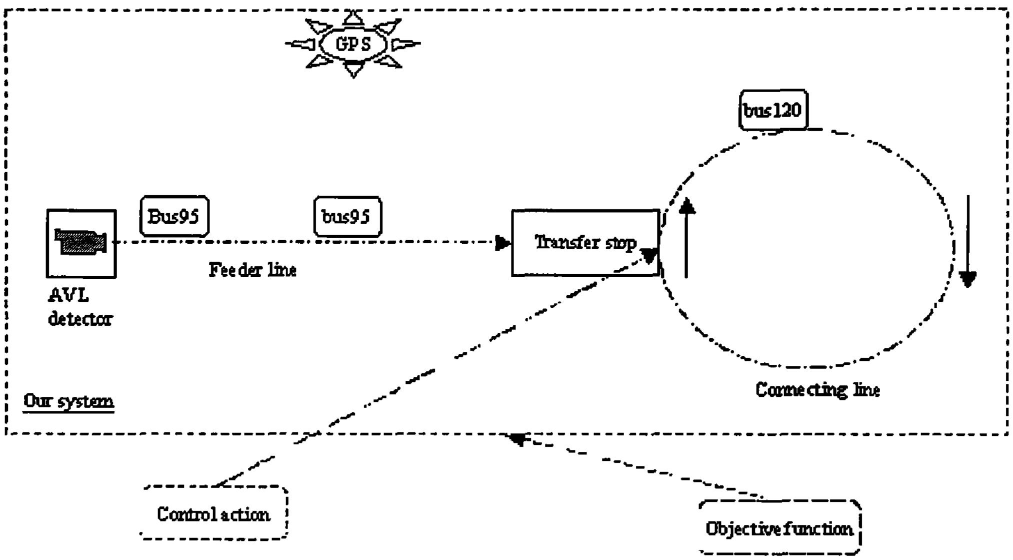


Figure 3.1 System Model

Physically, the system consists of the followings:

- 2 bus routes: feeder line and connecting line.
- Buses including feeder buses and a connecting bus.
- Transfer stop where the feeder buses meet the connecting bus.
- Monitor system, we studied 2 options:
  - AVL detectors
  - GPS
- AVL detector installation location, if this monitor system is chosen. It is typically another bus stop.
- Transfer passengers

Also, we have the following logical compenents in our system logically:

- o Control actions: they are used to dispatch the connecting bus, and dispatching functions are built in each particular control action.
- o Objective function: it is used to measure the system performance.
- o Dispatching function: the core for each control action, is evaluated to dispatch the connecting bus (decision to hold or not).

Generally, a bus route specifies a sequence of bus stops, buses and a timetable. Feeder line is the high frequency bus route which provides the transfer passengers to the transfer stop. It has its own schedule and is running independently with the connecting line. We will primarily study its headway, its travel time from the detector location to the transfer stop, the number of transfer passengers on board of each feeder bus. A feeder bus enters our system from its departure time from the detector location and exits from our system after it arrives at the transfer stop and unload all the passengers. Refer to figure 3.1, route No. 95 is the feeder line and bus 95 is a feeder bus.

Connecting line is the low frequency bus route which takes over the transfer passengers from the transfer stop. It has its own schedule, but at the transfer stop it can be held beyond its schedule departure time due to the control action. We will mainly study its departure time, arrival time, trip travel time and the number of passengers on board. Refer to figure 3.1, route No. 120 is the connecting line and bus 120 is the connecting bus. Route No. 120 is a round trip started from the transfer stop and back.

Feeder bus and connecting bus have the same characteristics but in different transit routes. A bus in our system has its departure time, arrival time and a certain number of passengers on board.

Transfer stop is the bus stop where the above 2 routes connect and some passengers will transfer from one route to the other at this stop. Control actions are applied at the transfer stop to the connecting bus.

Monitoring system is real time electronic devices that provide information about bus location along the transit network.

AVL detectors work together with some electronic device installed on each bus being monitored. When such a bus passes the AVL detector, the detector can record the current time and transmit the information to other devices or human operators. This information can be retrieved and referenced on-line or later. Normally the detectors are installed in some bus stops along a transit route. Refer to figure 3.1, there is an icon of camera used to represent an AVL detector.

Another option is GPS, which is an advanced and expensive real time global positioning system. It is a satellite based technique and can monitor a very large area. If we use this GPS as the monitoring system, we do not need "detector location" in our system. GPS can provide real time information about the current position of a selected bus, when it is requested. Refer to figure 3.1, there is a "satellite based" GPS in the top of the figure.

AVL detector installation location, sometimes we just call it "detector location", is the place where the AVL detector is installed to monitor the buses passing through the stop on-line.

Transfer passengers are not modeled individually but as aggregate numbers. We are only interested in the number of passengers boarding, alighting and transferring, not at their origin or final destination. We assume an average boarding time which is the time used to get on bus, of 2 seconds. This is consistent with figures used by OC Transpo in other simulation modeling (Lule Chen [9]).

Control actions or control strategies are the dispatching actions used to make decision to hold the connecting bus or let it go at once. We will study several control actions comparatively. We will have a detailed description about each of them in this chapter. When we apply some control actions, we specify a threshold as the maximum holding time of the connecting bus. The control actions are applied at the transfer stop to the connecting bus, refer to the figure 3.1.

Objective function is about how to look at the system performance. We used a cost/benefit function, and this function considered 2 important factors: number of passengers transferring, on-board the connecting bus and downstream, as well as the respective waiting times relevant to control action. This function is defined in a way that, there are cost and benefit only when we apply control to the system, i.e. without any control, we assume there is no cost or benefit. In this sense, the cost or benefit is actually an extra cost or extra benefit compared with no such controls applied. This objective function is used to evaluate the whole system, refer to figure 3.1.

Dispatching function is the core functionality of each control action. For example, the control action "Wait First Until Threshold", is described as: when the connecting bus is ready, always wait for the next incoming feeder bus until threshold, whichever comes first. In control "Wait for Good Probability" and "Wait for Good Benefit", the holding and dispatching decisions are made on the basis of the value obtained in computing the dispatching function as discussed later.

The following is a list of the relevant terms and their definitions that will be used in the later discussion.

Table 3.1 Basic Terminology

Feeder line:	the high frequency route, the source of the transfer passengers
Connecting line:	the low frequency route, the destination bus route of the transfer passengers
Transfer stop:	the bus stop where the transfer occurs
S.d.t:	Scheduled departure time, which is given on the timetable for both the feeder line and connecting line;
Bus ready time:	time at which the bus is ready; ready means: the moment at which current time is greater or equal to s.d.t and all passengers in queue have boarded. Only applied to the connecting bus. It is at the bus ready time that a decision to apply or not the control action (hold or not the connecting bus) is made.
Holding time:	the extra waiting time for the passengers on the connecting bus due to the control action;
BET (Boarding End Time)	is used to record the time when the passengers will finish boarding according to the current knowledge about the number of passengers in the waiting queue.
Detector location:	(Normally a stop) where the AVL detector is located, and from where the departure time of the feeder bus can be obtained; AVL stands for Automatic Vehicle Location.
Control actions:	applied only to the connecting bus, to determine its actual departure time based on the real time information regarding the feeder buses; the departure time of the connecting bus is never earlier than its ready

	time.
Measurement:	the long run evaluation of a control action, through an objective function;
Cost:	Is measured as the extra waiting time the passengers on board the connecting bus and downstream of the connecting line, must incur due to the holding decision from the control action.
Benefit:	Is measured by the time saved by the transfer passengers thanks to holding the connecting bus to make the transfer succeed: these passengers save the time to wait for next connecting bus;
Threshold:	the maximum holding time of the connecting bus. Basically, this threshold is determined mainly by relevant factors from outside our system. For example, if a other transfers or bus routes outside our system may be affected if the connecting bus is delayed by more than 3 minutes, so threshold is 3 minutes.

3.1.2 System Operations

A feeder bus enters our system from the detector location, and its departure time from there is recorded for later reference. It travels to the transfer stop, passengers alight and this feeder bus exits from our system. Its schedule is not changed in the system.

We assume a single bus cycles through consecutive trips along the connecting line bus route. The expected total trip travel time is smaller than the headway on this bus route. This bus never exits the system since it is either waiting to leave at the transfer stop, or in transit. However, we are not interested in the details of each trip: only at the trip departure times from the transfer stop as determined by the control actions, and at the trip travel time, which allows us to determine the bus ready time for the following trip. Upon arrival, the connecting bus will delay its departure for the following trip until the "bus ready time". when the bus is ready, control action is applied to determine whether or not to hold the bus or not. If it is held for the next feeder bus, then it will leave after that feeder bus arrives and transferring passengers have boarded. If the feeder arrives earlier than s.d.t plus threshold. If there is no feeder bus showing up, the connecting bus will leave at s.d.t plus threshold and start its next trip.

Passengers enter the system through the feeder bus, and they exit the system "in" the connecting bus when it leaves from the transfer stop. The connecting line has a longer headway than the feeder line, typically 7 times more. So, during the connecting bus's trip, there are several feeder buses which arrived at the transfer stop and many transfer passengers alight at the transfer stop. In the real system some of them may leave to choose alternative transportation means, and meanwhile some other passengers from other sources join in the waiting queue for the next available connecting bus. For simplicity, we assume the total number of passengers waiting for the connecting bus equals the number of passengers alighting from the feeder bus. When the connecting bus arrives, the waiting passengers can board on the bus, and we assume each one takes the same amount of time to board defined as the average boarding time. If the bus leaves at the bus ready time, there is no explicit holding due to control, hence there is no extra cost or benefit incurred. If the bus is held after it is ready, then the passengers on board of the connecting bus and downstream of the connecting line will wait longer than anticipated due to the control action and this extra waiting becomes a cost of the control action. On the other hand, if the next feeder bus catches the connecting bus, the passengers from that feeder will not have to wait for the next connecting bus (a time roughly equal to the headway of the connecting line): this is the benefit of the control action. The difference of the extra cost and the extra benefit is the net benefit of this control action at this time.

Only at the bus ready time, the control action can be applied to the connecting bus. Some information about the location of the feeder may be needed to make the decision to hold or not: this information is provided by the AVL detector or GPS system. Based on this information, the control action uses its logic rules or its algorithm to determine to hold the bus or not. The objective function is used to provide a quantified measurement for the final result of this controlled transfer.

3.1.3 System Evaluation

Generally speaking, the objective is to provide better service to the passengers and particularly, we want to shorten the waiting time for transferring passengers. As discussed in the previous sections of this chapter, the control actions will determine whether or not to hold the connecting bus at a given point in time. Holding may lead to reduction in waiting time for transfer passengers if the transfer is successful as they need not wait for the next connecting bus, and holding will lead to extra waiting time for non-transfer passengers as the connecting bus is asked to leave after its ready time. So we need a balanced solution for this situation. We defined an objective function to evaluate the system performance.

This objective function considered 2 important aspects about the transfer: the number of passengers and their waiting time.

There are several time measures of interest:

1) The headway of the connecting line. This time determines how long the passengers will have to wait if they just missed the current trip of the connecting bus.
2) The s.d.t and threshold. Scheduled departure time is the time at which the connecting bus should normally leave without any control actions. The threshold is the maximum time that the connecting bus would be allowed to be held by control.
3) The actual arrival time of the next feeder bus that the connecting bus is waiting for. This determines the actual holding time of the connecting bus.

Passengers involved in transfer include:

1) Passengers on the connecting bus: if the bus is held after s.d.t, they will wait for extra time.
2) Transferring passengers on the next feeder bus: it refers to the first feeder bus which can not catch the connecting bus if we do not hold the connecting bus past s.d.t. So if we hold the connecting bus, the waiting time of these passengers could be reduced by about the headway of the connecting line.
3) Passengers downstream along the connecting line: they will wait for extra time if we hold the connecting bus.

We define the objective function as follow:

$$\begin{aligned} \text{NetBenefit} &= \text{benefit} - \text{cost} \\ &= Nf * Ts - Nc * Tw \\ &= Nf * (HW - Td) - (Nb + Nd) * Tw \end{aligned}$$

where:

NetBenefit is the net benefit, which is the difference between cost and benefit ;

Nf is the number of passengers who reduce waiting time due to the control;

Nc is the number of passengers who wait extra time due to the control,

$$Nc = Nb + Nd$$

Nb is the number of passengers on board of the connecting bus;

Nd is the number of passengers waiting in downstream along the connecting route;

HW is the headway of the connecting line;

Ts is the time that is saved for some passengers due to the control;

Tw is the extra waiting time for some passengers, or the bus holding time.

Td is the connecting bus delay time with regards to its schedule.

There are other potential benefits not being considered in this objective function. First, non-feeder passengers arrival during the bus holding will generate some benefit. However normally few connecting route passengers will arrive to the stop a few minutes later than sdt, since the bus schedule information is quite public and easily accessible and the consequence of missing connecting bus is to wait 30 minutes for the next one. Secondly, there are possibly passengers downstream of the connecting route catching the connecting because of the bus holding at the transfer stop. For the same reason, it is unlikely that many passengers will arrive a few minutes later than schedule and would benefit from the bus holding. Another reason is that transfer of interest (from high frequency route to local route) typically becomes more important during peak hour in the afternoon when people go back home from work. During this period, there are few people transferring from home to work, or these transfers are less important than the transfers in the other direction. In any case, our function is conservative in that it is underestimating the benefit.

From the objective function, basically we expect the good control actions to have a large positive number as the net benefit (also called total benefit in later chapters). If one control action has a greater net benefit than another, we will have the conclusion that the first one is better. However, usually we do not analyze the absolute value of the net benefit of one control action or one scenario by its own.

We give an example here to illustrate how we use the objective function. Suppose the connecting bus is scheduled to leave at 8:00AM, but it is not ready until 8:02AM because there are passengers boarding and at this bus ready time, there are 20 passengers on board. Then we apply control action and decide to hold it for the next feeder bus. The next feeder bus with 5 transfer passengers arrives at 8:04AM, then transfer passengers board on the connecting bus, and the connecting bus actually leaves at 8:05. The connecting line has headway of 30 minutes and there are 10 passengers waiting in the downstream of the connecting route. For this example, we have the following values defined:

Nf = 5

Nc = Nd + Nb = 10 + 20 = 30

HW = 30 min

Tw = 8:05 – 8:02 = 3 min

Td = 8:05 – 8:00 = 5 min

$Ts = HW - Td = 30 - 5 = 25$ min

So, the evaluation is:

$$NetBenefit = Nf * Ts - Nc * Tw = 5 * 25 - 30 * 3 = 35$$

This is the measurement about the control action we chose.

On the other hand, if we do not apply any control, we will have the following. The s.d.t is 8:00AM and the connecting bus is ready at 8:02, then it just leaves without any holding. For this "No Control" scenario, we define it has NetBenefit of zero, remember the rule used to define the objective function: there are cost and benefit only if there is extra holding due to control.

3.2 Control Actions

Simply speaking, control actions are the dispatching of the connecting bus. The decision to hold the connecting bus or not is based on the real time information provided by the monitoring system, such as AVL and GPS.

In this section, we define several control actions and for each control action, we define what is the operation procedure and what could be the actual departure time of the connecting bus.

For a real time transfer control, we can choose one of these control actions and use this control for a certain period of time. Particularly, when the connecting bus is ready, there are 2 options: to go at once or to hold. At this time, we will follow the rule we choose – the control action to make a decision: to hold or to leave. If a holding decision is made, we then need to determine when the connecting bus should leave.

We define control action in a general sense, and even no-control is a control action, which is used for comparison purposes.

3.2.1 General Introduction

When the connecting bus arrives at the transfer stop, we will go through the procedures described in the table below.

Table 3.2 The General Transfer Control Procedure and Logic

(1). Connecting Bus Arrival Time or the Current Time	(2). To Apply Control?	(3). Decision Made	(4). Actual Departure Time of the Connecting Bus
Earlier than s.d.t or BET	No. defer to time of max{sdt, BET} to check this table again	N/A	N/A

At bus ready time (later than both of sdt and BET, but no later than sdt plus Threshold)	Yes. Options are: 1. No Control 2. Wait First 3. Wait Next 4. Good Prob 5. Good Benefit 6. Prob GPS 7. Benefit GPS	If hold,	If the next feeder bus arrives no later than sdt plus threshold, the connecting bus will leave from then on whenever all transfer passengers finish boarding.
			If the next feeder bus arrives later than sdt plus threshold, the connecting bus will leave at sdt plus threshold. There should not anybody boarding.
		If leave,	The connecting bus will leave right now, and we already know that no passengers are boarding.
Later than sdt plus threshold	No.	N/A	The connecting bus will leave from now on whenever all passengers finish boarding.

Before we discuss the above table, we repeat several concepts. Sdt is the scheduled departure time of the connecting bus at transfer stop. Threshold is the maximum holding for the connecting bus. Bus ready time is the time when the connecting bus is "ready" to leave, i.e. the current time is not earlier than sdt and there is no passengers boarding. We use this to specify when we can apply control.

BET is Boarding End Time. There are two ways to compute BET. First, when the connecting bus arrives at the transfer stop, BET is changed to the sum of the current time and total passengers boarding time (number of passengers times average boarding time). Second, while the connecting bus is at the transfer stop, when a feeder bus arrives, the BET is reset by the following algorithm:

- If the current time is later than the current BET, there is no passengers boarding now, so BET is reset to the sum of the current time and total passengers boarding time (number of passengers from this feeder bus times the average boarding time).

- If the current time is no later than the current BET, there are some passengers boarding now, so BET is reset to the sum of the current BET and total passengers boarding time (number of passengers from this feeder bus times the average boarding time)

So, BET is a representation about when the passengers boarding will finish based on the current knowledge about the number of passengers in the waiting queue.

BET is introduced to model the passengers boarding time, and especially to deal with an interesting scenario: while some passengers are boarding on the connecting bus, another feeder bus arrives. Using BET, this possibility can be modeled realistically: the connecting bus has to wait for these passengers to finish boarding, and only if there are no passengers boarding can it leave from the transfer stop.

Table 3.2 should be read from left to right. The most left column is 3 possibilities of the actual arrival of the connecting bus, or to be exact, it is the time when to apply control to the connecting bus. Each column is the consequence of its left neighbor.

Let's look at the table 3.2. When the connecting bus arrives no later than sdt or BET, we can not make any decision to hold or not. We say it is not the bus ready time yet. So we simply defer to the maximum of sdt and BET to make the decision then.

When the connecting bus arrives later than sdt, it will check this table at time BET. When the current time is later than both of sdt and BET but no later than sdt plus threshold, i.e. at the bus ready time, we apply control action to the connecting bus. The actual decision and algorithm are totally dependent on the control action selected. But the decision made is only to hold the connecting bus or not. In the table 3.2, we listed the corresponding departure time for each case.

When the connecting bus arrives later than sdt plus threshold, we let it leave whenever there is no passengers boarding.

The following is a specific example for the dispatching function of "Wait With Good Probability Within Threshold" (sometimes we also call it "Wait for Good Probability" for simplicity) of the table 3.2. It also shows the relevant event generation process – listed as “generate event ‘control’ at time sdt, and it basically means to go through this control process again later, at time sdt”. We will see details about event generation in chapter 4. (HW: the headway of the connecting line).

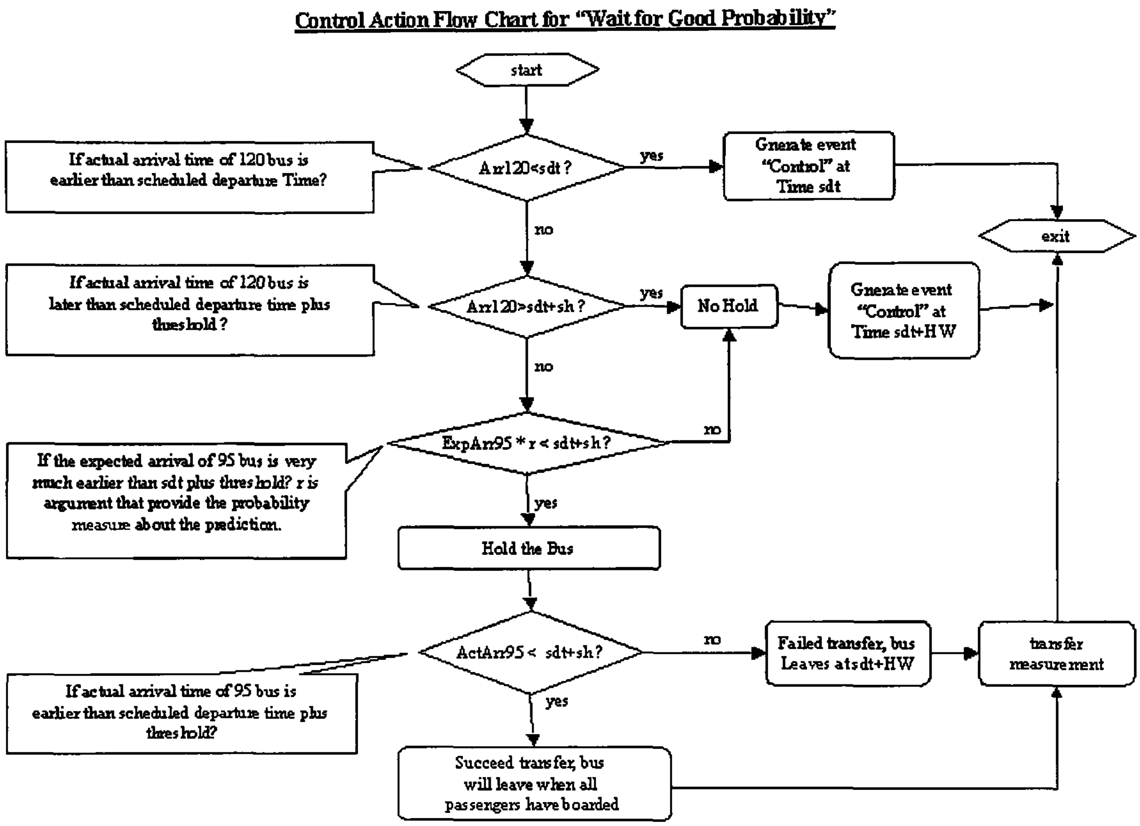


Figure 3.2 Control Action Flow Chart

In the following sections, we will introduce each control action listed in “ Apply Control” column of table 3.2. We defined control action in a general sense, i.e. any method could be used to dispatch the connecting bus, but not necessarily using real time information, and even no control is a control action.

3.2.2 No Control

This is the base case where there is no explicit control over the bus at all, i.e. the departure time of the connecting bus will always be its bus ready time. This scenario is used for comparison purposes with other control actions.

Because of the randomness in trip travel time due to traffic jams, or excessive number of passengers and so on, it may take more time than scheduled for the connecting bus to complete its trip, and arrive at transfer stop; thus, the connecting bus may be late. On the other hand, it may be earlier than schedule although earliness is not big issue in our research. Therefore, "No Control" does not mean the connecting bus will always be on time. However, we define this case has zero benefit in our objective function: this is the baseline case.

The reason that earliness is not in our research interest, is that: we assume the connecting bus will wait until at least the scheduled departure time on time table before leaving. This is to say, passengers do not expect the connecting bus to leave early, so their waiting time before s.d.t needs not be included in the analysis.

In this case, the actual departure time of the connecting bus is connecting bus ready time, and by its definition the bus ready time is always later than or equal to s.d.t.

3.2.3 Wait First Until Threshold

In this case no real time information is used. The connecting bus will always wait for the first feeder bus that arrives in the time interval [s.d.t, s.d.t + threshold]. The actual procedure is: when the connecting bus is ready, it checks if there is any feeder bus which already arrived in the above time interval: if yes, it goes at once; otherwise, it will wait for the next incoming feeder bus within threshold.

Particularly, if the connecting bus arrives later than s.d.t, it will check back to see if there is already a feeder bus arrived after s.d.t. If yes, it will not hold any more. Otherwise, it will wait for the next feeder bus.

In later chapters, this control action is also called "Wait First" for simplicity.

3.2.4 Wait Next Until Threshold

In this case no real time information is used. This is identical to 3.2.3 "Wait First Until Threshold", when the connecting bus is on time, i.e. when it does not arrive later than s.d.t. The operation is simple: after the connecting bus is ready, wait for the next feeder bus until threshold.

The only difference with the previous case 3.2.3 "Wait First Until Threshold" is that, when the connecting bus arrives later than s.d.t, it will not check what happened before it arrives, and will wait for the next feeder bus, even if a feeder bus arrived between s.d.t and the connecting bus ready time.

In later chapters, this control action is also called "Wait Next" for simplicity.

3.2.5 Wait With Good Probability Until Threshold

Under this scenario, the departure time from the detector location of the feeder bus which is next to arrive is available: however, no information about the number of passengers on board is available. Based on the travel time distribution, we know the probability that the feeder bus will arrive before the s.d.t plus threshold. That is the only real time information we get when we dispatch the bus. If this probability is acceptable (high enough), we will hold the connecting bus. The core function is to check the defined dispatching function to determine the probability of catching the feeder bus before sdt plus threshold.

In later chapters, this control action is also called "Good Probability" for simplicity.

3.2.6 Wait with Good Benefit Until Threshold

For this control action, we assume that all relevant real time information is available, particularly the departure time of the feeder bus from the detector location and the number of passengers on that feeder bus. Or at least, we have means to calculate the expected value based on historical data. Given this real time information, we can calculate the expected cost and benefit, and then decide to hold the connecting bus or not.

Cost/benefit dispatching function:

E[totalBenefit] = E[Ts] * E[Nf] – E[Tw] * (Nc + E[Nd])

(here, E[X] represents "expected X", which is an a prediction about X.)

where E[totalBenefit] : the expected benefit/cost measurement, benefit minus cost;

E[Ts]: expected time-saving by the passengers on the feeder bus due to our real time control; E[Ts] is the time between the arrival time of feeder bus and s.d.t of next connecting bus.

E[Tw]: expected extra waiting time of passengers on connecting bus and downstream of connecting line; E[Tw] equals the holding time of the connecting bus past the ready time.

E[Nf]: number of transfer passengers on feeder bus;

Nc : number of passengers on connecting bus, suppose the bus driver or the controller know this number exactly;

E[Nd]: predicted number of passengers in downstream of connecting line;

Theoretically, when E[totalBenefit] is greater than zero, the holding should be better than no holding, but we could set a minimum acceptable value to make this benefit significant in practice.

We would assume that this control action should by default include the "probability checking" process of the "Good Probability" control action. To see this note that this control action first evaluates the probability that the feeder bus will arrive at the transfer stop before sdt plus threshold, as in "Good Probability"; if this test is passed, it evaluates the expected benefit before making a decision. If the probability does not meet the requirement, the connecting bus will not be held, and the benefit expectation will be not evaluated.

In later chapters, this control action is also called "Good Benefit" for simplicity.

3.2.7 Wait With Good Probability Using GPS

This is operationally the same as the control action 3.2.5 "Wait With Good Probability", except that the monitoring technique is different. GPS provides more updated and accurate real time information that AVL in our system. So, this is not a complete different control action but only used to compare the difference of using these 2 monitoring techniques. The difference is that dispatching decision made at the connecting bus ready time uses information about location of feeder at the connecting bus ready time, rather than the information about when the feeder left the detector stop. Hence, the information is more up-to-date, we have a better idea of when feeder will arrive, given we know its exact position at the connecting bus ready time, and less variability in the arrival distribution is expected, since the feeder bus should be closer to the transfer stop at the time we get real time information about its location.

In later chapters, this control action is also called "Good Probability GPS" for simplicity.

3.2.8 Wait With Good Benefit Using GPS

This is operationally the same as 3.2.7 "Wait With Good Benefit Using GPS", except that it checks both the probability and the benefit to determine to hold the connecting bus or not, while 3.2.7 control only checks the probability.

In later chapters, this control action is also called "Good Benefit GPS" for simplicity.

Generally, the above control actions can be categorized into 3 classes:

(1) 3.2.2 to 3.2.4 are not using real time information.

(2) 3.2.5 and 3.2.6 are using AVL detector information.

(3) 3.2.7 and 3.2.8 are using GPS real time information. The installation of GPS system requires a higher investment for the transit property than a detector-based AVL system.

3.3 Mathematical Modeling

Here we apply some mathematical tools to analyze the system we discussed in the previous section. Mathematical modeling is useful in the sense that key relationships may be defined and quantified. However, we can not fully address the aforementioned details by mathematical relationship: instead, we focus on a simplified model.

3.3.1 No Detector Case With Only the Randomness of Feeder Line

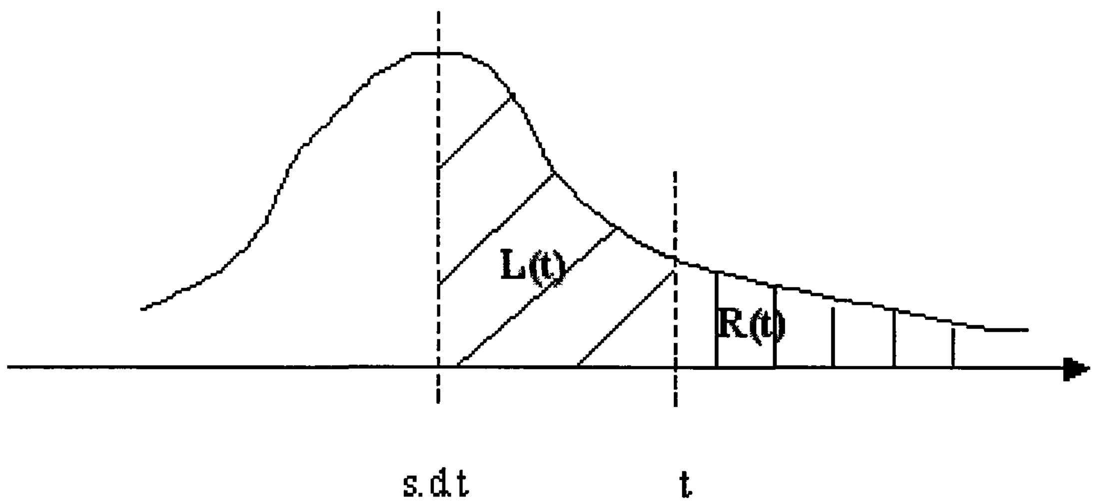


Figure 3.3

Now we will make some assumptions so as to have a simple model and apply mathematical tools straightforwardly. In this case, the main assumptions about our system are:

1. No real time information concerning feeder bus;
2. No threshold, i.e. it is assumed there is no limit to the holding time of the connecting line;
3. We study the Control action "wait for feeder bus which is next to arrive". This is the simplest control action to study analytically;

4. We assume the connecting bus always returns from its trip on time for the following trip (no lateness of connecting bus);

5. Feeder buses do not pass each other, and at time s.d.t, we know which is the next feeder to arrive and know the arrival distribution of that next feeder;

6. Relevant information about number of passengers is available.

Let T be the actual arrival time of feeder bus (No. 95 bus), which is a random variable with distribution function of $f_T(t)$. In the above graph, s.d.t is the scheduled departure time of the connecting bus, t is a certain time point at which the next incoming feeder bus could arrive. L(t) refers to the probability that the feeder bus arrives between s.d.t. and time t, and R(t) refers to the probability that the feeder bus arrives after time t.

At time sdt, one feeder bus is coming but we do not know which. We do not know when it left from the previous bus stop. We assume no bus passing for simplicity, otherwise we would need to look at the minimum of all possible feeder bus arrivals. We assume a < sdt < b and we have a holding decision to make only if the arrival of feeder is after sdt. By definition, no holding implies, no benefit or cost, so if the connecting bus is not held, the benefit is zero. Hence a conditional arrival time distribution is used here. The conditional feeder bus arrival time distribution, given the feeder bus has not arrived at time sdt is given by:

$$g_T(t) = f_T(t) / \int_{sdt}^{\infty} f_T(t)\, dt \qquad \text{---------------} \ (1)$$

for t > sdt.

Let B(t) be our cost/benefit function, and it is a function of the feeder arrival time:

$$B(t) = Nf\,[HW - (t - sdt)] - Nc\,(t - sdt)$$

$$= Nf * HW - (Nf + Nc) * (t - sdt) \qquad \text{---------------} \ (2)$$

Where t is a particular arrival time of 95 bus, and t-sdt is the holding time. When t is less than or equal to sdt, B(t) = 0, from the point that there is no associate benefit if there is no holding. Nf is the number of passengers related to feeder line (they will get benefit from the control), Nc is the number of passengers related to connecting line (they will pay some cost for the control), and HW is the headway of connecting line.

For any feeder that arrived before s.d.t, the passengers transferring will have a successful transfer without the need for a control action. We wish to see what is the expected benefit (positive or negative) of holding until the next feeder arrival.

Based on the above formulas (1) and (2), we can have the expected benefit as:

$E[B(t)] = \int^{\infty}_{sdt} B(t) * g_T(t) * dt$ --------------(3)

This is only for control action "wait next feeder" (but not "wait next within threshold"), which means the connecting bus will always wait for the next feeder bus without threshold. The control action "wait first feeder" is the same with this model, because we assume the connecting line has no randomness, i.e. the 120 bus is always on time (i.e. it is ready no later than s.d.t).

3.3.2 Numerical Analysis

3.3.2.1 Uniform Distribution as Feeder Bus Arrival Time

We will look at an example. We assume some basic data here: Nf=5, Nc=15, HW=30, sdt=10 and feeder bus arrival time distribution is uniform distribution in interval [a, b]. The following graph is the distribution function for feeder bus $f_T(t)$.

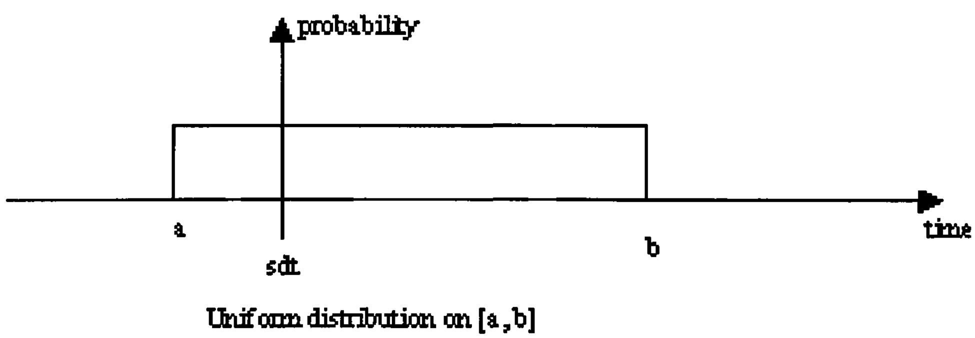


Figure 3.4 Uniform Distribution

Then the distribution function of feeder bus arrival time is

$f(t) := 1/(b-a)$, if $a <= t <= b$;

$f(t) := 0$, otherwise.

Then the distribution function of the feeder bus arrival time, given it does not arrive before or at time s.d.t, is

$g_T(t) = = f_T(t) / \int_{sdt}^{\infty} f_T(t)\, dt = 1/(b-sdt)$, refer the formula (1) in this section

Then, $E[B(t)] = \int_{sdt}^{+\infty} B(t) * g_T(t) dt = \int_{sdt}^{b} (Nf*(HW-(t-sdt))-Nc*(t-sdt))/(b-sdt)\, dt$

So,

$E[B(t)] = HW * Nf - (Nc+Nf) * (b-sdt)/2$ --- (generic function)

This is a generic form of expected benefit as a function of all relevant arguments.

The following graph is the curve for $B(t)* g_T(t)$ in interval [10, 16], with uniform distribution on [8, 16] (a=8, b=16). We can have:

$E[B(t)] = \int_{10}^{16} (5*(30-(t-10))-15*(t-10)) / 6\, dt$,

We can see that without considering other factors, more holding, less benefit. The shadow area is the integration on [10, 16] and the value of E[B(t)] is 90. The conclusion is that for long run operation, this control has an average benefit of 90.

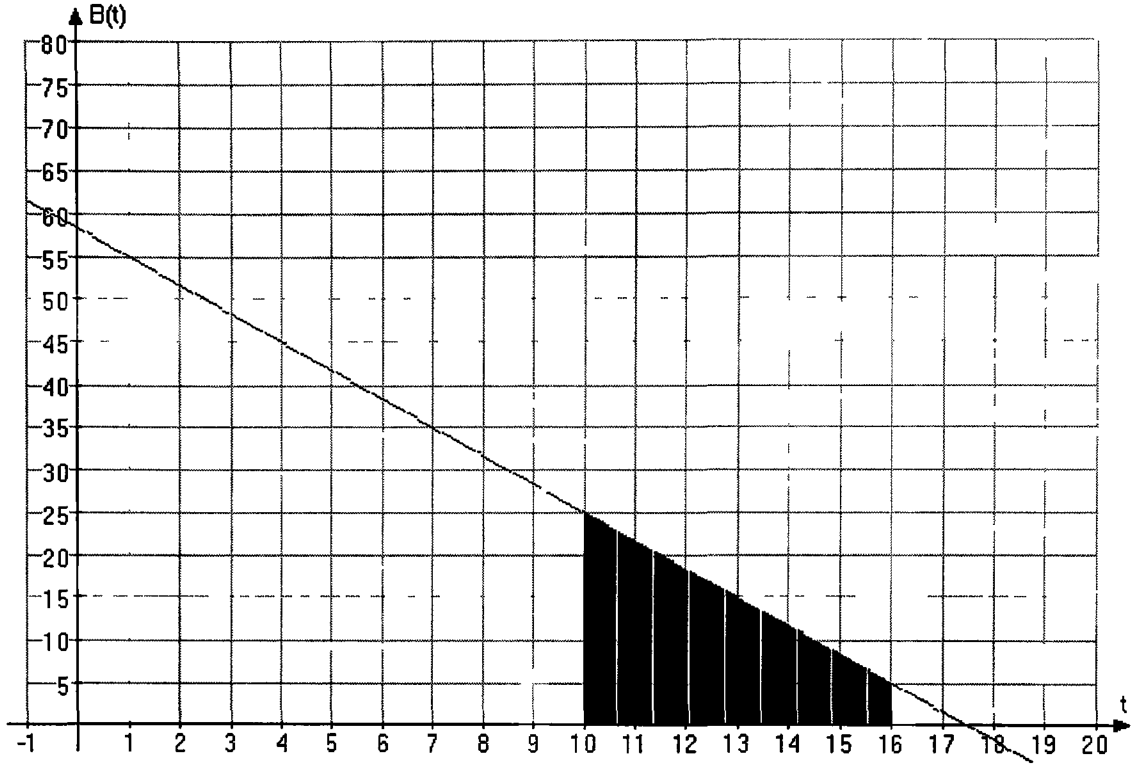
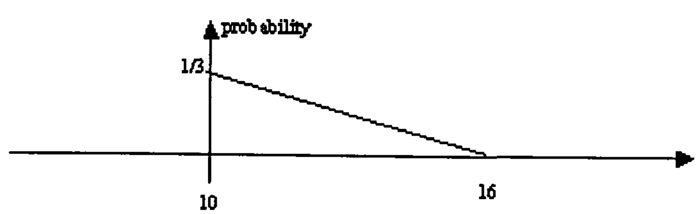


Figure 3.5

3.3.2.2 Triangle Distribution as Feeder Bus Arrival Time

We let $f_T(t)$ to be a triangle distribution, see the following graph and is on the interval [10, 16]. The actual meaning is that the feeder bus can not arrive earlier than s.d.t and later than 6 minutes after s.d.t. The probability is decreasing linearly while the time is later and later.



triangle distribution on [10, 16]

Figure 3.6 Triangle Distribution

For all other arguments we still use the same data as mentioned in the above example (see uniform distribution case). We have $f_T(t) = 1/3 - t/48$, by definition and $\int^{\infty}_{sdt} f_T(t)\, dt = 3/8$, so:

$g_T(t) = = f_T(t) / \int^{\infty}_{sdt} f_T(t)\, dt = 8/9 - t/18$

$E[B(t)] = \int^{+\infty}_{sdt} B(t) * g_T(t) *dt = \int^{b}_{sdt} (5*(30-(t-10))-15*(t-10))\ (8/9-t/18)dt$

Thus we have the following graph for $B(t)* g_T(t)$ and the expected benefit (the shadowed area) E[B(t)] is 110.

In general format, let's assume the triangle distribution start from sdt and end at time b. Thus we have the following equation by the probability definition:

$(b - sdt) * h * ½ = 1$, so, $h = 2 /(b\text{-}sdt)$;

And the point (b,0) is on the line, we can have the function as follows:

$$f_T(t) = \frac{-2 * t}{(b - sdt)^2} + \frac{2*b}{(b - sdt)^2}$$

Since $g_T(t) = = f_T(t) / \int^{\infty}_{sdt} f_T(t)\, dt$, and $\int^{\infty}_{sdt} f_T(t)\, dt$ is 1. So, $g_T(t) = f_T(t)$. The expected benefit function for this case can be get by the following:

$E[B(t)] = \int^{+\infty}_{sdt} B(t) * g_T(t) *dt$,

$$= \int^{+\infty}_{sdt} [Nf*(HW-(t-sdt))-Nc*(t-sdt)] * g_T(t) *dt$$

$$E[B] = \frac{2}{b - sdt} \{ (Nf * Hw * b + (Nc + Nf) * sdt * b$$

$$- \frac{b+sdt}{2} [Nf * Hw + (Nf+Nc) * (b + sdt)]$$

$$- \frac{(b^3 - sdt^3) * (Nc+Nf)}{3(b - sdt)} \}$$

3.3.3 No Detector Case with Randomness of Feeder and Connecting Line

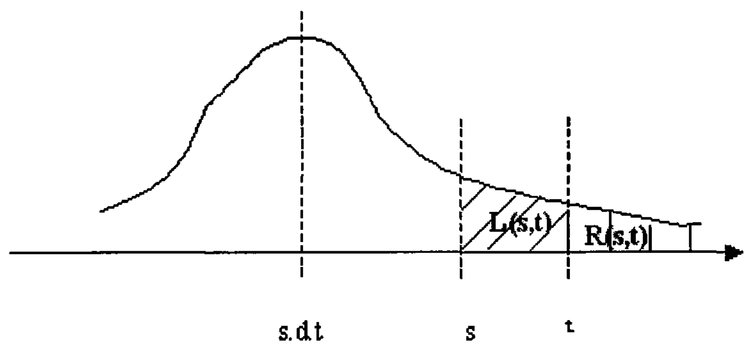


Figure 3.7

In this case, we just remove assumption No. 4 "No lateness of the connecting line" from the list, and the others still apply.

In the above graph, s.d.t is the scheduled departure time of the connecting bus, s is a certain time point when the connecting bus could arrive, t is a certain time point when the next incoming feeder bus could arrive. L(s,t) is the probability that the next incoming feeder bus arrives between time s and time t. R(s,t) is the probability that the next incoming feeder bus arrives after time t. The curve is still the distribution function of arrival time (variable T) of the next incoming feeder bus $f_T(t)$. In addition, let C be the random variable of actual arrival time of 120 bus, with distribution function $f_C(s)$.

If the arrival time of the connecting line is a random variable, then there are 2 possibilities: 1) the connecting bus arrives no later than sdt, and 2) the connecting bus arrives later than s.d.t. In the first case, all the analysis and model in 3.3.1 are applicable. For the second case, we need to do a little more analysis.

Possibility 1. 120 arrival time s <= s.d.t

The same as in 3.3.1 (no randomness of connecting line), we get the same cost/benefit function B(t), and the same expected benefit E[B(t)].

In this case, we suppose 120 arrives earlier or at the time s.d.t, and it is the same assumption made at section 3.3.1, therefore, the above analysis applies here.

Possibility 2. 120 arrival time s > s.d.t

Hence we define the distribution of the feeder arrival time as

$g_T(s,t) = f_T(t) / \int_s^{\infty} f_T(t)\, dt$ ----------------- (4)

Where s is an arrival time of the connecting bus, t is an arrival time of the feeder bus, and s<t is always true.

The cost/benefit function becomes:

$B(t, s) = Nf\,[HW - (t-s)] - Nc\,(t-s)$ -------------(5)

The expected benefit is:

$E[B(t, s)] = \int_s^{\infty} B(t, s) * g_T(s,t) * dt$ -------------(6)

Combine the 2 possibilities together, we may have overall expected benefit as:

$E[B] = E[B(t)] * P(C<=sdt) + E[B(t, s)] * P(C>sdt)$

So,

$$E[B] = E[B(t)] * \int_{-\infty}^{sdt} f_C(s) * ds + E[B(t, s)] * \int_{sdt}^{+\infty} f_C(s) * ds \quad \text{---------(7)}$$

Therefore, formula 7 is the expected benefit for this case.

Generally, the above mathematical model provides a means to measure the simplified control action "Wait for Next" in terms of the expected benefit. That is to say, given some historical or experimental data, such as number of passengers, distribution of feeder bus arrival time and connecting bus arrival time and etc, we can have a quantified measurement about this control action under the simplifying assumptions made. However, the drawback is that the mathematical relationships become complex rapidly, even with simplifying assumptions.

3.3.4 Further Discussions

3.3.4.1 Why Introduce Monitoring Techniques

We will remove the assumption (1) (no detector) from the list in 3.3.1 (no detector with only the randomness of feeder line).

Let D be the random variable for actual departure time of the feeder bus from detector location.

Let Tr be the random variable for actual travel time from detector to the transfer stop, and its distribution function is $f_{Tr}(t)$.

Let T be the random variable for actual arrival time of feeder bus at transfer stop, and its distribution function is $f_T(t)$. We have: $T = D + Tr$.

First, we assume the connecting bus is ready before or at time s.d.t, and that the feeder bus is not arrived yet. Without real time information, the decision to hold or not the connecting bus would be based on expectation or the value of D (based on scheduled time tables) and on Tr (based on historical data) in order to find T. But if we know the departure time from the detector location is d, then the probability of this feeder bus arriving at time t is:

$$P(T=t \mid T>sdt, D=d) = P(Tr = t-d \mid Tr > sdt-d) \text{ ------------------ } (8)$$

With real time information, we are dealing with another random variable Tr with smaller mean and variance, instead of studying T. In our real time control actions, we need to know some information about the incoming feeder bus so that we can have an "expected arrival time" of this bus. The control action will primarily be based on this prediction. Therefore, with the help of the AVL detectors, we can make a more confident prediction of T, because we know D. We are thus more likely to make the right decision about holding or not the connecting bus. This is the reason for introducing new technologies into real time bus dispatching. Generally, using real time monitoring technology will improve the real time prediction, so as to make the service more reliable and efficient.

3.3.4.2 Why Threshold

Our benefit function: $B(t) = Nf * [HW - (t - sdt)] - Nc * (t - sdt)$, where $t > sdt$, otherwise, $B(t)=0$, by our definition. Obviously, when t increases, (t - sdt) increases, but B(t) decreased. Simply speaking, more holding, less benefit. We do not want to have a negative benefit for our control actions, i.e. $B(t) > 0$. Thus,

$$Nf * [HW - (t - sdt)] - Nc * (t - sdt) > 0,$$

So: $t-sdt < Nf * HW / (Nf + Nc)$.

The conclusion is that, given Nf, Nc and HW, the maximum holding is $Nf * HW / (Nf + Nc)$. Here we only consider the relevant 2 routes in the system of our interest.

However, for each individual bus transfer, Nf and Nc vary and these numbers might not be available in real time, so we prefer to give the system a fixed threshold, which is easy to operate in practice. But we can try several values for this threshold to compare the simulation results.

For example, if HW=30 minutes and we know Nc/Nf =9, i.e. there are 9 times as many passengers on the connecting line as on the feeder line, then, we can set the threshold to 3 minutes.

If HW=30, and we know Nc/Nf=14, i.e. there are 14 times as many passengers on the connecting line as on the feeder line, then, we can set threshold to 2 minutes. Compared with the above scenario, it means we should not hold the connecting bus too long, since there are relatively more people on the connecting line.

From the above graphs in section 3.3.2.1, we can see that the expected benefit is going down towards zero over time and possibly negative. Therefore, there is a certain time point after which we will not get any benefit if we continue holding the bus.

3.3.4.3 Why Simulation

In this section, we discussed some simple mathematical models for our research work, and get some basic conclusions. However, this approach is not very easy to use to model some administrative and logical operations in reality. For example, to consider the passengers boarding time, we suppose it is about 3 seconds for each passenger to get on the bus. The bus should not leave if there is any passenger left in queue, and new passengers could be added to the queue upon the arrival of another feeder bus. This is complex to model mathematically. On the other hand, such details can more easily be included in a simulation model. Therefore, we will make our model more realistic and more detailed in computer simulation.

3.4 Comprehensive Model

So far, we have introduced our system of research and discussed some mathematical work about simplified models. Now we will build a more complicated model and use computer simulation to study the system.

3.4.1 Feeder Line

The feeder line consists of a sequence of feeder buses traveling from the detector location to the transfer stop.

A feeder bus enters into our system when it leaves from the detector location or the bus stop upstream of the transfer stop along the feeder route. A gamma distribution is used to model its departure from the detector location. It has a certain number of passengers on board and this number is a uniform distributed random number for simplicity.

A gamma distribution is used to model the travel time from detector location to the transfer stop and this random number has mean equal to the average travel time from detector to the transfer stop. Richard P. Guenchner and Kasimin Hamat's [3] research shows that Gamma distribution is better than Normal, which was used by most researchers to model the bus traveling time.

Feeder line has a fixed headway, so the average interval between 2 feeder buses leaving the detector location has mean equal to this headway.

Upon arrival, all transfer passengers will join the waiting queue for the connecting line immediately, and no transfer to other routes is modeled. We will not model any passengers alighting and boarding time explicitly, since we assume this time is included in the travel time. Then this feeder bus exits from the system.

3.4.2 Connecting Line

The connecting bus departs from the transfer stop with a number of passengers from the transfer stop. A gamma distribution is used to model its total trip travel time – start from the transfer stop and back to the transfer stop, and it has a fixed headway, which is the difference between 2 consecutive scheduled departure times in the timetable. Richard P. Guenchner and Kasimin Hamat's [3] research shows that Gamma distribution is better than Normal.

The passenger alighting time at the transfer stop is included in the travel time. So, upon arrival, the connecting bus is empty and can start to board passengers. Meanwhile it checks its own schedule to see if it is later than s.d.t. The connecting bus is ready only when the time is equal to or later than s.d.t and there are no passengers boarding – the passenger boarding time at the transfer stop is modeled as a fixed number for each passenger. When the connecting bus is ready, the control action is applied ("No Control" is also considered as a special case of control). If it is not held, it leaves at once and starts its next trip. If it is held, it will wait for the next incoming feeder bus. If the next incoming feeder bus arrives before s.d.t plus threshold, the connecting bus will be held to let all the transfer passengers finish boarding and then leave. If the next incoming feeder bus has not arrived at time s.d.t plus threshold, the connecting bus will leave at s.d.t plus threshold. The passengers alighting and boarding time at other bus stops along the connecting route are included in the travel time.

All the passengers on board of the connecting bus when it leaves the transfer stop are from the transfer stop.

3.4.3 Transfer Stop

The key of the transfer stop is the passengers waiting queue. This queue is modeled as just a positive integer number. When a feeder bus comes, the number of transfer passengers will be added into the "queue", and when the connecting bus leaves this queue will be reset to zero.

3.4.4 Monitoring Systems

AVL detectors and GPS system are modeled only in a logical way, i.e. we did not model its physical devices. For AVL detectors we model its installation location that is one of our research interests, and certainly we model its functionality that it can provide real time information about the departure time of a feeder bus from the detector location, or the passing time from detector. Generally, when the detector is installed too close to transfer stop, for example when the traveling time from the detector to transfer stop is less than the feeder line headway, there could be sometimes the real time information is not available since no feeder bus is between detector and transfer stop. On the other side, if the detector is installed to far away from transfer stop, the real time information provided by the detector becomes stale. For example, if one detector is installed 5 minutes away from the transfer stop, thus we know when the feeder bus passes the detector so as to calculate its expected arrival time based on the variance about this 5 minutes, which should be small such as 3 minutes. If another detector is installed 10 minutes away from the transfer stop, then we will expect its variance to be larger, such as 6 minutes. Therefore in the latter case the detector provides worse real time information than in the former case, in term of the information accuracy.

For GPS system, only the functionality is modeled: whenever it is requested, it can provide the current position of a feeder bus.

Compared to detector technology, GPS can provide more up-to-date information – whenever it is requested. With the detector technology, there are usually some delay between the actual time at which the feeder bus left the detector and the time when the real time information is requested by the connecting bus (sdt normally), GPS provides accurate information regarding the exact location of the feeder bus at the time this information is requested by the connecting bus.

3.4.5 Passengers and Measurement

The passenger numbers are generated from a uniform random variable. Measurement is the objective function we discussed in the previous sections.

Benefit = Nf * Ts – Nc * Tw

where:

benefit is the net benefit;

Nf is the number of passengers who saved waiting time due to the control;

Nc is the number of passengers who wait extra time due to the control;

Ts is the time that is saved for some passengers due to the control;

Tw is the extra waiting time for some passengers;

3.4.6. Control Actions and System Operations

The description about the system operation as a whole can be found in the first section of this chapter and our model will be consistent with that description. The control actions used to dispatch the connecting bus can be found the second section of this chapter.

Chapter 4. OCTranspo Case Study and Simulation

This chapter will introduce how the simulation program was implemented to evaluate the modeled bus transferring. The topics include: the simulation model, data sources and important design and implementation details about the computer program.

4.1 Introduction and Simulation Model

OCTranspo (Ottawa-Carleton Public Transit Agency) route 95 is the backbone of the transit network. Route 95 crosses the Ottawa region from East to West and back, going through the downtown area. It operates on the East-West segment of OCTranspo's Rapid Transit way, which is a roadway dedicated to the exclusive use of OCTranspo buses. It has a short headway which is about 4 minutes, so it is high frequency bus route. Route 120 is a local bus route, which connects with route 95 at Place d'Orleans bus stop. Along the route 95 before Place d'Orleans, there is a bus stop at Blair, where an AVL detector is installed. So, each No. 95 bus passing Blair is recorded.

So far, there is no timed transfer implemented between No.95 and No.120. Both of them run under "schedule control" separately, i.e., all buses try to follow their own timetable independently, and to adhere to their schedule.

The simulation is designed and developed according to the comprehensive model introduced in the previous chapter.

Generally, the model consists of: route No. 95 buses (the feeder buses), route No. 120 bus (the connecting bus), bus stop at Blair (the detector location), bus stop Place d'Orleans (the transfer stop). Optionally, GPS is an alternative for AVL detector. There is a series of control strategies and an objective function to measure the system performance.

1. Feeder Line

Route No. 95 is the feeder line and it provides transfer passengers into the transfer stop in our model. No.95 buses are running on their own schedule. Route No. 95 buses come from its origin and will pass the stop Blair so its passing time is recorded and used to predict its arrival time at Place d'Orleans. Once it arrives at Place d'Orleans, all the transferring passengers alight and this No.95 bus leaves the system. These transfer passengers will wait for the next available No. 120 bus if there is no No. 120 bus available in the Place d'Orleans. If there is a No.120 bus in Place d'Orleans, then these passengers will join the queue and board on the bus one by one.

No. 95 route runs mainly on the transit way which is exclusive to OCTranspo buses. Its headway at the peak hours is about 4 minutes, according to the timetable.

2. Connecting Line

Route No. 120 is the connecting line and it takes over transfer passengers from the transfer stop in our model. No. 120 bus may be held under certain control action.

No.120 is a local bus with headway of about 30 minutes. Its trip is a round trip: starts from Place d'Orleans and back to Place d'Orleans at the end. The traveling time of this round trip is about 24 minutes, and the remaining 6 minutes (headway minus travel time) is the slack time.

3. Transfer Stop

Bus stop Place d'Orleans is the transfer stop in our model and route No. 95 and route No. 120 connect at this bus stop.

4. Detector Location

An AVL detector has been installed at Blair bus stop, which is about 10 minutes upstream from Place d'Orleans on No. 95 route; this detector could provide the real time information of the departure time of No.95 bus from Blair.

5. Monitoring System

(1) AVL detector

At OC Transpo, an AVL system has been under development since the late 1980's, and switched into use in 1995. Tests and evaluations have been done by phases and have shown this AVL to be a highly effective and efficient system.

(2) GPS

GPS is an advanced technique in monitoring moving objects. It is not installed in OC Transpo. So this option needs new investment, and it is out of our research to provide a clear conclusion about this issue. However, our research could be one of the important considerations in decision making.

6. System Performance Evaluation

The objective function is used to measure the system performance. As we mentioned before, it has following representation:

NetBenefit = Nf * Ts – Nc * Tw   (refer to chapter 3 for details)

4.2 Simulation Data

The data sources include: real OC Transpo time tables (OCTranspo, 1998), intuitive approximations and assumptions. For each argument there is a typical default value, such as No. 95 route has a headway of about 4 minutes. This value will be used in the simulation by default. But for the comparison and simulation purposes, we also used some alternative values in simulation runs. For example, we run simulation program with 5 minutes headway of No. 95 route, as to compare the corresponding simulation result with that of the 4 minutes headway simulation run. Another example, the current travel time from the AVL detector location (Blair bus stop) to the transfer stop (Place d'Orleans pus stop) is about 10 minutes (typical default value). In simulation, we can choose several values such as 4, 6, 8, 10, 12 or 14. In this way, we can have a general experimental conclusion about how far away we should install the AVL detector, so that we can have the best measurement, such as in total benefit.

4.2.1 Headway

Both of the headways of feeder line (route 95) and connecting line (route 120) are from OC Transpo timetable for 1998. In fact, their headways are not consistent all the time in a day, and particularly it is shorter in the peak hours than that in the non-peak hours. We chose only the headway of the peak hours as the headway in the simulation. It is because, at the peak hours, more people take public transit buses and normally they are commuters who are busy to go to work or back home. So better transfer service is quite important.

4.2.2 Travel Time

Travel time is a random number due to various traffic conditions, like road condition, traffic light, traffic accidents and so on. We can not model every detail for all these factors, and also we lack the real data. So we choose Gamma distributed random numbers as the travel time for both feeder buses and connecting bus. Many researchers used Normal distribution to model the travel time, but Richard P. Guenchner and Kasimin Hamat's [3] research shows that Gamma distribution is better than Normal.

It has the curve shape like in the following graph. The tail in the left side is short and big, and the tail in the right side is long and narrow. This feature meets at least 2 basic requirements for modeling travel time: first, most of the distribution are in the middle, i.e. around the scheduled departure time in time table, second, it is unlikely to arrive much earlier than s.d.t but can arrive much later than s.d.t. For example, if the average travel time is about 10 minutes, it is not likely that a bus can finish traveling this distance in 7 minutes, but it may take as much as 15 (10+5) minutes.

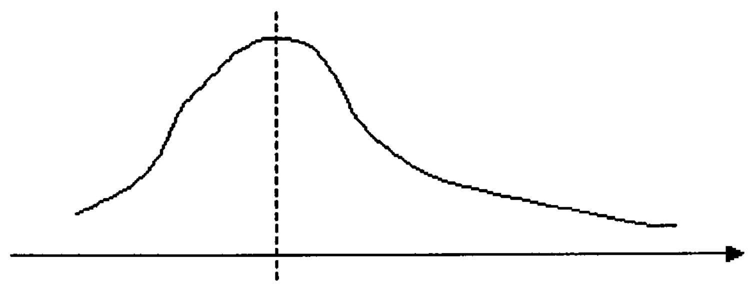


Figure 4.1 Gamma Distribution

4.2.3 Number of Passengers

For simplicity, we assume it is a uniform distribution. We will define its upper and lower limits from intuitive observation and assumptions.

Basically, when a connecting bus is held, there are more passengers waiting on board of this bus than that in the next incoming feeder bus (route 95). The reason is that those passengers waiting on route 120 bus are from the previous 6 to 8 feeder buses.

We assume the number of passengers on board of a feeder bus is the uniform distributed random variable on interval [10, 20] including transfer and non-transfer passengers. We assume the number of passengers waiting along No. 120 route in the downstream is a uniform random variable on interval [10, 30]. The number of passengers waiting on board of No. 120 while it is held, is the accumulation of all the transfer passengers from previous arrival feeder buses and passengers from other sources.

We assume that the proportion of transfer passengers to all the passengers on a feeder bus is 0.3. For example, if there are 10 passengers on a No. 95 bus, there are 3 passengers will transfer to No. 120 bus.

4.2.4 Other Data

Passengers boarding time has typical value of 2 seconds and it is assumed that every passenger use the same time to board.

4.3 Simulation Program Design and Implementation

A simulation program is designed and developed to simulate the OCTranspo route 95, route 120 and transfer. The control actions we discussed in previous chapters are implemented and the simulation program can output relevant data such as the total benefit, and total holding time of the connecting bus.

4.3.1 Introduction

We decided to design and implement our own simulation program for this research work, in order to have full control over the model and the simulation program as so to customize them to meet our requirements. The general requirements for the simulation are: correct functionalities and easy of modification. The performance, optimization and other considerations are less important. No existing simulation software have been found that could conduct our experiments directly.

We studied 2 types of simulations: distributed and centralized. In a distributed implementation model, a central program sends out timing information to all the other sub-systems and each sub-system works in its own way based on its schedule. If required, one sub-system can communicate to another. A centralized simulation model uses a central program to drive most of its operations: each sub-system will define its own rules about how to do the work and the central program will notify the sub-systems what to do and when to do it. A distributed model is good to simulate parallel operations of sub-systems but it can be more complicated than the other approach. This "parallel operation modeling" feature is not absolutely required for our research, so we choose a centralized implementation model. Since the system or the components in the system have no dependence on time, we choose to use a discrete event-driven simulation approach.

This simulation program is developed in Java, which is a fully object oriented programming language. It can run on Windows, Unix and Macintosh platform. It has a graphic user interface to specify the simulation parameters. Java is one of the most popular application programming languages and relevant software, support, service and application are very common and normally free of charge. This will provide us a reliable, low-cost and easy-to-get developing environment. A specific simulation language, which is developed typically for a particular area or functionality, lacks these advantages.

"Object oriented" feature of Java makes it more feasible for a single person to design and implement a whole functional program. The important step of Object Oriented design is Class design, which consists in building a computer model of the real system. This Class model typically has a straightforward mapping to the logical model. This makes it much easier to validate the simulation program design and to discuss with the experts in the application domain.

Figure 4.3.1.1 shows a generic flow chart of the simulation program. Generally, the user can define some parameters before starting the program, such as how long you would like the simulation to run in terms of simulation time. The simulation then initializes some data used as the start point of the simulation, such as the first event and the initial system state.

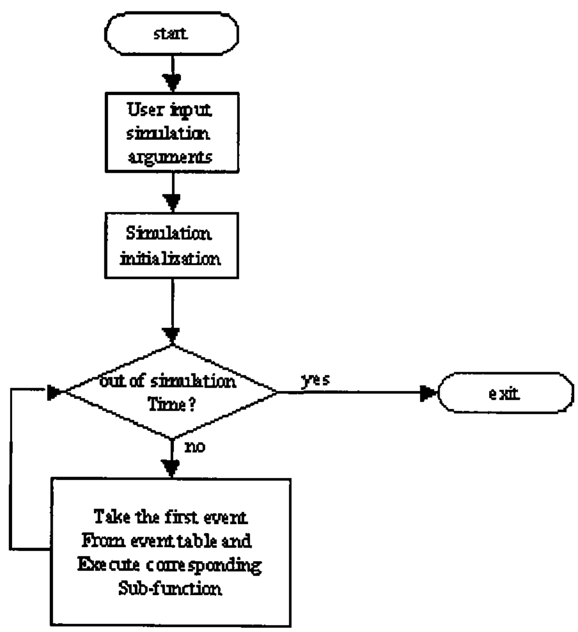


Figure 4.2 General Simulation Flow Chart

The next step is to take the first event from the event table and execute the corresponding sub-routines. An event is used to represent that something occurs and the system state should be changed. The simulation program keeps running only if there are still some events in the events table. Events table lists events in the order of occurrence time: events will be fired one by one according to this order. Event can and should generate another event so as to make the events table not empty. System state can be seen as some static data used to describe the system, and when an event is fired, the program will change these data following some predefined rules.

The simulation program will keep on running until the user defined simulation time is gone.

Practically, the bottom big block shown in Figure 4.3.1.1 involves a lot of work and it can have very complex structure and model. We will discuss some of the details in the later sections of this chapter.

4.3.2 Class Model

A Class is a representation template of a kind of real world entities or logical entities. Basically a Class has a series of data used to model the real entity's relevant information, and a series of functions or sub-routines which can check, modify and operate this Class's data.

The basic idea of this simulation program Class design is that, for each physical item in real life, there is a Class in program used to represent this real item. In addition, there are some Classes for logical or conceptual item, such as TransferManager, and these Classes work on bus dispatching, transfer management, system evaluation and so on.

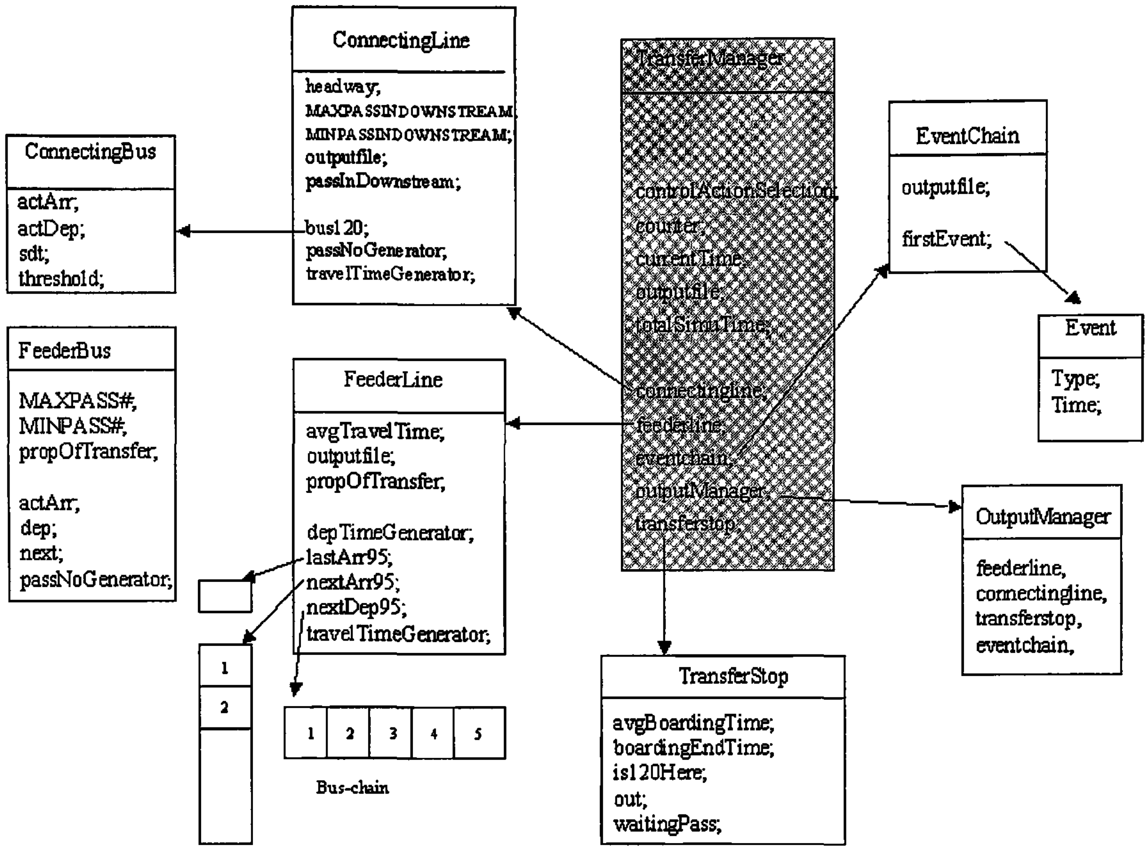


Figure 4.3 Class Model

Basically, figure 4.3 is a description of the important designed Classes and data fields inside each Class. The title in the upper part of each block is the name of each Class, and the items in the lower part are the data field used to model the relevant information about the real entity. Subroutines associated with each class are not listed.

For example, we have a Class called Event which is used to represent the event discussed in previous section, and Class Event has data fields like *type*, *time* and so on. So when we have an instance of Event (a particular event), we can know what is its type and occurrence time. The system state is basically the sum of all the data fields shown in the above graph, although it is not necessary in our simulation to define what consist system state. Therefore what happens in the program is that, when the main program knows that an event occurs, it will change the relevant data fields shown in the above Classes graph and generates relevant new events shown in the graph in the previous section. Note, the above graph is only for illustration about the Class model design, but does not include all the Classes in the simulation program.

4.3.3.Event Model

Event does not take any time to occur and the system state switch is done in no time either. How the system states change will depend on which event occurs, on the other hand, system state has some influence on which event is going to occur next as well. For example, when route 120 bus arrives at the transfer stop (Place d'Orleans in our real case), we model it as an event of "120Arrival", and it will cause that a certain number of transferring passengers will aboard the connecting bus and there will be no passengers waiting at the transfer stop after all of them have boarded on the connecting bus. Then the connecting bus will check its schedule and make decision to hold or leave and so on.

The important information contained in the event are the occurrence time of this event and what is the event type. In our system, 6 types of events are defined. Every time an event is triggered, we will have an instance of this type of event. For each type of events, how the system states change is totally dependent on the type and the occurrence time of this event.

In the simulation program, an event is generated by another event. While an event occurs, there are 2 important work to do: first, to changes the system state, and second to generate another relevant event. The simulation program maintains an event table, which is a list of events that will occur later chronologically.

The figure 4.4 shows the events and their generation relationships. We use the following format to present an event: ***eventName(occuringTime).*** ***EventName*** is used to represent the event type, and ***occuringTime*** is used to represent the occurrence time of this specific event.

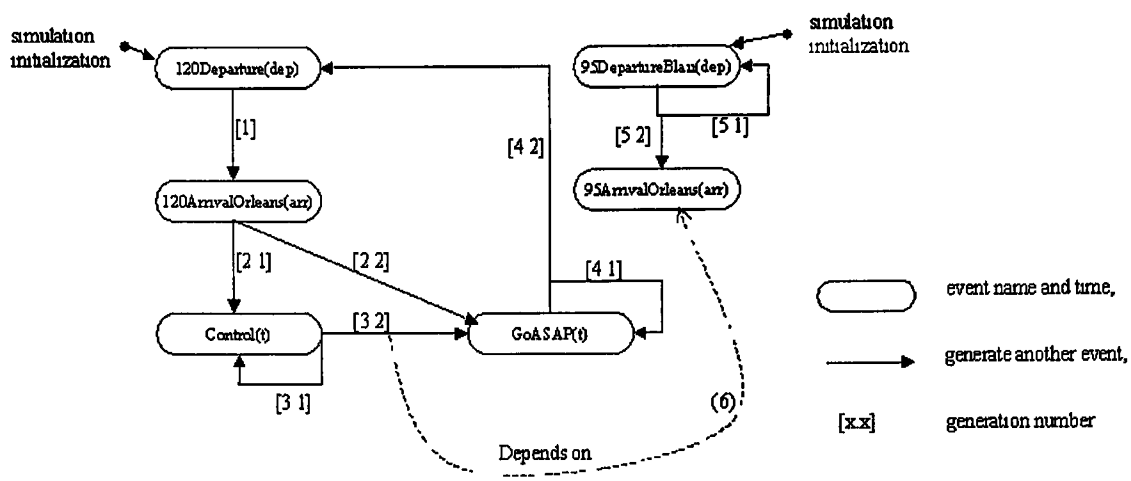


Figure 4.4 Event and Event Generation

An event means something happened in the system. The following table lists the meanings of each of 6 kinds of events in our simulation.

Table 4.1 Event Introduction

No.	Event	Description
(1)	95DepartureBlair(t)	95 bus departs from Blair at time t;
(2)	95ArrivalOrleans(t)	95 bus arrived at Orleans at time t;
(3)	120Departure(t)	120 bus leaves Orleans at time t;
(4)	120ArrivalOrleans(t)	120 bus arrives at Orleans at time t;
(5)	Control(t)	we apply a chosen control action on the 120 bus at time t;
(6)	GoASAP(t)	120 bus has already decided to leave from time t;

The following table gives some explanations of event generation in the figure 4.4 (the number is from the graph).

Table 4.2 Event Generation Details

No.	Event	Event Generated	Description
[1]	120Departure(dep)	120ArrivalOrleans(arr)	Whenever 120 bus leaves the transfer stop, it generates its owns arrival event back to transfer stop, to take place at time arr. The travel time is from random number generator
[2.1]	120ArrivalOrleans(arr)	Control(arr)	When arr (actual arrival time of 120 bus) is less than sdt plus threshold, it generates event Control automatically at the same time.
[2.2]	120ArrivalOrleans(arr)	GoASAP(t)	When arr is greater than sdt plus threshold; especially, in "no control" case, there is only this branch, but no (2.1).
[3.1]	Control(t1)	Control(t2)	Here t1 and t2 are not equal. When control action can not determine the departure time at time t1 (in fact, it is the time it want to leave, but may or may not be possible, due to the limitation of the BoardingEndTime), the program simply delays the time to make such a decision to the earliest known time - t2, at which this decision is possible to be made.
[3.2]	Control(t)	GoASAP(t)	Here t is the same. When control action is done and the program can determine the exact departure time of 120 bus (again, it is its wish only), we will generate an event to let it "go as soon as possible".
[4.1]	GoASAP(t1)	GoASAP(t2)	When the 120 bus wish to leave but can not, because some passengers are boarding. The program will delay this to the earliest possible time point.
[4.2]	GoASAP(t)	120Departure(t)	When the bus will go and can go, we will generate a new 20Departure(t) event at the same time t.
[5.1]	95DepartureBlair(dep1)	95DepartureBlair(dep2)	Whenever a 95 bus leaves Blair. It needs random number generator. A feeder bus leaving Blair generates the departure event of

			another feeder bus to take place at time dep2.
[5.2]	95DepartureBlair(dep1)	95ArrivalOrleans(arr)	Whenever a 95 bus leaves Blair. It needs random number generator. A bus leaving Blair generates its owns arrival event to Orleans to take place at time arr.
[6]	Dependency of (3.2)	95ArrivalOrleans(arr)	Some control actions may check the arrival of a 95 bus.

Another important work is to change the system state. As mentioned in section 4.3.1, (figure 4.2, flow chart), the bottom block is very complex. Generally, this event handler consists of 6 event-handling sub-routines. One of them will be invoked by simulation program when an event is triggered. Each of the event-handling sub-routine changes the relevant system states so as to reflect and record the occurrence of this event. The following is a general list of what each event-handler does. Some of the contents below are for the purpose of the simulation program development (refer to the Class Model) and events generation are also included.

1. if the first event is 95DepartureBlair(t):

   New events: 95Departureblair(…) and 95ArrivalOrleans(…)

   Related classes: FeederLine

   Operations:

   (1). Update information in FeederLine;

   (2). Generate new event 95DepartureBlair(t);

   (3). Create new event 95ArrivalOrleans(t);

2. if the first event is 95ArrivalOrleans (t):

   New events: none

   Related classes: FeederLine and TransferStop

   Operations:

   (1). Update information in FeederLine;

   (2). Update the information in TransferStop.

3. if the first event is 120Departure(t):

   New events: 120ArrivalOrleans()

   Related classes: ConnectingLine and TransferStop

   Operations:

   (1). Update the information in TransferStop.

   (2). Update the information in ConnectingLine.

   (3). Generate a new event 120ArrivalOrleans().

4. if the first event is 120ArrivalOrleans(t):

   New events: Control(t)

   Related classes: ConnectingLine and TransferStop

   Operations:

   (1). Update the information in TransferStop.

   (2). Update the information in ConnectingLine.

   (3). Generate new event Control().

5. if the first event is GoASAP(t):

   New events: GoASAP() or 120Departure()

   Related classes: TransferStop

   Operations:

   (1). get bET from TransferStop;

   (2). if(t<bET), then to create event GoASAP(bET).

   (3). if(t>=bET), then to create event 120Departure(t).

6. if the first event is Control(t):

   New events: Control() and GoASAP()

   Related classes: ConnectingLine and TransferStop

   Operations:

   (1). get data from ConnectingLine and TransferStop;

(2). Apply control: the actual work is depending on which control action is used. Refer to the control action section for details of each control action.

(3). Generate new event Control or GoASAP.

4.3.4 Random Number Generation

Random number generation is one of the basic data sources of the simulation. There are primarily 3 numbers to generate: the arrival time of route 95 buses at the detector location (Blair bus stop), the travel time from detector location to the transfer stop of the route 95 buses, and the trip travel time for route 120 bus.

For the above 2 travel times, we use Gamma distributed random variables to model them. Researchers showed that Gamma distribution can be used to represent the travel time better than Normal distribution.

The following is the Gamma Random Number Generation algorithm for gamma(a,1), Best's rejection algorithm XG (1978)

```
b =a-1, c=3a - 3/4,
Repeat
        Generate iid uniform(0,1] r.v.:  U,V;
        W=U(1-U), Y= (c / W)½ (U - ½), X=b + Y;
        If X >= 0;
                Then Z =64 * W* W * W * V *V
                Accept =[ Z <= 1 - (2* Y*Y)/X]
                If Not Accept ,
                Then Accept = [log(Z) <= Z (b* log(X/b) - Y)]
Until Accept;
Return X.
```

Table 4.3 Generated Gamma Random Numbers (part)

G	#	G	#	G	#	G	#	G	#	G	#	G	#	G	#
489	3	519	14	549	22	579	99	609	143	639	37	669	10	699	4
490	2	520	17	550	33	580	112	610	94	640	45	670	18	700	9
491	5	521	12	551	37	581	111	611	98	641	46	671	12	701	7
492	4	522	10	552	27	582	104	612	63	642	29	672	14	702	3
493	3	523	16	553	35	583	124	613	57	643	35	673	15	703	6
494	4	524	4	554	35	584	123	614	61	644	41	674	12	704	3
495	3	525	11	555	28	585	138	615	36	645	44	675	15	705	6
496	5	526	19	556	33	586	152	616	44	646	28	676	7	706	5
497	4	527	15	557	37	587	115	617	45	647	34	677	13	707	1
498	5	528	14	558	43	588	148	618	42	648	27	678	11	708	3
499	6	529	9	559	48	589	147	619	53	649	28	679	9	709	3
500	3	530	8	560	56	590	158	620	44	650	40	680	5	710	7
501	4	531	14	561	43	591	160	621	37	651	25	681	12	711	5
502	5	532	14	562	54	592	166	622	42	652	25	682	8	712	1
503	5	533	12	563	49	593	175	623	49	653	19	683	9	713	5
504	7	534	16	564	57	594	175	624	47	654	23	684	6	714	1
505	3	535	17	565	55	595	180	625	52	655	17	685	6	715	4
506	7	536	16	566	54	596	176	626	50	656	19	686	6	716	3
507	7	537	21	567	53	597	157	627	31	657	35	687	9	717	0
508	6	538	20	568	70	598	177	628	39	658	27	688	13	718	4
509	10	539	20	569	88	599	172	629	36	659	19	689	8	719	2
510	6	540	15	570	74	600	195	630	40	660	19	690	9	720	3
511	2	541	23	571	71	601	174	631	55	661	17	691	9	721	5
512	7	542	21	572	84	602	167	632	29	662	18	692	10	722	0
513	6	543	22	573	84	603	167	633	29	663	21	693	14	723	2
514	7	544	19	574	85	604	151	634	27	664	13	694	7	724	3
515	4	545	24	575	78	605	158	635	41	665	18	695	2	725	5
516	5	546	38	576	98	606	170	636	33	666	15	696	4	726	2
517	7	547	22	577	104	607	132	637	31	667	17	697	3	727	3
518	13	548	28	578	88	608	137	638	36	668	14	698	4	728	5

There are totally 10,000 gamma distributed numbers generated with expected mean of 600, and the above table shows part of the distribution data table.

G: the value of generated Gamma distributed number.

#: how many such Gamma numbers generated.

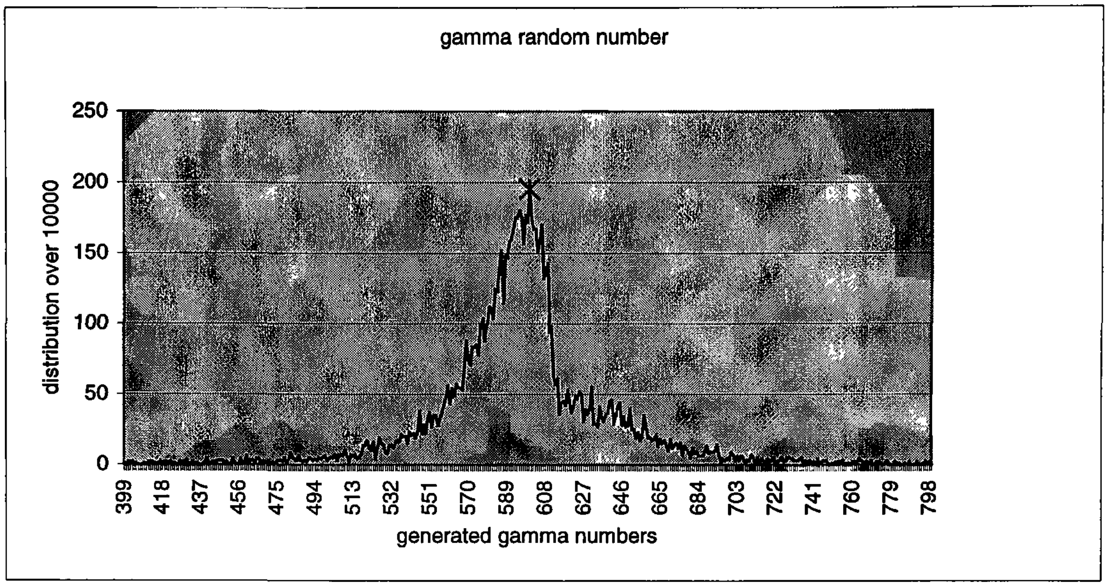


Figure 4.5 Generated Gamma Random Numbers

The highest point in the above graph (where there is a cross in the top), is the point with G=600 and #=195, i.e. 195 gamma numbers generated with value of 600, and it is the highest distribution.

Another set of the random numbers are the number of passengers involved. We use uniform distribution to model it for simplicity.

4.3.5 Feeder Line Implementation Details

Feeder line is one of the key elements in the simulation. Refer to the following graph Figure 4.6 (from graph 4.3, Class Design), we will explain the feeder line implementation in the simulation.

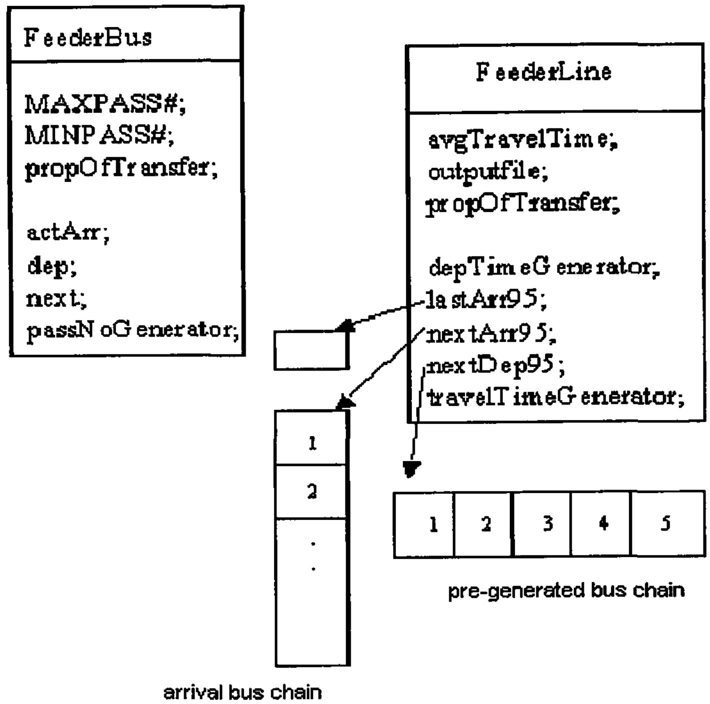


Figure 4.6 Feeder Line Implementation

To better illustrate the implementation details, let's look at the following graph Figure 4.7 which is another point of view to the same question.

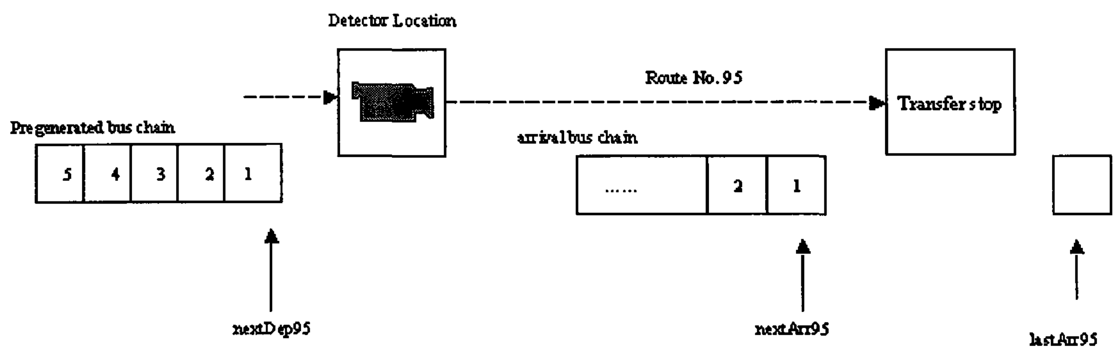


Figure 4.7 Feeder Line Illustration

During the simulation initialization (before any meaningful events really occurs), the program generates a bus chain shown in graph as "pregenerated bus chain" which is used to model those buses that will arrive at Blair (detector location) in order. From the graph, bus 1 in the right side is the next feeder bus to arrive at Blair, bus 2 is the second and so on. The length of this chain is 5, i.e. we always maintain a chain of 5 buses. When a bus leaves from Blair (passes the detector), it is removed from the "pregenerated bus chain" and inserted into the other chain "arrival bus chain". At this time, the simulation program generates a new bus and add it into the "pregenerated bus chain".

Here we can see that, the "bus generation" is done much sooner than when the bus passes the detector or enters our system explicitly. First, let's look at what the "bus generation" means. Basically "bus generation" should define all relevant data about a feeder bus and make it ready to enter our system. Particularly, the following data need to be defined:

- Arrival time at Blair, i.e. detector passing time.

  This is a gamma distributed random number with mean equal to the scheduled arrival time at Blair; as specified by official OC Transpo timetable.

- Expected arrival time at Orleans (transfer stop).

  This is the sum of arrival time at Blair and the expected travel time from Blair to Orleans.

- Actual arrival time at Orleans.

  The actual traveling time from Blair to Orleans is a gamma distributed random number with mean equal to the expected travel time of this distance. The actual arrival time of a feeder bus is the sum of its detector passing time and the actual traveling time between Blair and Orleans.

- Number of passengers on board.

  It is a uniform distributed random number with predefined maximum number and minimum number of passengers.

Therefore, "bus generation" means all the above numbers are made to be available. "pregenerated bus chain" sets the order of the arrival time at Blair of each generated feeder bus.

The "arrival bus chain" is to model the feeder buses currently traveling from Blair to Orleans, so it is in the order of the actual arrival time at Orleans of each feeder bus, which is generated during "bus generation". The length of this chain is not fixed and is depending on the detector location. For example, if the average traveling time from the detector location such as Blair is about 10 minutes and the feeder line has headway of 4 minutes. Normally, there should be 2 to 3 buses in this chain, but not necessarily. This chain may to be empty.

When a feeder bus leaves from Blair, it joins the "arrival bus chain" immediately. When a feeder bus arrives at Orleans, it is removed from this chain. Note that the order of buses in the pre-generated bus chain and in the arrival bus chain is not necessarily the same (i.e. some bus passing may take place between Blair and Place d'Orleans).

Refer to the graph below: bus a has its schedule of arrival time at Blair and bus b has its own schedule which is normally one headway later, and we assume they have independent, identical distribution of Gamma. The shaded area is where bus passing can happen. For example in the graph, if bus a is very slow and bus b is very fast, then bus b arrives earlier than bus a (arrival time shown as an arrow).

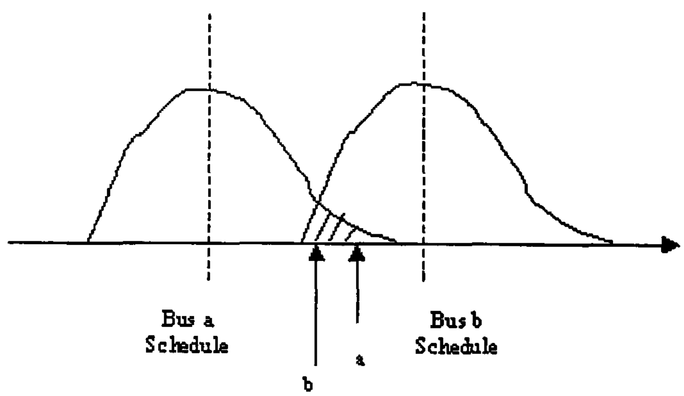


Figure 4.8

However, during the "bus generation", we have no idea if the next bus generated will be "faster" than an existing one. We can not have the current generated feeder bus enter our system until we can make sure that there will not be any potential bus passing issues before they arrive Blair. For this reason, the simulation is designed to generate buses much earlier than when they will enter the system by passing the detector at Blair, and all the bus passing issues are resolved inside the "pregenerated bus chain". Here is an example. The time in each block (bus) is its arrival time at Blair generated during "bus generation".

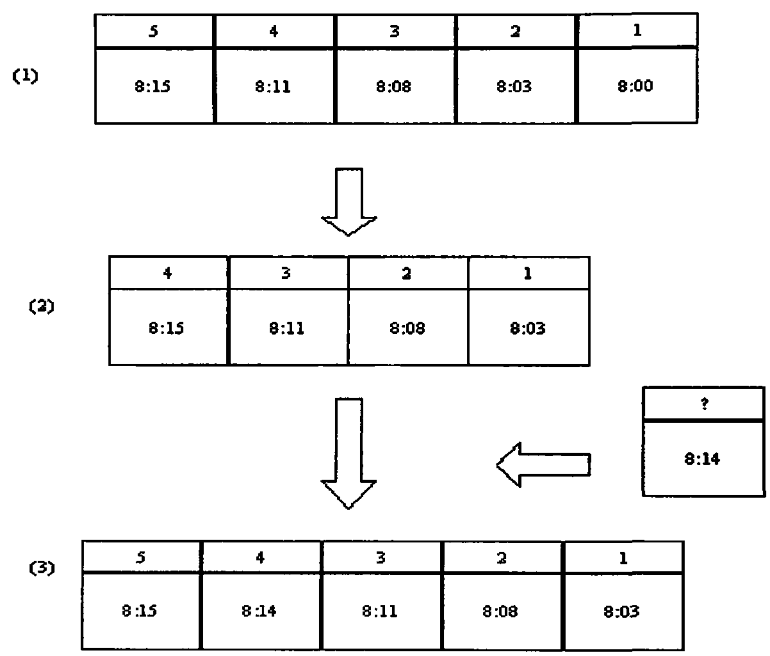


Figure 4.9 Arrival Bus Chain Example

Bus chain (1) is the original, and at time 8:00 the first feeder bus in this chain arrives at Blair and is remove from the chain. The chain becomes bus chain (2) in the graph, and at the same time a new bus is generated with arrival time at Blair of 8:14. So this new bus is inserted into "pregenerated bus chain" between position 3 and position 4. Actually this new bus is inserted in position 4 and will arrive at Blair before the bus in position 5, which was generated earlier.

If we do not have this "pregenerated bus chain" and if, instead, buses left Blair in the order they were generated, then in the above example, in bus chain (3) in the graph, the bus in position 5 would leave from Blair before the bus in position 4, since it was generated earlier. But when the bus at position 4 generated, we found that it should leave

Blair before that bus …… So, the simulation is designed in this way to model the bus passing before feeder buses arrive at Blair.

Similarly for the "arrival bus chain", each bus is inserted into this chain in its order of the actual arrival time at Orleans. So the bus passing in this period is modeled as well.

4.3.6 GPS Emulator

GPS emulator is a program used to emulate the operation of the GPS system for our simulation. It works as follows:

1. Whenever a feeder bus requests real time information, the GPA emulator looks up the first feeder bus in the "arrival bus chain" (it is the next arrival) and check its actual arrival time at Orleans, assumed to be 95ActArr. Let curTime be the current time: then GPS emulator simply return the difference between 95ActArr – curTime as the expected remaining travel time to Orleans.
2. Use the expected remaining travel time as the mean to generate a gamma distributed random number and use this new random number as the actual remaining travel time to update the relevant data in the feeder bus.
3. Remove the old event from the event table and add a new event into the table.
4. re-order the arrival bus chain: this 95 bus may not be in the previous position.

GPS system normally provides information about the current location of a feeder bus. From this, controllers would need to estimate the time of arrival, i.e. the travel time from the current location to the transfer stop. But in our simulation, the time measure is more useful than distance. Assuming the average speed is always the same, the time measure and the distance measure are equivalent to each other.

4.3.7 Control Action Implementations

We have introduced the general logical description for each control action. Here we discuss how these control action are implemented in the simulation program in detail.

For "No Control", "Wait First" and "Wait Next", there is no extra implementation than what is in the control actions introduction.

For "Good Probability", if there is no real time information available (i.e. all buses which have left Blair have already arrived at Place d'Orleans), then the connecting bus will leave at once. If there is real time information, then we will check the following formula to ensure the probability of catching the feeder bus:

$$FDT + ATT * r < sdt + threshold$$

Where:

FDT: feeder bus departure time from the detector location.

ATT: average travel time from the detector location to the transfer stop.

r: is the argument to ensure a certain probability of catching the feeder bus.

Sdt and threshold have the same meaning as before.

According to this formula, if the feeder bus is expected to arrive before sdt plus threshold, then we will hold the connecting bus. Argument r is associated to a certain probability. We make a simplifying assumption here that the coefficient of variation is constant regardless of the value of ATT. We realize that normally both distance to transfer stop and time of day will affect both ATT and the variance of its distribution. Alternately, we could use the appropriate percentile point from whichever distribution is under consideration. Refer to the following example for how argument r works.

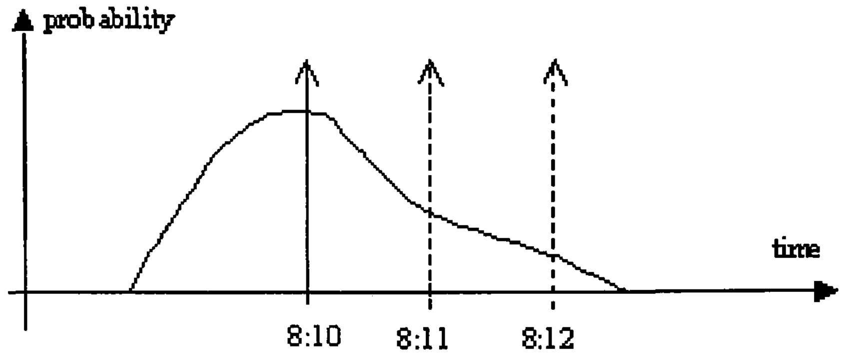


Figure 4.10 Control Action Example

In the graph figure 4.10, we assume the average travel time from the detector location to the transfer stop is 10 minutes. If this feeder bus left from the detector location at time 8:00AM, then the probability of arrival before 8:10AM at the transfer stop is 50% (assuming the average travel time is the median of the distribution). The probability of arrival before 8:11AM and 8:12AM are 90% and 97% respectively, based on the travel time distribution. Thus, if we wish to have a 97% probability to catch the feeder bus, we actually expect the feeder bus will use no more than 12 minutes in travel time, and the argument r is 12/10=1.2. Suppose the sdt is 8:11AM and the threshold is 3 minutes, then the formula discussed above becomes:

$$8{:}00 + 10 * 1.2 < 8{:}11 + 3$$

$$8{:}12 < 8{:}14$$

The conclusion is that we can hold the connecting for this feeder bus, and we have at least 97% of catching this bus.

If we just want to have a 90% probability of catching the feeder bus, the argument r is 11/10=1.1 in our example. In this case, we would hold for buses whose FDT=8:03 or earlier.

Obviously from the above analysis, if we wish to have a very large confidential level of catching the incoming feeder bus, the connecting bus is not likely to be held: we call this "conservative holding". If we choose a lower confidential level of catching the feeder bus, then the connecting bus will be held more often, but the probability would be higher that the feeder bus arrives after sdt plus threshold and that no benefit for holding.

For the "Good Benefit" control action, we first go through the same probability check as in the control action "Good Probability", and then check the expected benefit of a successful transfer by evaluating the following formula:

E[NetBenefit] = E[Ts] * E[Nf] – E[Tw] * (Nc + E[Nd])

For details, please refer to the introduction in section 3.2.6.

This function is in fact a function of the arrival time of the feeder bus, since Ts and Tw are totally dependent on the feeder bus arrival time. In the simulation program, this function is evaluated twice: first using the expected arrival time of the feeder bus, and second assuming the arrival time of the feeder bus is sdt plus threshold. Only if the sum of 2 evaluations are positive, we can hold the connecting bus.

Normally, the benefit increases first and decreases later, as the bus holding time increases. The reason to do 2-time-points checking is to ensure that not only at the feeder bus expected arrival time we can have the positive expected benefit, but also even if the incoming feeder bus is a later than expectation (but not beyond the threshold), we still have a positive expected benefit. The following graph shows the general shape of the total benefit with the feeder bus arrival time and a case where we can have the positive benefit only if the feeder arrives at our expected arrival time. If the feeder bus is a bit later than the expectation, we will have a negative benefit for the holding.

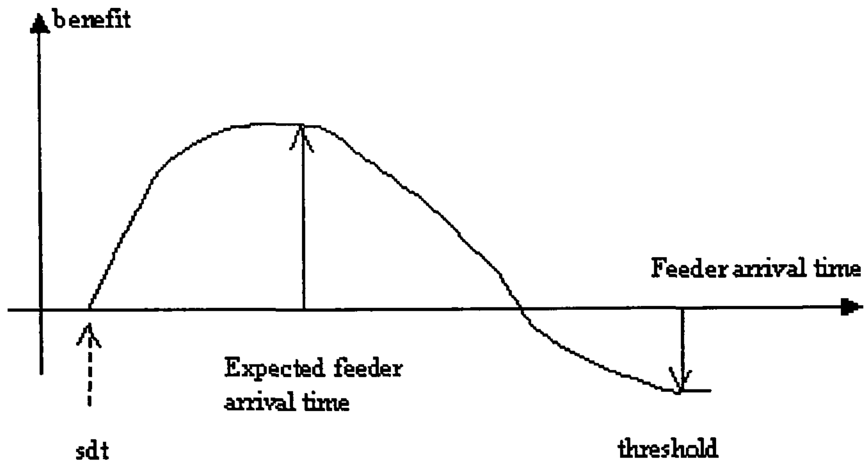


Figure 4.11 "Good Benefit" Example

Both "Good Probability GPS" and "Good Benefit GPS" are using the same decision making process as with the AVL detector. As mentioned before, the difference is only the technical means of providing real time information.

The relevant number of passengers is the actual number of passengers, i.e. we assume this information is always available to us.

4.4 Simulation Validation and Verification

Simulation program validation work was done while each piece of the program was developed. Typically, we ran the simulation program or a part of it, and we did the same work manually. If the 2 results were the same, the program passed the validation. After the whole simulation program was finished, several output text files were designed to show the simulation execution details. The primary validation and verification means is to look at these files and check with manual calculation. The validation work included:

- The sequence is correct, for example the time points are in the correct order.
- Random number generation is correct.
- The control action is correctly executed.
- Trace an event handling process.

- Walk through manually the first 20 events, and validate each step.

We will go through the following output files from a simulation run and show how to do the validation and verification work.

4.4.1 Simulation Output Files

The simulation generates some text files which can be used to validate whether the simulation program is correct or not, and to verify the details in low level for each important program entity (e.g. FeederLine class). These output files can be used to check the execution details about the simulation program, but they are not supposed to be used by normal simulation user typically. The following is a list of all output files and corresponding class or program entity.

Table 4.4 Simulation Output Files

No	**File Name**	**Relevant Class**
1	Connectingout??_??.txt	ConnectingLine
2	Feederout??_??.txt	FeederLine
3	managerout??_??.txt	TransferManager
4	stopout??_??.txt	TransferStop
5	simulationresult??_??.txt	All Classes

Note:

1) ??_?? means the current date running simulation, e.g. 3_12 for Dec, 3rd.
2) they are in the directory output/.

For each simulation run, the above files are overwritten once. So if you wish to keep these files, you need to make a copy and save them under other names. From the above naming convention, we can see that if the simulation runs are conducted in different date, then there is no need to make another copy of the output files.

There is a simulation file called simuOutput.txt reside in the same directory with the java source code. It has a record for each transfer of one simulation run, i.e. one transfer per line. Its name has no date information embedded, so whenever you wish to keep this information, you need to make another copy saved as another name.

There is an output file called siresult.txt reside in the same directory with the java source code. It has a record for each simulation run, i.e. one simulation run per line. So this file is never overwritten. The new information record is appended at the end of this file.

Connecting Line Output File

Listed in table No. 1, it is used to check the details about the connecting line relevant information in the simulation. in November, 13th, 2000, I ran simulation and the output file of the Connecting Line is named "connectingout13_11.txt". It has the following format (only a small part):

Table 4.5

arrOrleans	sdt	depOrleans	#passDownstream
1415	0	0	18
1575	1800	1869	10
3378	3600	3600	22
5009	5400	5426	27
6815	7200	7200	16
8613	9000	9000	15
10430	10800	10904	15
12365	12600	12600	27
13964	14400	14400	20
15992	16200	16200	17
17500	18000	18000	23
19435	19800	19839	16
21249	21600	21600	15
23157	23400	23430	14
24843	25200	25200	27
26525	27000	27000	16
28447	28800	28800	11
30135	30600	30600	13

Where:

ArrOrleans: is the arrival time of the connecting bus at the transfer stop - Orleans.

Sdt: is the scheduled departure time of the connecting bus from the transfer stop.

DepOrleans: is the actual departure time of the connecting bus from the transfer stop.

#passDownstream: is the number of passengers waiting along the connecting route downstream.

Please note that all times expressed are in seconds from time 0. for example, sdt=1800 means that a connecting bus is scheduled to leave 30 minutes after time 0.

Feeder Line Output File

Listed in No.2 in the table, it is used to check the details about the feeder line. For example, a file named "feederout13_11.txt" has the following format (only a small part):

Table 4.6

depBlair	arrOrleans	TransPass#	TotalPass#
240	610	5	18
490	1086	7	24
716	1308	6	23
952	1593	4	15
1241	1859	5	19
1458	2047	3	13
1669	2266	6	23
1917	2515	6	21
2158	2717	4	14
2359	2939	5	17
2620	3202	4	16
2862	3465	5	17
3123	3696	5	17
3333	3932	6	22
3599	4232	5	17
3873	4478	6	21
4085	4663	7	24

Where:

DepBlair: is the actual time when the feeder bus passes the detector, since in our case study, it is by default installed in Blair bus stop.

ArrOrleans: is the actual arrival time of feeder buses at the transfer stop.

TransPass#: is the number of transferring passengers on a feeder bus.

TotalPass#: is the total number of passengers on board of a feeder bus.

Transfer Manager Output File

Listed as No.3 in the table, it is used to check the details about the Transfer Manager in the simulation program, and actually it only records the events information. For example, a file named "managerout13_11.txt" has the following format (only a small part):

Table 4.7

event#	eventName	eventTime
1	120Departure	0
2	95DepartureBlair	240
3	95DepartureBlair	490
4	95ArrivalOrleans	610
5	95DepartureBlair	716
6	95DepartureBlair	952
7	95ArrivalOrleans	1086
8	95DepartureBlair	1241
9	95ArrivalOrleans	1308
10	95DepartureBlair	1458
11	120ArrivalOrleans	1575
12	Control	1575
13	95ArrivalOrleans	1593
14	95DepartureBlair	1669
15	Control	1800
16	95ArrivalOrleans	1859

Where:

Event#: is the sequence number assigned to each event in order of their execution (not of their generation).

EventName: is the type of an event. As discussed in chapter 4, there are 6 types of events in the simulation program.

EventTime: is the event occurrence time, i.e. at which time this event should be triggered.

Transfer Stop Output File

Listed as No. 4 in the table, it is used to check the details about the transfer stop – Orleans bus station. For example, a file named "stopout13_11.txt" has the following format (only a small part):

Table 4.8

event	eventTime	passCost	passBenefit	BET	is120Here
120Departure	0	0	0	0	FALSE
95ArrivalOrleans	610	5	0	0	FALSE
95ArrivalOrleans	1086	12	0	0	FALSE
95ArrivalOrleans	1308	18	0	0	FALSE
120ArrivalOrleans	1575	18	0	1611	TRUE
95ArrivalOrleans	1593	22	0	1619	TRUE
95ArrivalOrleans	1859	22	5	1869	TRUE
120Departure	1869	22	5	1869	TRUE
95ArrivalOrleans	2047	3	0	0	FALSE
95ArrivalOrleans	2266	9	0	0	FALSE
95ArrivalOrleans	2515	15	0	0	FALSE
95ArrivalOrleans	2717	19	0	0	FALSE
95ArrivalOrleans	2939	24	0	0	FALSE
95ArrivalOrleans	3202	28	0	0	FALSE
120ArrivalOrleans	3378	28	0	3434	TRUE
95ArrivalOrleans	3465	33	0	3475	TRUE
120Departure	3600	33	0	3475	TRUE
95ArrivalOrleans	3696	5	0	0	FALSE
95ArrivalOrleans	3932	11	0	0	FALSE

Where:

Event: is the event name. One line or one record is the relevant information about the transfer stop when this event is triggered.

EventTime: is the event occurrence time.

PassCost: is the number of passengers waiting for the connecting bus. It is not same as the PassCost in the final simulation result data table.

PassBenefit: is the number of passengers who got benefit from the control action.

BET: is for Boarding End Time, which is the current known passenger boarding finish time.

Is120Here: is a Boolean value – when this event occurs, if there is a connecting bus at the transfer stop, it is TRUE, otherwise, it is FALSE.

General Simulation Output File

Listed as No. 5 in the table, it is used to check general execution flow of the simulation program. It records a lot of details about the feeder bus, connecting bus, transfer stop, the control action and so on. To use this file, you need to have some knowledge about the transfer and simulation, and to read it carefully. For example, a file named "simulationresult13_11.txt" has the following format:

```
==Simulation Begin==

Arguments:
 Total Simulation Time = 600000
 Avg Travel Time of 95 = 600
 Avg Travel Time of 120= 1440
 headway of 95     = 240
 headway of 120    = 1800
 uniform r.n.g seed1  = 123456789
 uniform r.n.g seed2  = 987654321
 controlActionSelection= 3
 threshold          = 180
 avgBoradingTime     = 2

 new FeederLine
  new 95bus with: dep@490, arr@1086, with pass 24
  new 95bus with: dep@716, arr@1308, with pass 23
  new 95bus with: dep@952, arr@1593, with pass 15
  new 95bus with: dep@1241, arr@1859, with pass 19
 new ConnectingLine.
 new EventChain
```

```
(1) Event: 120Departure at 0
 time = 0 sdt  = 0
 passOn120 = 20
 TranStop: handle event 120Departure,
TransferStop : bET=0, sdt=1800
120Here=false, passCost=0, passBene=0

 ConnLine: handle event 120Departure,
  time = 0
  sdt  = 0
next120arr=1575, next120sdt=1800
 -> event 120ArrivalOrleans, at 1575

(2) Event: 95DepartureBlair at 240
  new 95bus with: dep@1458, arr@2047, with pass 13
! 1st in arrChain.

 -> event 95DepartureBlair, at 490

(3) Event: 95DepartureBlair at 490
  new 95bus with: dep@1669, arr@2266, with pass 23
 -> event 95DepartureBlair, at 716

......

(10) Event: 95DepartureBlair at 1458
  new 95bus with: dep@2620, arr@3202, with pass 16
 -> event 95DepartureBlair, at 1669

(11) Event: 120ArrivalOrleans at 1575
 ConnLine: handle event 120ArrOrleans,
  arr = 1575
  sdt = 1800
current=1575, sdt=1800
 TranStop: handle evet 120ArrivalOrleans,
TransferStop : bET=1611, sdt=1800
```

Here only shows a few event handling processes, and for this simulation run there are totally 6746 event handlings. Basically, the above example shows the following information:

(1). The simulation arguments for this run.

(2). The pregenerated feeder buses before the simulation starts to run.

(3). The details for each event handling in simulation.

Simulation Evaluation Output File

This file is named simuOutput.txt, and it has the following format (only a small part):

Table 4.9 Output for Each Transfer Simulated

counter	arr120	sdt	holdingTime	delaytime	dep120	controlsucceed	passcost	passbenefit	pass120down	passzero	cost	benefit	netbenefit	totalbenefit
1	1415	0	0	0	0	FALSE	0	0	18	0	0	0	0	0
2	1575	1800	69	69	1869	TRUE	22	5	10	0	2208	8655	6447	6447
3	3378	3600	0	0	3600	FALSE	0	0	22	33	0	0	0	6447
4	5009	5400	26	26	5426	TRUE	38	3	27	0	1690	5322	3632	10079
5	6815	7200	0	0	7200	FALSE	0	0	16	34	0	0	0	10079
6	8613	9000	0	0	9000	FALSE	0	0	15	33	0	0	0	10079
7	10430	10800	104	104	10904	TRUE	30	5	15	0	4680	8480	3800	13879
8	12365	12600	0	0	12600	FALSE	0	0	27	31	0	0	0	13879
9	13964	14400	0	0	14400	FALSE	0	0	20	39	0	0	0	13879
10	15992	16200	0	0	16200	FALSE	0	0	17	36	0	0	0	13879
11	17500	18000	0	0	18000	FALSE	0	0	23	32	0	0	0	13879
12	19435	19800	39	39	19839	TRUE	37	3	16	0	2067	5283	3216	17095
13	21249	21600	0	0	21600	FALSE	0	0	15	30	0	0	0	17095
14	23157	23400	30	30	23430	TRUE	32	4	14	0	1380	7080	5700	22795

Simulation Final Result File

This file is named siresult.txt, and it has the following format (only a small part):

Table 4.10 Simulation Parameters

SimuTime	DecLoca	Trip120	HW95	HW120	Control	Thresh	avgBT
600000	600	1440	240	1800	No_Control	180	2
600000	600	1440	240	1800	Good_Probability	180	2

Table 4.11 Simulation General Result

TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
0	0	0	0	0	0	0	0	0	0
359526	3984	5225	0.9722222	0.9166667	8932	332	9005	24.646216	1745.9819

Actually, the above is one table. Here it is split into 2 just for representation. The first line is for "No Control" and all the measurements are zero by definition. The second line is for "Good Probability" with the default set of the arguments.

4.4.2 Validation and Verification

In this section, we will introduce how the simulation validation was done in general and how the simulation program user can use these methods to validate and verify the simulation execution. However, not all of the validation work we did during the simulation program development is listed here. In fact, these output data files are not separated with each other. They have close relationships to each other. For example, if we find an event in transfer manager output file (e.g. event#=2, eventName=95DepartureBlair eventTime=240), then we should find the same event in feeder line output file and transfer stop output file. The content in each file should have a correct logic relationship. This is a more complex validation work and we did some of this correlation validation work during the simulation development as well.

4.4.2.1 Validating the General Correctness

This includes verifying that the time sequence is in the correct order, the number of passengers is never beyond the limit (as a uniform distribution), the total delay time is never less than the total holding time by definition and so on. For the time order issue, the correct order should be that each number in a line should be greater than the number in the previous line above. For the number of passengers issue, since they are uniform distribution on [a, b], the correct values should never be outside interval [a, b].

First, check the connecting line output file with arrOrleans, sdt and depOrleans columns for time order issue. Check the #passDownstream column for number of passengers issue.

Second, check the feeder line output file for depBlair and arrOrleans for time order issue. Check TransPass# and TotalPass# for number of passengers issue.

Third, check the transfer manager output file with eventTime for time order issue.

Fourth, check the transfer stop output file with eventTime and BET columns for time order issue.

Fifth, check the simulation evaluation output file and simulation final results file for the holding time and delay time issue.

4.4.2.2 Validating Random Number Generation

There are 2 kinds of random number generations in the simulation: the travel time of Gamma distribution and the number of passengers modeled as uniform distribution.

First check the connecting line output file with the arrOrleans and depOrleans. The difference between the value of column arrOrleans and the previous depOrleans is the actual connecting bus trip travel time, which should be Gamma random number with mean of 24. Using MS Excel software, this work was easily done, and we have the following graph.

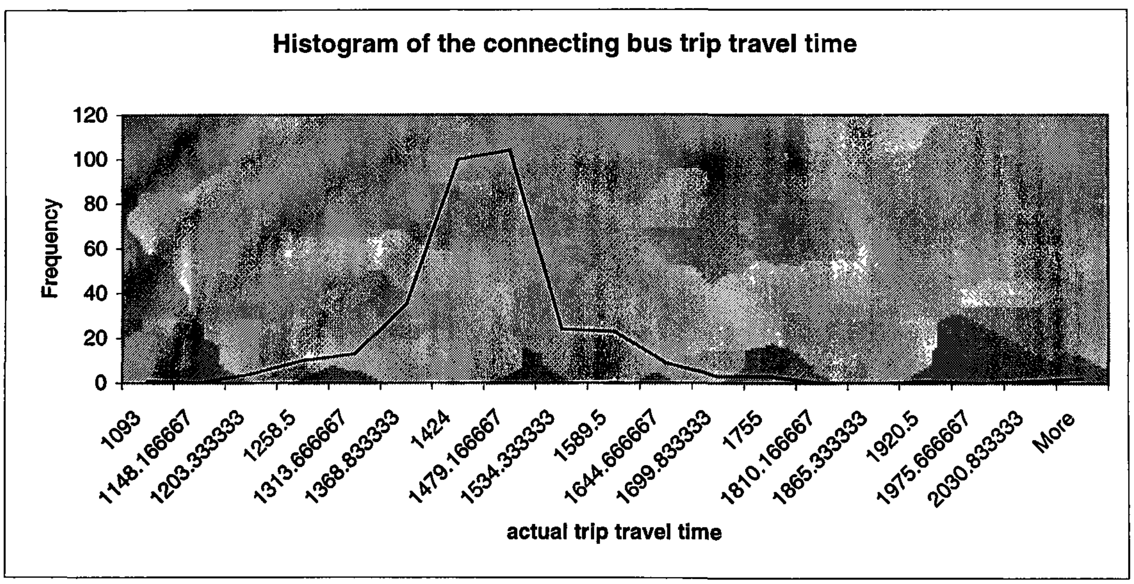


Figure 4.12 Validation of Gamma Random Numbers

We can see that it basically looks like Gamma distribution centered around 1440 seconds (24 minutes). Actually the following is the descriptive statistics.

Table 4.12

Mean	1433.282282
Standard Error	6.054090239
Median	1427
Mode	1435
Standard Deviation	110.4767798
Sample Variance	12205.11887
Minimum	1093
Maximum	2086
Sum	477283
Count	333

The same validation experiments can be done against any applicable output data, and has been applied to all of the simulation random number generation.

4.4.2.3 Validating the Control Action

In each control action, there is a large amount of data to manipulate to make the final decision to hold the bus or not. We need to validate that this control action is done correctly. For the correct control action execution, please refer to the chapter 3 control action introduction and the chapter 4 - event handling relevant introductions.

Generally this validation work needs to check several output files. First check the general simulation output file and find the event Control. In our above example, it is the following text:

(12) Event: Control at 1575
TM: Wait Next with Good Prob
sdt=1800, bet=1611, current=1575
->event Control at 1800

(15) Event: Control at 1800
TM: Wait Next with Good Prob

```
sdt=1800, bet=1619, current=1800
check prob, expectedArrival=1961.0, sdt+sh=1980
hold and succeed, next act arrival=1859, sdt+sh=1980
->event GoASAP at 1860

 (17) Event: GoASAP at 1860
   time = 1860   bet  = 1869
   -> event : GoASAP at time: 1869

 (18) Event: GoASAP at 1869
   time = 1869   bet  = 1869
   -> event : 120Departure at time: 1869

 (19) Event: 120Departure at 1869
  time = 1869 sdt  = 1800
  passOn120 = 12
  TranStop: handle event 120Departure,
TransferStop : bET=0, sdt=3600
120Here=false, passCost=0, passBene=0

  ConnLine: handle event 120Departure,
   time = 1869
   sdt  = 1800
next120arr=3378, next120sdt=3600
  -> event 120ArrivalOrleans, at 3378
```

We ignore some other event not related to this issue. Basically, event 12 means the current time is 1575, the control action is "Good Probability", sdt is 1800, BET is 1619, we can not make decision because it is not the bus ready time yet (earlier than sdt). The result of this event handling is to generate a new event Control at time 1800. Event 15 means, at time 1800 we apply control action again. The probability check has a positive result, because the sdt plus threshold is 1980, and the expected arrival of the feeder bus with 97% probability is no later than 1961. The connecting bus is held and makes the transfer succeed. The actual arrival of the feeder bus is 1859, so the new generated event is the GoASAP at time 1859 (1859+1 seconds is used for programming reason and makes no difference in simulation results). Event 17 is dealing with the GoASAP, because the passengers boarding has not finished, the new event GoASAP at time BET of 1869 is generated. At last, event 19 makes the connecting bus really leave the transfer stop and start the next trip at time 1869.

4.4.2.4 Other Validation Work

As mentioned, we can also trace an event handling process for each type of the 6 events. This will need to check the correlation between several output files and general procedure is quite similar with 4.4.2.3. For example, we can pick up an event 95ArrivalOrleans at time 610 from the feeder line output file, and check the following:

- If the event is showing up in the transfer manager output file correctly.
- If the event is showing up in the transfer stop output file correctly, including verifying if the number of waiting passengers increases by the number of transfer passengers on this feeder bus.
- Check the general simulation output file for the event handling in feeder line and transfer stop.
- It should have nothing to do with the connecting line.

Another important validation work is to walk through some event handling process step by step. This involves really large amount of the manual work to do and we only introduce the important steps here. First, using transfer manager output file as an event index and start from the first event. Check the event handling introduction to find the relevant classes and the corresponding work (including the state modification and the new event generation) that should be done, then go to each relevant output file to verify if it is done correctly. After check the whole event processing, go back to the general simulation output file and verify the new event, and start checking the next event.

The verification work is following the same approach but looking for some simulation execution details instead of the programming errors.

After extensive validation and verification effort, we can safely say that the simulation model and program is validated and reliable.

Chapter 5 Simulation Experiments and Results

This chapter will present the simulation experiments that have been performed, and how to interpret the simulation results. Some simulation scenarios are conducted and the results are analyzed to draw important conclusions.

5.1 General User Guide

5.1.1 Graphic User Interface

The simulation program provides a graphic user interface so as to make it convenient for the user to input specific parameters prior to running the simulation. As shown below, in the Transfer Simulation window, user can specify 4 sets of data to be used in the simulation, categorized into 2 blocks.

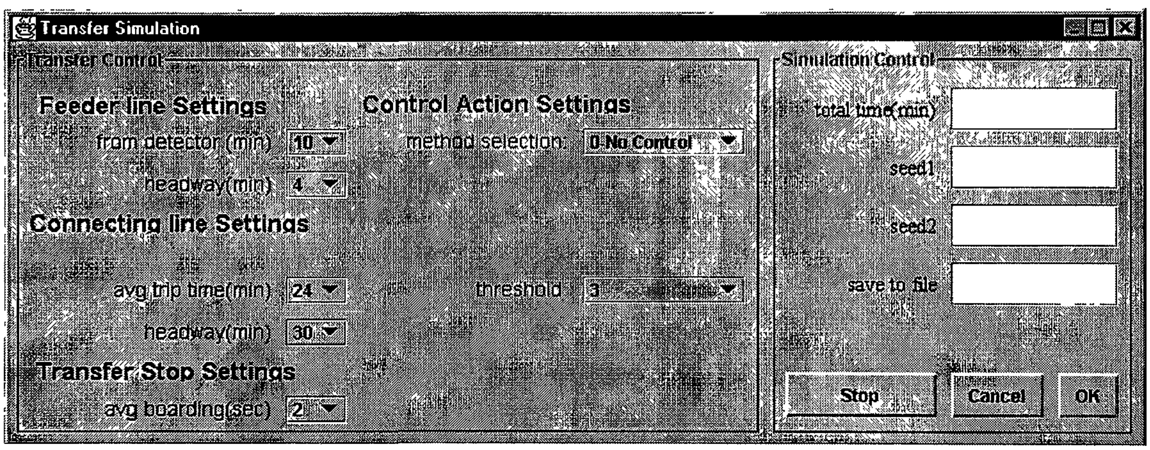


Figure 5.1

1. **Transfer Control**

   This block of data is going to be used for the transfer. They are classified into 3 sets.

   1) Feeder Line Settings

      - *from detector (min)* : to choose the average travel time from the location of the AVL detector to the transfer stop in minutes. If the average traveling speed is fixed, then it is equivalent to the distance in kilometers. The available numbers in the menu are 2, 4, 6, 8, 10, 12 or 14 minutes.

- *Headway (min)* : the headway of the feeder line in minutes. The available numbers are 4, 6 or 8 minutes.

2) Connecting Line Settings
   - *Avg trip travel time(min)* : the average travel time for each trip of the connecting bus, i.e. the time used to complete a round trip starting from the transfer stop and back to the transfer stop in minutes. The available numbers in menu are: 24, 26 or 30 minutes.
   - *Headway (min)* : the headway of the connecting line in minutes. The available numbers are 20, 26, 28, 30, 32, 34 or 35 minutes.
   - *Threshold (min):* the maximum holding for the connecting bus in minutes. The values available in the menu are 1, 2, 3, 4, 5, and 6 minutes.

3) Transfer Stop Settings
   - *Avg boarding time (sec)* : the average boarding time for each passenger in seconds. We assume that every passenger will use the same time to board on the connecting bus from the transfer stop. The values available in the menu are: 1, 2, 3, 4, or 6 seconds.

4) Control Action Settings
   - *Method selection* : the control action that is going to be used to dispatch the connecting bus. We will use the same control action for each simulation run. The available actions are:

     0- No Control

     1- Wait First

     2- Wait Next

3- Good Probability

4- Good Benefit

5- GPS, Probability

6- GPS, Benefit

2. Simulation Control

This block is used to define the simulation related parameters to control the simulation running.

1) Total time (min) : the total simulation time in minutes, i.e. the total operation time of the bus transfer system. This is an integer number.
2) Seed1 : used by the random number generator.
3) Seed2 : used by the random number generator. The Gamma random number generator will need 2 seeds. For the same seeds, random number generator will generate exactly the same sequence of the random number. For other random generation, such as Uniform numbers, they are using a seed calculated from the above 2 seeds as well. This will ensure that our simulation experiments are based on the same set of random data.
4) Save to file : user can choose a file name to store the simulation output result data, the program will provide the default value for this if user does not provide one. But it is primarily for the simulation designer to verify and validate the simulation program and data results. To use the Excel template, leave this blank: since the Excel template can only search for a fixed filename.

5.1.2 General User Guide

To run this simulation program, you need a java-enabled operating system and almost all the popular OS are supporting java. You also need JVM (java virtual machine) installed to run this, and it requires JDK1.2 and above versions because of the java Swing application. Either using a SDK tool or JDK, just run "*java TransferSimulator*", then the simulation GUI (graphic user interface) will pop up for the user to specify the simulation parameters.

After the simulation arguments including the transfer arguments are selected from the GUI, click the OK button and the program will start running the simulation. We can see a black DOS window at the same time normally showing the program progressing messages.

We provide some ready-to-use Excel templates for customized simulation scenarios. However, if you want to design your own simulation plan, you can analyse the data found in the simulation data output tables.

The Excel template can be used to build statistics tables and graphs summarizing the output results, and allowing for comparison of different scenarios. To use the Excel template, follow the steps below:

(1). Select the simulation parameter in the GUI, and do not change them during the experiment. Then select the control actions one by one: highlight the control "0-No Control" and click OK button, when the simulation finishes, highlight the second control "1- Wait First" and click OK button, and so on. When you complete the 6 control actions, run MSOffice Excel and open an Excel template named "simulation_analysis.xlt". We will get the following screen. Click the button titled "Controls", the data output table and graphs are generated automatically by Excel template. Note here there should be totally 6 simulation runs, corresponding to 6 control actions.

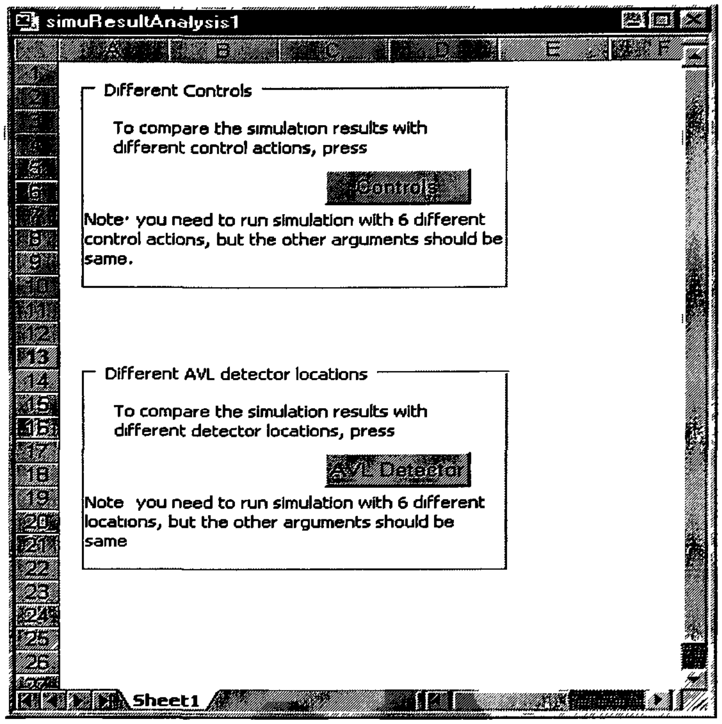


Figure 5.2

(2). Select all the simulation arguments except the "from detector (min)". Note here if you select a control action that does not use the detector, then the output will be undefined and does not make any sense. For "from detector (min)", you select one value from the menu and run the simulation by clicking the OK button, and you select another one and click the OK button and so on. There are should be totally 6 simulation runs. After all this is done, open Excel template "simulation_analysis.xlt", click on the button titled "AVL Detector" and the data output table and graphs are generated automatically by the Excel template.

5.2 Simulation Data Output Template

In the following discussion, we will show many data tables sharing the same format. Here we give the definitions about each data column, there are 8 measurements resulting from the simulation.

Table 5.1 Simulation Output Data Table Header

TotalBenefit:	The accumulated total benefit over all transfers being simulated.
TotalHolding:	The accumulated total holding time over all transfers being simulated.
TotalDelay:	The accumulated total delay time of the connecting bus for whatever reasons, over all connecting buses simulated.
SuccessRate "s_Rate"	Ratio of s/t, where s is the number of times that we held the connecting bus and the transfer was successful, t is the total number of times that we held the connecting bus. We say a transfer is successful if the connecting bus was held from its ready time and a feeder bus arrived between the connecting bus ready time and sdt + threshold.
GoodControlRate "g/s_Rate"	Ratio of g/t, where g is the number of times that we held the connecting bus, the transfer was successful and the benefit measure is positive, t is the total number of times that we held the connecting bus.
PassCost:	The number of passengers who incur extra "cost" due to the control. Cost is actually the waiting time here. These passengers are those who arrived before sdt and the passengers waiting downstream along the connecting line.
PassBenefit:	The number of passengers who get "benefit" due to the control. Benefit is actually the time-saving here. These passengers are those who arrived after sdt but before the actual departure time of the connecting bus, and would not have caught the connecting bus without the control action.
PassZero	The number of passengers who did not get cost neither benefit. This is because if there is no bus holding, we define the cost/benefit measure is zero. If the result of the control is "no holding", this number represents the number of passengers whose benefit and cost equals 0 (there is no benefit or cost if there is no holding). This include all passengers on board and downstream at he connecting bus ready time.
Avg Cost	The average cost in terms of the total cost divided by the total number of passengers who paid extra cost due to control.
Avg Benefit	The average benefit in terms of the total benefit divided by the number of passengers who benefited from the control.

Additionally, we want to clarify 2 concepts.

Holding time: is the extra holding time of the connecting bus exclusively due to control action.

Delay time: is the difference between the actual departure time of the connecting bus and the scheduled departure time.

Generally speaking, the greater the total benefit, the more effective the control action. As with total holding time, total delay time is a surrogate of the tradeoff of the control action. Note that if a bus is held, it is more likely to begin the next trip late, then having a positive delay time.

We have a set of default simulation arguments. In the following experiments, we will use these values unless otherwise specified:

(1). Route 120:

- headway = 30 minutes
- trip travel time = 24 minutes
- threshold = 3 minutes

(2). Route 95:

- headway = 4 minutes
- travel time from detector = 10 minutes

(3). Transfer stop:

- average boarding time = 2 seconds

(4). Control actions: different each run. A total of 7 runs is done with the following control actions:

- No Control
- Wait First
- Wait Next
- Good Probability

- Good Benefit
- GPS, Good Probability
- GPS, Good Benefit

(5). Total Simulation Time = 10,000 minutes, no seeds or saved files are specified, as the default values are used.

5.3 Control Action Comparison

5.3.1 Introduction

In this section, we will use "total benefit" to measure the performance of several control actions. The total benefit is calculated from the objective function originally defined in chapter 3:

$$
\begin{aligned}
\text{NetBenefit} &= \text{benefit} - \text{cost} \\
&= Nf * Ts - Nc * Tw \\
&= Nf * (HW - Td) - (Nb + Nd) * Tw
\end{aligned}
$$

From the objective function, we can see that when HW (the connecting line headway) increases, the total benefit will increase. The more transfer passengers on feeder line and less passengers on connecting line, the more benefit. We will examine these conclusions later.

5.3.2 Simulation Runs

The first experiment uses the default parameters for each of the six control actions. The simulation arguments table, all are in seconds, follows:

SimuTime	DetecLoca	Trip120	HW95	HW120	Thresh	avgBT
600000	600	1440	240	1800	180	2

The simulation output data table, all time are in seconds, follows:

Control	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
Wait_First	-355945	41891	43132	0.6967	0.489489	17159	1110	0	130.4638	1696.1117
Wait_Next	-355945	41891	43132	0.6967	0.489489	17159	1110	0	130.4638	1696.1117
Good_Probability	359526	3984	5225	0.97222	0.916667	8932	332	9005	24.64622	1745.9819
Good_Benefit	344284	3412	4653	0.96875	0.9375	8675	304	9290	21.56726	1747.9605
Good_Prob_GPS	616847	17852	19093	0.965	0.77	12878	929	4462	75.46397	1710.0883
Good_Benefit_GPS	633751	13702	14943	0.95092	0.877301	11752	804	5713	63.31416	1713.7052

5.3.3 Discussion

5.3.3.1 Total Benefit Measurement

PassCost, passBenefit and passZero

If we look at the number of passengers issue, comparing the control action "Good Probability" and "Good Probability GPS", the total benefit increases by about 72%. The number of passengers with cost increases 44% and the number of passengers with benefit increases by 180%. The conclusion is that, with the help of GPS system, the connecting bus is held more than AVL detector case, and we have 44% more passengers paid extra cost, but we have 180% more passengers getting benefit from the control action. This suggests that the more accurate real time information obtained from GPS allows for improved transfer effectiveness, which is what was expected.

Holding time and delay time

The connecting bus is held more in GPS case than under the detector case. This is due to the fact that GPS can provide the latest information about the feeder bus location so that we can make a decision with less uncertainty. In the AVL detector case, many times we are not sure if the feeder bus will arrive before the threshold with a certain probability, so we can not hold the connecting bus with this risk and we lost some potential control benefit.

Let's look at the total benefit measurement first.

Control Action	TotalBenefit
Wait_First	-355945
Wait_Next	-355945
Good_Probability	359526
Good_Benefit	344284
Good_Prob_GPS	616847
Good_Benefit_GPS	633751

From the simulation results suggests 3 classes: "Good Benefit GPS" and "Good Probability GPS" are the best. "Good Probability" and "Good Benefit" are in the middle. "Wait First" and "Wait Next" are the worst.

1. "Good Benefit GPS" and "Good Probability GPS" are the best.

Control	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
Good_Prob_GPS	616847	17852	19093	0.965	0.77	12878	929	4462	75.46397	1710.0883
Good_Benefit_GPS	633751	13702	14943	0.95092	0.877301	11752	804	5713	63.31416	1713.7052

This result reflects the basic fact: GPS can provide better real time information service than AVL detector, and real time monitoring techniques can help to improve the dynamic bus dispatching greatly. From the data table, we see that control actions "Good Benefit GPS" and "Good Probability GPS" almost have double benefit of that of control actions "Good Probability" and "Good Benefit".

They are expected to be the best with the help of the advanced and expensive GPS technology. There is no significant difference between "Good Probability" and "Good Benefit" in total benefit, no matter what monitoring technique is used. It is because that the extra work for "Good Benefit" compared with "Good Probability" is to check the expected benefit. The expected benefit is primarily determined by the number of passengers, which is a uniform distributed random number. Only if there is very much more number of passengers incur cost than the number of passengers get benefit, the benefit check will play an important role.

However, the holding time and delay time are significantly improved due to the benefit check. Less passengers are involved in the transfer control (passZero increases by 1251), 1126 passengers are no longer involved in "extra waiting" (passCost) with 125 are no longer involved in "time saving" (passBenefit).

Also, let us compare the average cost and benefit. For "Good Benefit GPS", there are 804 passengers reduced their waiting by 1713/60=28.5 minutes on average. There are 11752 passengers incurring extra holding time of 63/60=1.05 minutes on average. The total benefit is 633751 (person * second) = 10562 (person * minute) during 10,000 minutes simulation. Since there are a total of 804 passengers who got benefit, the average net benefit per passenger is 10562/804=13.1 minutes. For "Good Probability GPS", 929 passengers reduced their waiting time by 1710/60=28.5 minutes on average, and there are 12878 passengers who incurred extra waiting time of 75.5/60 = 1.26 minutes on average. The total benefit is 616847 (person * second) = 10280 (person * minute) during 10,000 minutes simulation. The average net benefit per passenger is 10280 / 929 = 11.1 minutes.

2. "Good Probability" and "Good Benefit" are in the middle.

"Good Probability" and "Good Benefit" are better than "No Control". They are not the best, but they are more realistic and feasible and can improve the service. The big difference with the "Good Probability GPS" and "Good Benefit GPS" is due to the monitoring technology.

Control	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
Good_Probability	359526	3984	5225	0.97222	0.916667	8932	332	9005	24.64622	1745.9819
Good_Benefit	344284	3412	4653	0.96875	0.9375	8675	304	9290	21.56726	1747.9605

Compared with GPS based control actions, these 2 controls had lower avg_cost and a little higher avg_benefit, but very much less total benefit. The main problem is the connecting bus is held less which can be seen by checking the holding time in the table. It is because the AVL detector provides real time information which is not as accurate or reliable as that of the GPS based technology.

The total benefit for "Good Probability" is 359526 (person * second) = 5992 (person * minute), and the average net benefit is 5992/332 = 18 minutes which is even higher than that of GPS based controls. However, fewer passengers benefit form these control actions.

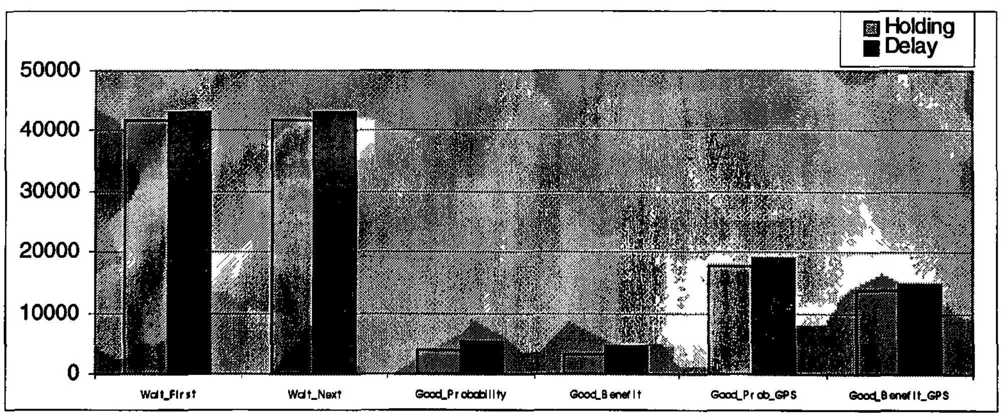
We should point out that, we used a very high confidence level (about 97%) for the control actions, and we call this "conservative holding" control actions.

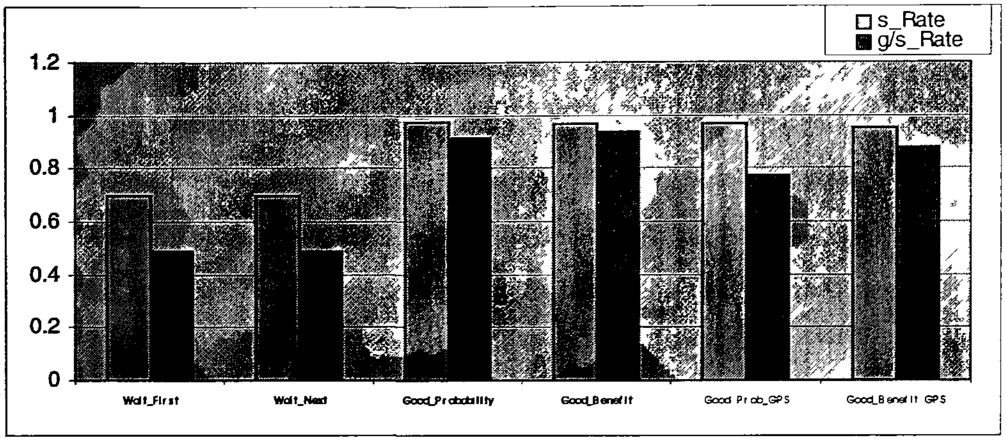
3. "Wait First" and "Wait Next" are the worst.

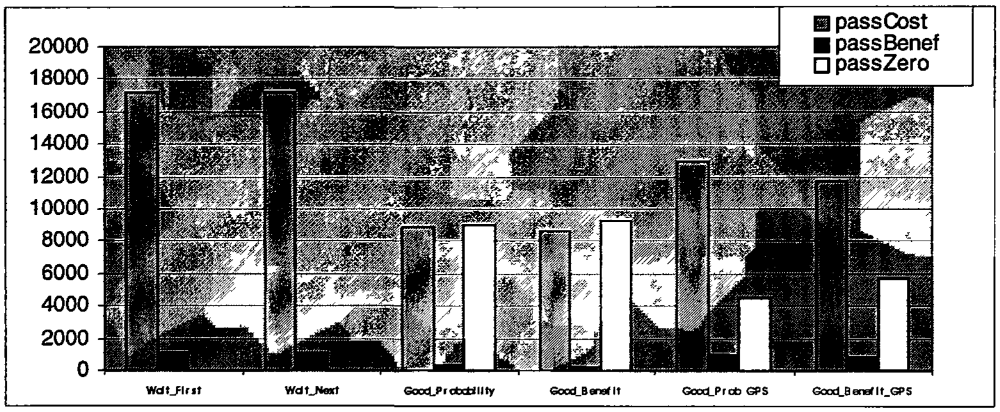
They are worse than "No Control", because they represent a strategy that the connecting bus is almost always held without arguing the potential benefit or cost, which is the result of lacking necessary real time information. They can be used to test against the bus route schedule actually, but this is not our research scope and we only use these as for the sake of comparison with actions making use of real time information.

Control	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
Wait_First	-355945	41891	43132	0.6967	0.489489	17159	1110	0	130 4638	1696.1117
Wait_Next	-355945	41891	43132	0.6967	0 489489	17159	1110	0	130.4638	1696.1117

Actually, if the trip travel time of the connecting bus is larger relative to the headway, the "Wait First" will be better than the "Wait Next". It is because "Wait First" checks some historical information or the arrival of a feeder between time sdt and the ready time, to determine whether to hold or. "Wait Next" is a pure static control in the sense that it has no information about the current situation at all when the connecting bus is held. We do not recommend this type of "always holding" strategies to be implemented for real time bus dispatching.




s_Rate
g/s_Rate
1.2
1
0.8
0.6
0.4
0.2
0
Wait_First
Wait_Next
Good_Probability
Good_Benefit
Good_Prob_GPS
Good_Benefit_GPS


passCost
passBenef
passZero
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Wait_First
Wait_Next
Good_Probability
Good_Benefit
Good_Prob_GPS
Good_Benefit_GPS

5.4 The Optimized AVL Detector Location

5.4.1 Introduction

This experiment is to find out where is the best location to install the AVL detector to achieve the maximum control benefit. The current and default location in the simulation is in the Blair bus stop, which is upstream of the transfer stop at Place de'Orleans along route 95. The average travel time from the detector location to the transfer stop is about 10 minutes.

Because the control actions "No Control", "Wait First" and "Wait Next" do not use the real time information provided by the detector at all, we do not study these control actions this time. The control actions using GPS are not considered here either, since they are not relevant with AVL detector. The candidates for this scenario are "Good Probability" and "Good Benefit".

In our discussion, the detector location is defined in terms of a certain time away from the transfer stop. The implicit assumption is that the feeder line has the same statistical average speed, so that the distance can map into an exact time measure.

Generally, the detector location mainly determines the quantity and quality of the real time information and will have an important impact on any real time information application. There are 3 aspects to consider about this issue. First is the information availability. If the AVL detector is set too close to the transfer stop, it may be that at the connecting bus ready time, no feeder bus has yet crossed the detector so that no information is available about the expected arrival of the feeder bus. No holding could take place due to lack of information. Second is the real time information accuracy, i.e. we want to reduce uncertainty and we want an decision to be made based on the latest information. This requires that the detector should be installed as close as possible to the transfer stop. Third is the real time information application requirement. This is application specific. For example in our real time transfer dispatching, we can install the detector in the place from where the average traveling time to the transfer stop is one threshold. Then the control action can be generally simplified to: if and only if there is a feeder bus passed the detector, we will hold the connecting bus within the threshold.

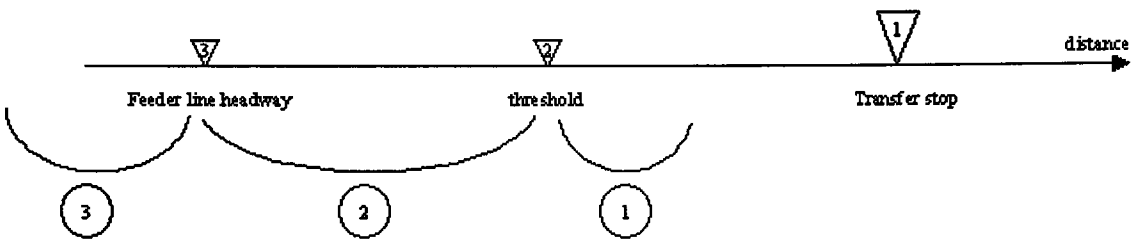


Figure 5.3

To illustrate our discussion, let us look at an example here. In the figure 5.3, triangle 1 is the transfer stop, triangle 2 is the place from where the average traveling time is equal to one threshold and triangle 3 is the place from where the average traveling time is equal to the headway of feeder line. Circle 1, 2 and 3 are the area where we can install the detector. From the discussion in the above paragraph, we have the following table showing the general advantages and disadvantages for each option.

Table 5.2 Detector Location General Comparison

	Very Far	Area (3)	Area (2)	Area (1)	Very Close
Information Availability	Always	Usually	Often	Sometimes	Rarely
Information Accuracy	Very Bad	Bad	Normal	Good	Very Good

As indicated above, the most relevant factors with the optimized detector location we propose would be the threshold and the feeder line headway. The following figure is an example about how the detector location can affect the total control benefit. Suppose the sdt of the connecting bus is 8:10 and the threshold is 1 minute. In case 1, the detector is installed 8 minutes away and in case 2 it is 4 minutes away from the transfer stop. We use the control action "Good Probability" to dispatch the connecting bus. When the connecting bus is ready it checks the latest departed feeder bus from the detector location about its departure time. To have a high probability to catch this feeder bus, the control action simply check whether it will arrive before sdt plus threshold be the following:

FeederDepartureTime + AverageTravelTime * r < sdt + threshold.

Where: FeederDepartureTime is the feeder bus departure time from the detector.

AverageTravelTime is the average traveling time from the detector to the transfer stop.

r is used to ensure certain probability of catching this feeder bus, e.g. if the average travel time is 10 minutes, then the probability of the feeder bus will arrive before 12 minutes is 97%.

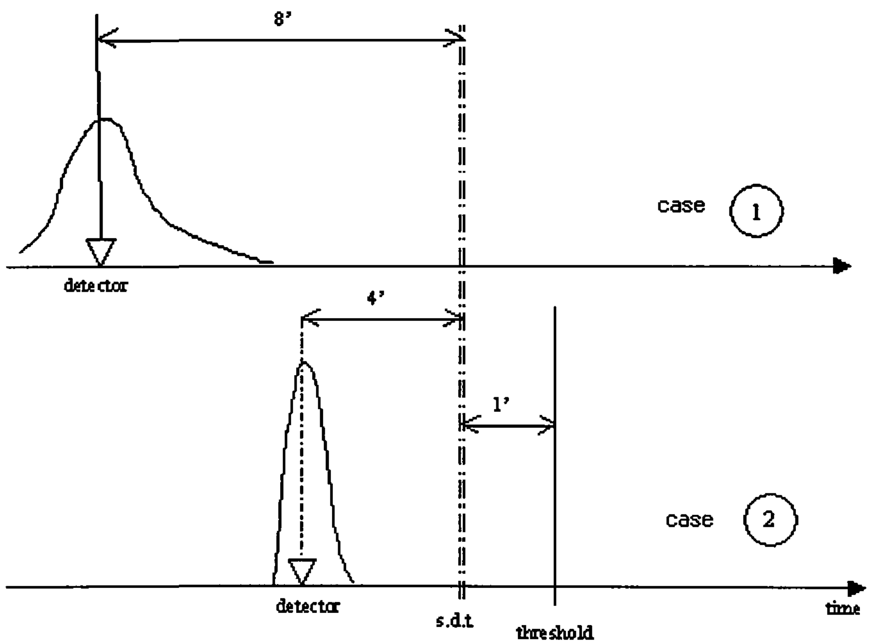


Figure 5.4

Case (1)

The feeder bus passes the detector at time 8:02, so with 97% probability, we expect the arrival time at the transfer stop is about 8:02 + 8 * 1.2 = 8:11:36. Due to this expectation, we will not hold the connecting bus.

Case (2)

Assuming the feeder bus has the same schedule as in case (1) and assuming its travel time from detector location 1 to detector location 2 is equal to the expected travel time, i.e. 4 minutes, its passing time to the detector #2 should be 8:06 with 97% probability, so the expected arrival time at the transfer stop is 8:06 + 4 * 1.2 = 8:10:48. So we will hold the connecting bus.

Having a detector closer to the transfer stop means the variance of the travel time between the detector and the transfer stop is smaller which means decisions should be more effective as they are based on less uncertainty.

5.4.2 Simulation Runs

Here is the common simulation arguments table, and time is in seconds.

SimuTime	Trip120	HW95	HW120	Control	avgBT
600000	1440	240	1800	Good_Probability	2

The followings are simulation results for different thresholds. Each of them is the total benefit with different detector locations (in terms of average traveling time).

Threshold equals to 1 minute.

Detctor	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	401963	767	2498	0.981481	0.98148	8389	255	9625	6.3400882	1784.902
2	419577	1125	2654	0.983051	0.98305	8550	276	9443	8.3870175	1780.0217
3	387611	1218	2634	0.964912	0.96491	8483	261	9525	8.9915124	1777.341
4	221664	798	2158	0.944444	0.94444	7805	156	10308	7.1183857	1777.0705
5	108112	563	1888	0.857143	0.85714	7284	85	10900	5.8725975	1775.1529
6	24805	101	1387	1	1	6724	17	11528	0.7972933	1774.4706
7	8885	97	1385	1	1	6660	8	11601	0.7732733	1754.375
8	12010	143	1401	0.666667	0.66667	6690	11	11568	1.0986547	1760
9	10364	183	1427	0.5	0.5	6715	11	11543	1.3518987	1767.4545
10	8902	213	1454	0.5	0.5	6715	11	11543	1.5448995	1752.3636
12	33349	261	1486	0.714286	0.71429	6797	26	11446	1.8712667	1771.8462
14	20404	268	1506	0.666667	0.66667	6757	19	11493	1.9336984	1761.5789

Threshold equals to 2 minutes

Detector	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	401963	767	2498	0.9814815	0.9814815	8389	255	9625	6.3400882	1784.902
2	429599	1330	2859	0.9836066	0.9836066	8622	288	9359	9.5171654	1776.5833
3	438120	1659	3075	0.9545455	0.9545455	8786	303	9180	11.272934	1772.8185
4	427768	1973	3333	0.9552239	0.9402985	8818	307	9144	13.069403	1768.7752
5	409940	2150	3474	0.9402985	0.9253731	8809	303	9157	14.199001	1765.7393
6	388086	2112	3398	0.9677419	0.9677419	8632	289	9348	14.121756	1764.654
7	315398	2014	3302	0.962963	0.962963	8359	245	9665	13.917095	1762.1673
8	213824	1525	2783	0.9487179	0.9487179	7888	174	10207	11.730603	1760.6609
9	146064	1285	2529	0.9310345	0.8965517	7550	128	10591	10.448742	1757.4375
10	82126	1225	2466	0.826087	0.7826087	7349	91	10829	10.606341	1759.033
12	44264	560	1785	0.8	0.8	6891	41	11337	4.0380206	1758.2927
14	25083	524	1762	0.75	0.75	6824	29	11416	3.7671454	1751.3793

Threshold equals to 3 minutes.

Detector	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	403694	812	2543	0.9818182	0.9818182	8415	259	9595	6.9490196	1784.4402
2	493748	13057	14586	0.9483871	0.7548387	11563	719	5987	63.976131	1715.5828
3	546017	21792	23208	0.9508929	0.6696429	13699	1027	3543	87.766552	1702.3671
4	533726	18089	19449	0.9444444	0.7070707	12880	902	4487	78.284627	1709.5698
5	551893	13485	14810	0.945122	0.7926829	11825	768	5676	64.856321	1717.2122
6	500534	7370	8656	0.9652174	0.8608696	10298	529	7442	40.604098	1736.6257
7	491643	5249	6537	0.9787234	0.9468085	9626	447	8196	29.857054	1742.8345
8	446053	4735	5993	0.9651163	0.9302326	9376	404	8489	27.572099	1743.9827
9	419984	4185	5429	0.9746835	0.9367089	9139	371	8759	24.938943	1746.3639
10	359526	3984	5225	0.9722222	0.9166667	8932	332	9005	24.646216	1745.9819
12	260135	2792	4017	0.9622642	0.9056604	8329	240	9700	19.215632	1750.7583
14	105702	2439	3677	0.9090909	0.7878788	7658	142	10469	18.384826	1735.8662

Threshold equals to 4 minutes.

Detector	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	403694	812	2543	0.9818182	0.98182	8415	259	9595	6.9490196	1784.4402
2	502748	13073	14602	0.9548387	0.76129	11563	725	5981	64.030096	1714.6593
3	551136	22999	24415	0.95671	0.66667	13920	1068	3281	90.787572	1699.3436
4	531103	23351	24711	0.9484979	0.66094	13982	1066	3221	91.563582	1699.1979
5	528486	23237	24562	0.9439655	0.65948	13939	1061	3269	91.473061	1699.8388
6	532775	22716	24002	0.9473684	0.67982	13805	1045	3419	90.157986	1700.867
7	502876	21160	22448	0.944186	0.68372	13404	977	3888	86.67174	1703.8117
8	528225	18586	19844	0.9444444	0.71717	12852	910	4507	79.886321	1708.7077
9	523994	16938	18182	0.9408602	0.74194	12479	857	4933	75.546438	1711.4796
10	517267	13752	14993	0.9567901	0.80247	11740	755	5774	66.306048	1716.1589
12	459855	7673	8898	0.9636364	0.89091	10124	512	7633	42.18629	1732.3223

14	345249	5691	6929	0.9753086	0.90123	9204	375	8690	33.039548	1731.5867

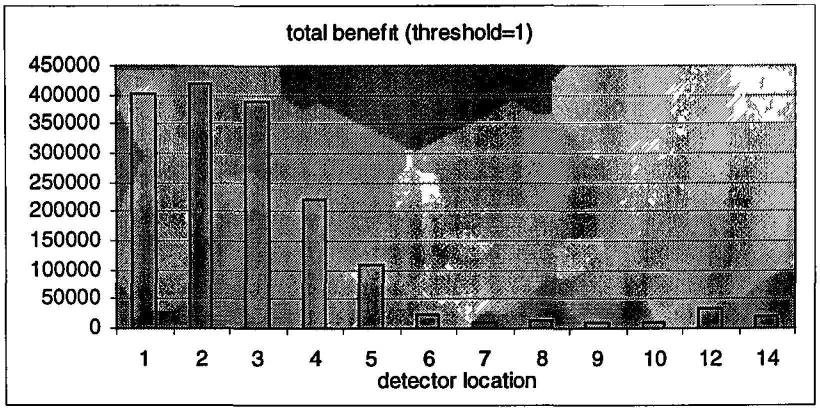
Threshold equals to 5 minutes.

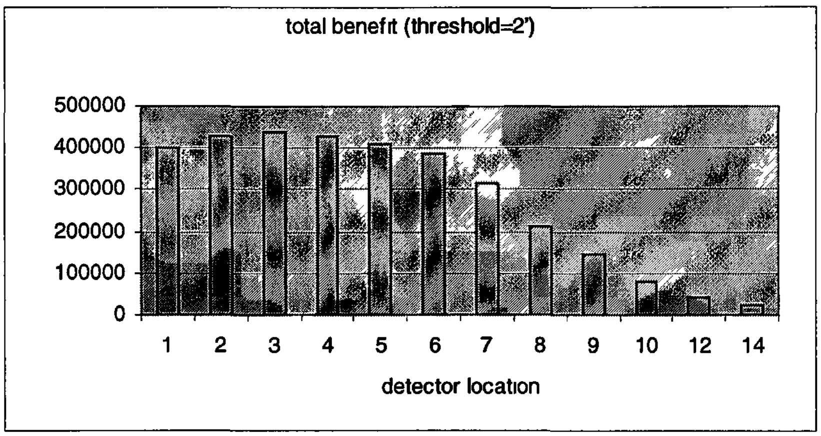
Detector	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	403694	812	2543	0.9818182	0.9818182	8415	259	9595	6.9490196	1784.4402
2	503082	13177	14706	0.9612903	0.7612903	11560	728	5981	64.427076	1714.092
3	551937	23266	24682	0.965368	0.6709957	13911	1077	3281	91.730573	1697.3083
4	106018	45057	46417	0.96875	0.534375	16367	1486	419	144.17798	1659.3398
5	173762	42941	44266	0.9648562	0.5335463	16155	1457	660	139.24327	1663.1688
6	337544	33752	35038	0.9710145	0.6014493	15132	1269	1868	118.62682	1680.5398
7	384438	30815	32103	0.9619772	0.6121673	14774	1209	2286	111.89312	1685.316
8	443179	28086	29344	0.9561753	0.6454183	14446	1158	2665	104.75765	1689.5579
9	466936	26388	27632	0.9508197	0.6516393	14243	1116	2910	99.912729	1693.5421
10	460552	24785	26026	0.9487179	0.6623932	13945	1062	3262	96.202151	1696.8842
12	514028	20903	22128	0.9478673	0.7061611	13216	969	4084	86.141798	1705.3437
14	454018	18030	19268	0.9408602	0.7419355	12469	846	4954	79.424252	1707.2801

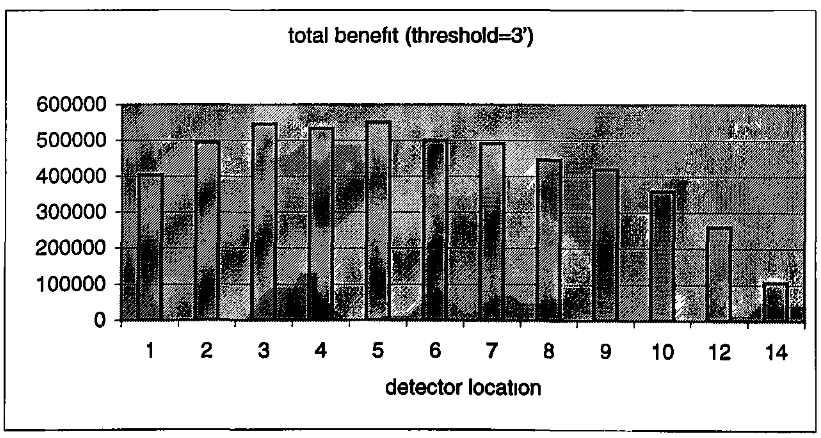
Threshold equals to 6 minutes.

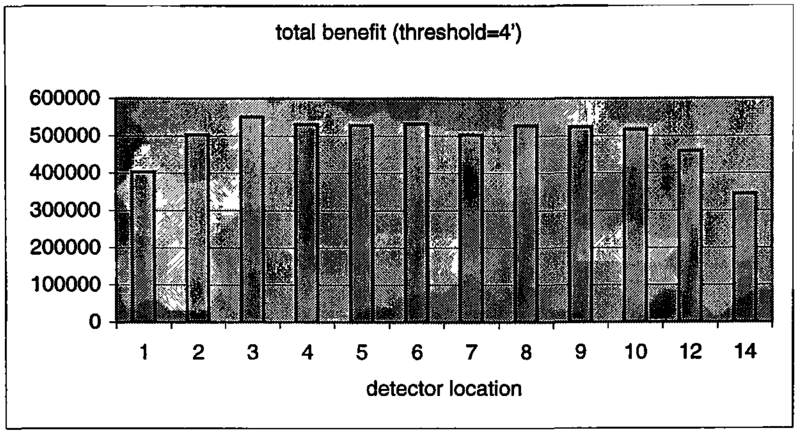
Detector	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	403694	812	2543	0.981818	0.981818	8415	259	9595	6.9490196	1784.4402
2	503082	13177	14706	0.96129	0.76129	11560	728	5981	64.427076	1714.092
3	551937	23266	24682	0.965368	0.670996	13911	1077	3281	91.730573	1697.3083
4	115993	45625	46985	0.98125	0.54375	16344	1509	419	145.78059	1655.8191
5	69909	48808	50133	0.981982	0.525526	16694	1578	0	151.92914	1651.5932
6	84728	48326	49612	0.981928	0.527108	16663	1567	42	150.41451	1653.5322
7	54165	48350	49638	0.972892	0.509036	16680	1550	42	150.5536	1655.0961
8	63149	47377	48635	0.966565	0.513678	16611	1524	137	148.33117	1658.1877
9	84617	46416	47660	0.96319	0.518405	16532	1507	233	146.21322	1660.1287
10	176136	42046	43287	0.967532	0.548701	16009	1421	839	136.86701	1665.8973
12	326315	33947	35172	0.963768	0.601449	15124	1262	1883	118.69003	1680.9691
14	330906	29399	30637	0.955645	0.637097	14322	1125	2822	109.39827	1686.8516

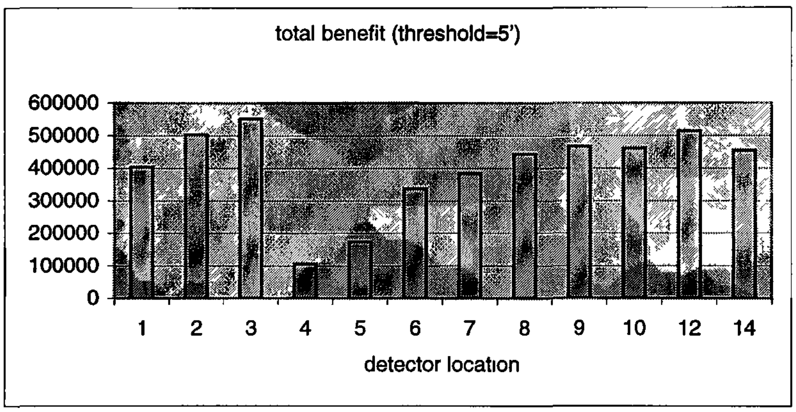
The followings are the total benefit measurement for different threshold (1 -- 6 minutes).

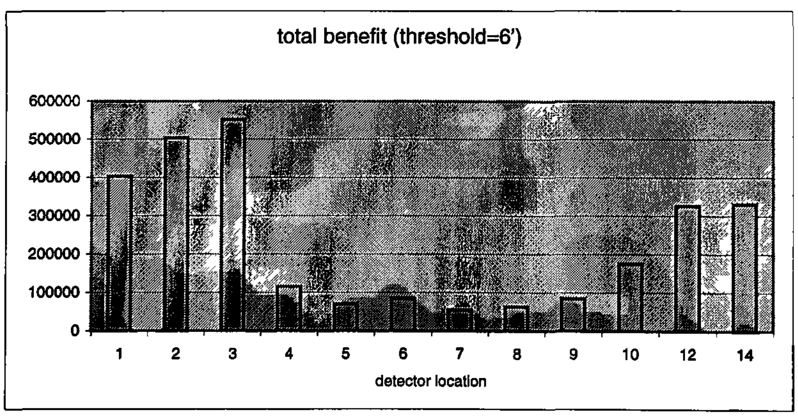

total benefit (threshold=1)
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
1 2 3 4 5 6 7 8 9 10 12 14
detector location


total benefit (threshold=2')
500000
400000
300000
200000
100000
0
1 2 3 4 5 6 7 8 9 10 12 14
detector location


total benefit (threshold=3')
600000
500000
400000
300000
200000
100000
0
1 2 3 4 5 6 7 8 9 10 12 14
detector location


total benefit (threshold=4')
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
8
9
10
12
14
detector location


total benefit (threshold=5')
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
8
9
10
12
14
detector location


total benefit (threshold=6')
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
8
9
10
12
14
detector location

5.4.3 Discussion

The general observation about the best location of the AVL detector is shown below.

Feeder Line Headway	Threshold	Locations (with the best benefit)
4	1	2
4	2	3
4	3	3 – 5
4	4	3
4	5	3
4	6	3

So generally, given route 95 headway of 4 minutes, the best detector location is 3 minutes away from the transfer stop. In the following, we will discuss some details and findings in the simulation experiments.

Threshold is used to prevent the connecting bus from holding too long, i.e. high cost control action. For control action "Good Probability", only the threshold is used to determine whether to hold the connecting bus or not. So, when the threshold is too large, the control "Good Probability" tends to behave like "Wait Next", which is worse than "No Control" in our previous simulation experiment. This is because, when the threshold is too large, it is very likely to pass the probability examination and to hold the connecting bus. From this, we will expect the total benefit degrades somehow when the threshold is getting very large. Threshold is also related to the feeder line headway: we can imagine that when the threshold is equal to or greater than the feeder line headway, almost every time we will hold the connecting bus. This is because, in a time interval of the feeder line headway there is usually a new feeder bus arrival. The relation between the threshold and the detector location is primarily from the information accuracy issue mentioned in the section 5.4.1. Given the same threshold, when the detector is located farther, the real time information is not up to date and the connecting bus is more likely not to be held. Refer to the example (figure 5.4.1.2) in section 5.4.1.

Feeder line headway's relation with the detector location will determine whether real time information will be available or not as discussed in section 5.4.1. When the detector is installed closer than one feeder line headway away from the transfer stop, sometimes the real time information may not be available about the feeder bus.

With the above general analysis, we can study the figure (1) to (4): feeder line headway of 4 minutes, the threshold is from 1 to 4 minutes. The total benefit for these cases are going up first and going down later smoothly. In Figure (1) with threshold of 1 minute, the detector locations of 1, 2 and 3 minutes are very good and 4 minutes and later are degrading the performance. This is because of the small threshold, and when the detector is installed farther away from the transfer stop, the variance is larger and the probability check is not likely to be good, so the connecting bus is not often held. While the threshold is becoming large, the peak value of the total benefit is increasing: when the threshold is set to 1 and 2 minutes, the peak value of the total benefit is about 44000. When the threshold is set to 3 and 4 minutes, the peak value of the total benefit is about 54000. This is because the threshold is too short for 1 and 2 minutes.

For figure (5) and (6), when the threshold is very large, as discussed above the control "Good Probability" tends to the control action "Wait Next". From the previous analysis and simulation experiments, we know that the control action "Wait Next" is even worse than the base case "No Control", and the main problem for "Wait Next" is that of excessive holding.

Here are the explanations about figure (5) and (6).

I. When the detector location is less than the feeder line headway, i.e. detector location equals to 1, 2, and 3 minutes in figures (5) and (6), sometimes the real time information is not available, so the connecting bus leaves at once. When real time information is available, normally the bus is held due to the large threshold in "Good Probability" control, however the actual holding time will not be so long: this is limited by the detector location. This actually prevents the connecting bus from long-time holding and yields a good result. On the other hand, when the bus is held, it normally can make a successful transfer due to the very large threshold (5 or 6 minutes). But if the detector is too close, the lack of information availability becomes the most significant factor and reduces the total benefit.

II. When the detector location is about the feeder line headway, the real time information is usually available. The connecting bus will need to check the probability and usually the result is to hold, because the threshold is very large. The detector location is not so close to transfer stop as in I (to shorten the actual holding time), and after a long time holding, the transfer is successful but the objective function generates a negative benefit.

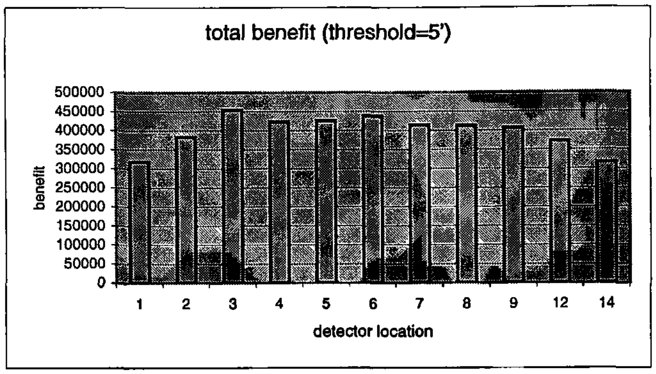
III. When the detector location is even becoming farther from the transfer stop, the connecting bus can have available information to check the probability, but the variance is becoming too large to be accommodated by the threshold. So, more times the connecting bus will not wait for the feeder bus resulted from "Good Probability" control action. The threshold starts to prevent it from long-time holding.

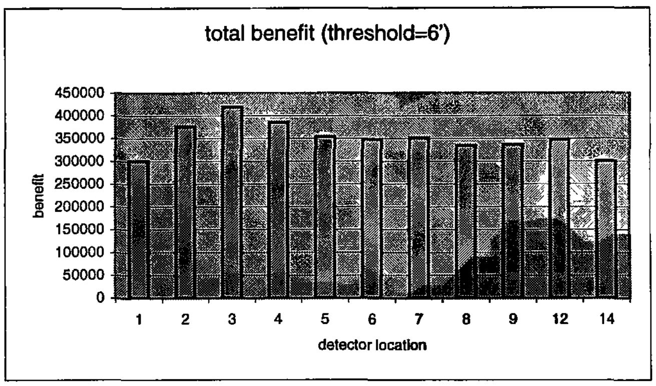
For figure (6), we have the following table from the simulation detailed data showing: how many times the connecting bus is not held over 334 transfers.

	Detector = 3'	**Detector = 4'**	**Detector = 10'**
Number of times "no real time infor"	Over 100	12	0
Number of times "too large variance"	0	2	26
In Total	Over 100	14	26

For these 2 cases, the interesting thing is that because the threshold is too large and the control action is almost transformed into "Wait Next", the detector location is good only if it can help the connecting bus not to hold too often. As discussed in control actions comparison, "always holding" is never good for our situation. In the case of detector of 4 minutes, the connecting bus is held most of the times and has the worst benefit measure.

The problem here is just showing the difference between 2 control actions "Good Probability" and "Good Benefit". Another simulation experiment is conducted with same set of the parameters except the control action is "Good Benefit".





Threshold of 5 minutes.

TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
318370	652	2383	0.975	0.975	7895	206	10168	6.2036732	1783.2427
383125	5349	6878	0.9342105	0.9210526	9075	404	8790	34.678898	1727.3168
453420	8797	10213	0.9417476	0.9223301	9950	552	7767	49.367739	1711.2844
422847	13390	14750	0.9435484	0.7741935	10613	670	6986	66.617733	1686.3597
425576	13047	14372	0.942623	0.7704918	10532	660	7077	65.344569	1687.553
438268	10135	11421	0.9357798	0.853211	10115	579	7575	54.339595	1706.2401
414249	10319	11607	0.9351852	0.8425926	10076	570	7623	55.31977	1704.6509
412160	9648	10906	0.9326923	0.8557692	9953	550	7766	52.948357	1707.5545
406245	9551	10795	0.9320388	0.8543689	9908	543	7818	52.603755	1707.9982
371859	9196	10421	0.92	0.85	9800	514	7955	51.931939	1713.6031

316853	9265	10503	0.9157895	0.8315789	9630	486	8153	53.328349	1708.6523

Threshold of 6 minutes.

TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
300681	526	2257	1	1	7666	183	10420	3.357031	1783.694
376460	3881	5410	0.9661017	0.9661017	8507	334	9428	23.784765	1732.9251
419674	5971	7387	0.972973	0.972973	8999	425	8845	34.487165	1717.7035
385854	9070	10430	0.9438202	0.7977528	9494	498	8277	48.152096	1692.7912
354267	9752	11077	0.9333333	0.7666667	9534	501	8234	51.393014	1685.1257
347962	9860	11146	0.9333333	0.7777778	9523	499	8247	51.721411	1684.3788
351653	10250	11538	0.9354839	0.7849462	9615	512	8142	53.080915	1683.6445
335391	9910	11168	0.9230769	0.8021978	9554	493	8222	51.990789	1687.8519
336480	10304	11548	0.9148936	0.7765957	9638	505	8126	53.438369	1686.1762
348363	8031	9256	0.95	0.8625	9164	440	8665	43.728066	1702.4705
300462	7484	8722	0.9324324	0.8513514	8974	396	8899	41.874972	1707.697

Control action "Good Benefit" can improve the service level when the threshold is defined very well. "Good Probability" only checks the threshold, but "Good Benefit" checks both of the threshold and the expected benefit.

5.5 Comparison Among Different Bus Holding Threshold

5.5.1 Introduction

We have discussed some relationships among feeder line headway, detector location and the threshold in previous section 5.4. Now we will study how the threshold should be defined. We consider the practical situation is that: the detector is installed already, but the threshold will be re-defined due to some impact from other routes, so we wish to see how the total benefit will be affected. From previous section, we primarily study 3 factors: feeder line headway, detector location and the threshold. So, we will check how threshold affect the total benefit with various feeder line headways settings.

Originally, the concept of "threshold" was introduced as a limitation to prevent the connecting line from prolonging the bus holding or delaying the rest of its trips or that of other connected routes (if there is timed transfer implemented for example). But here we are discussing about the internal requirement from our system: if there is no requirement from other routes, how can we define the threshold? So the context is that when there is no hard requirement from the connecting line and other connected or even non-connected routes, then we should optimize the threshold to have the maximum control benefit.

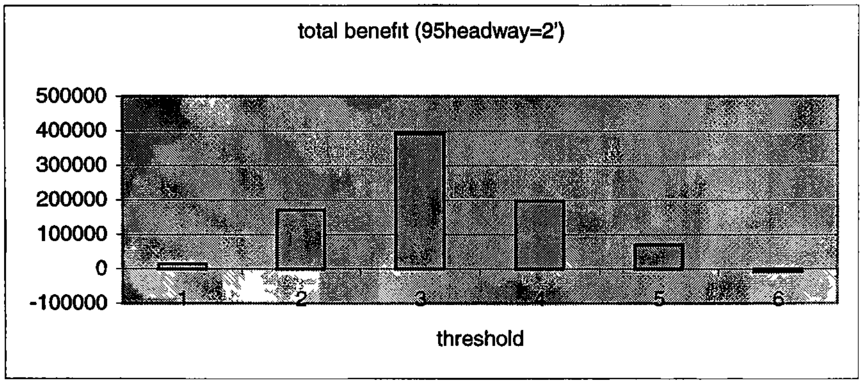
Generally, when the threshold is very small, the connecting bus will be held less. When the threshold is too large, control "Good Probability" goes towards "Wait Next" and the system performance goes down. Practically, if the threshold is set greater than the feeder line headway, the control is tending to "Wait Next", because statistically there should be a new feeder bus arrival in any time interval equal to the feeder line headway.

5.5.2 Simulation Runs

Here is the simulation arguments table.

SimuTime	DetecLoca	Trip120	HW120	Control	avgBT
600000	600	1440	1800	Good_Probability	2

Figure (1)



Figure(2)

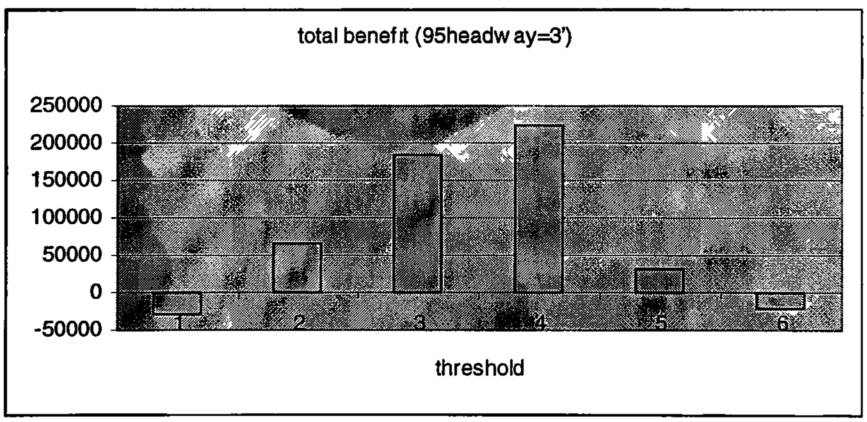


Figure (3)

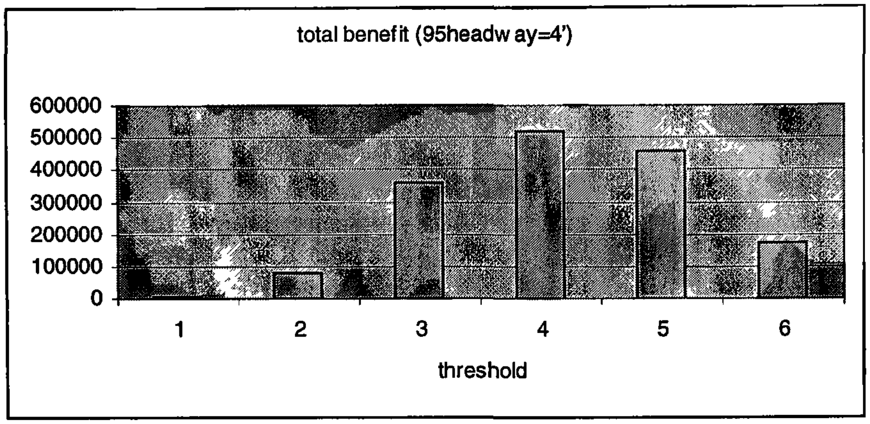
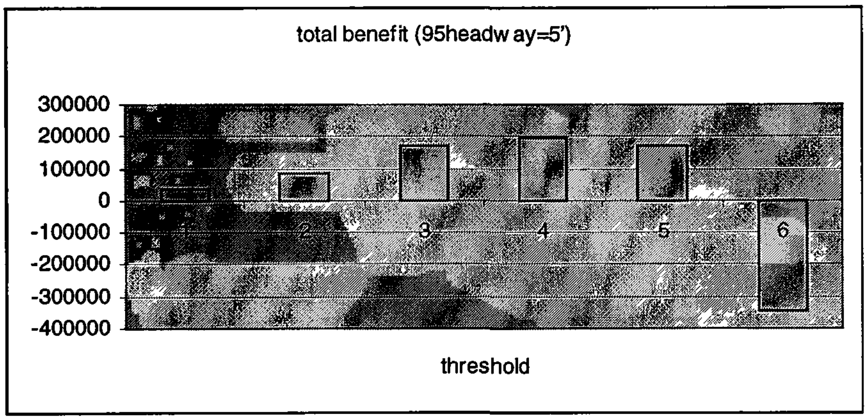


Figure (4)



Headway95 = 2'

threshold	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	15014	294	2500	0.625	0.625	7088	23	22685	3.6272573	1770.6087
2	171409	2125	4331	0.9387755	0.8163265	9794	213	19789	20.676332	1755.4601
3	393105	6675	8881	0.919708	0.810219	15447	618	13731	44.545154	1749.5049
4	197863	22805	25011	0.9482759	0.6068966	25475	1357	2964	83.821943	1719.4009
5	70218	28160	30366	0.9693252	0.5613497	27740	1558	504	93.595242	1711.5212
6	-7758	29771	31977	0.9758308	0.5589124	28058	1601	143	97.719795	1707.7227

Headway95 = 3'

threshold	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	-30252	349	1894	0.5555556	0.5555556	6999	20	15098	9.3771967	1768.95
2	65118	1000	2545	0.8695652	0.8695652	7592	97	14435	13.873551	1757.1753
3	185516	1865	3410	0.9534884	0.9069767	8510	198	13416	19.023384	1754.5707

4	222447	19604	21149	0.9560976	0.6487805	15476	903	5745	85.104032	1704.8915
5	32219	36926	38471	0.9559748	0.5314465	20012	1449	663	120.13692	1681.4348
6	-20435	39486	41031	0.9695122	0.527439	20380	1514	230	125.49377	1675.7781

Headway95=4'

threshold	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	8902	213	1454	0.5	0.5	6715	11	11543	1.544899	1752.3636
2	82126	1225	2466	0.82609	0.78261	7349	91	10829	10.60634	1759.033
3	359526	3984	5225	0.97222	0.91667	8932	332	9005	24.64622	1745.9819
4	517267	13752	14993	0.95679	0.80247	11740	755	5774	66.30605	1716.1589
5	460552	24785	26026	0.94872	0.66239	13945	1062	3262	96.20215	1696.8842
6	176136	42046	43287	0.96753	0.5487	16009	1421	839	136.867	1665.8973

Headway95=5'

threshold	TotalBenefit	Holding	Delay	s_Rate	g/s_Rate	passCost	passBenef	passZero	avg_cost	avg_benef
1	36141	348	1497	0.75	0.75	6780	29	9119	2.2023599	1761.1379
2	83455	882	2031	0.777778	0.7777778	7029	69	8830	5.4238156	1762.0145
3	170078	1670	2819	0.933333	0.9333333	7325	139	8464	9.8719454	1743.8129
4	195608	2682	3831	0.975	0.9	7598	182	8148	15.733614	1731.6044
5	173923	5857	7006	0.95082	0.6885246	8182	267	7479	34.279394	1701.8614
6	-344174	45028	46177	0.944206	0.3218884	12348	1065	2518	166.23704	1604.2451

5.5.3 Discussion

From the above 4 graphs, we can see that when the threshold increases, the total benefit increases at first and decreases after. The reason is that both very small threshold and very large threshold will degrade the system performance.

If the threshold is too small, it can not accommodate the traveling time variance from the detector location to the transfer stop. On the other hand, if the threshold is too large, as discussed in section 5.4 – The Optimized AVL Detector Location (5.4.3), the control action now actually becomes similar to "Wait Next" since the threshold is not working as holding time limit since the high frequency feeder will almost certainly arrive before the threshold is reached. We also have already seen some discussion about the threshold, feeder line headway and the detector location. So single factor here may not have a definite effect on the system performance, instead, more consideration should be on the relationships among these 3 aspects.

Since all these experiments have the same AVL detector location, we assume the real time information has similar variance, so we will not consider this issue here. We primarily examine the relationship between the threshold and the AVL detector location. From the above 4 experiments, we have the following table:

95 Headway	**Threshold with peak benefit**
2'	3'
3'	4'
4'	4'
5'	4'

Generally, the best threshold is about 3 to 4 minutes for our case. More importantly, we should avoid to define very small or very large threshold in the real time transfer control system.

5.6 Impact of the Number of Passengers

5.6.1 Introduction

Directly from the objective function, we can see that Nc/Nf can affect the benefit a lot. It is a linear relationship with total benefit.

One thing need to notice is that, Nc and Nf are not totally independent. If we split the Nc into Nb – number of passengers on board of the connecting bus, and Nd - number of passengers waiting in downstream, then Nb is actually the sum of the transfer passengers from the feeder line. In our default set of the simulation arguments, there are about 7 to 8 feeder buses arrived between 2 connecting bus trips, so the total number of passengers boarding on the connecting bus is about 7 to 8 times of the average transfer passengers from the feeder bus. But Nd is totally independent to Nf. So in the following experiments we run simulation with different number of passengers on feeder bus and in the downstream of the connecting line.

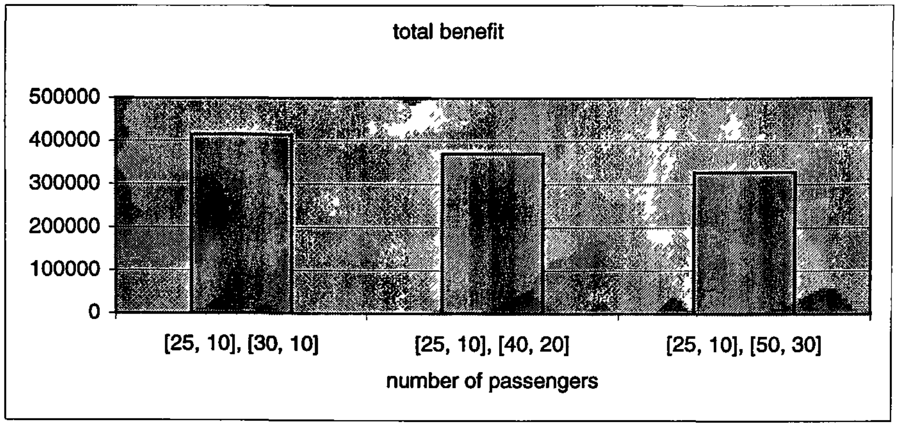
5.6.2 Simulation Runs

Simulation arguments table.

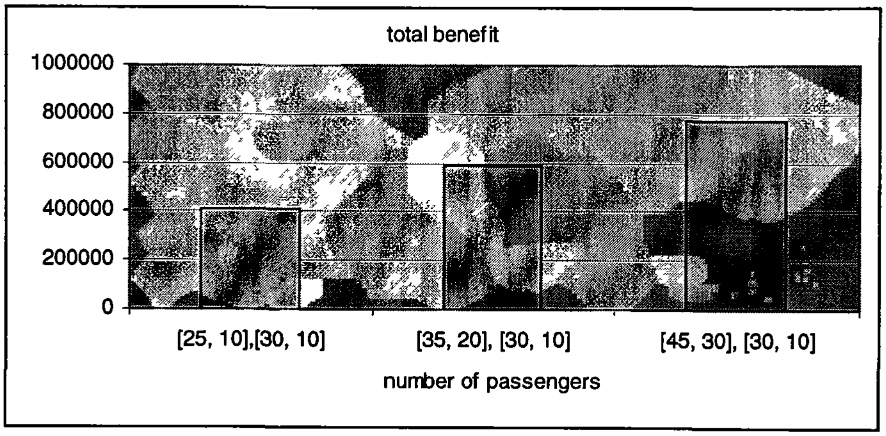
SimuTime	DetecLoca	Trip120	HW95	HW120	Control	Thresh	avgBT
600000	600	1440	240	1800	Good_Probability	180	2

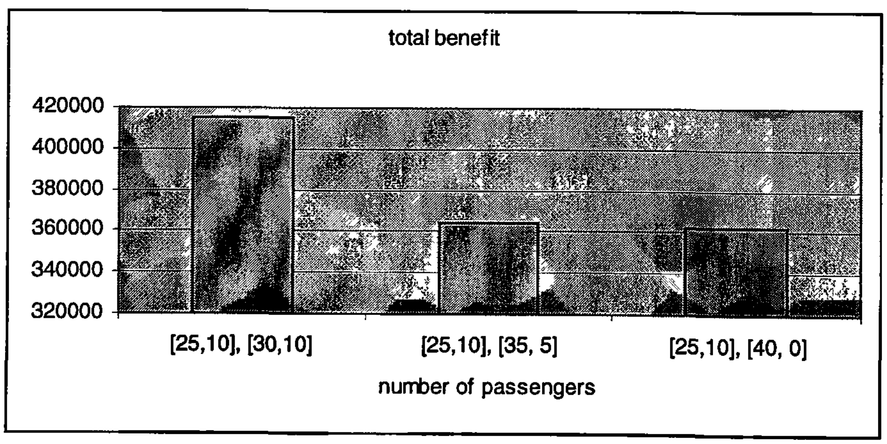
Simulation output data table.

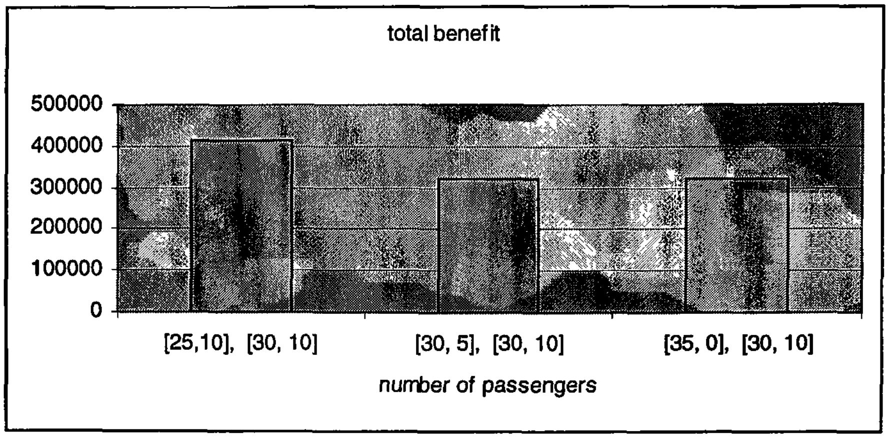
Feeder	Connecting	Total Benefit	Holding	Delay	s_Rate	g/s_Rate	Pass Cost	Pass Benef	Pass Zero	avg_cost	avg_benef
[25, 10]	[30, 10]	414951	3987	4933	0.9480519	0.9090909	9075	364	8833	24.46755	1749.9835
[25, 10]	[40, 20]	371601	3987	4933	0.9480519	0.8961039	12415	364	8833	21.3768	1749.9835
[25, 10]	[50, 30]	328251	3987	4933	0.9480519	0.8831169	15755	364	8833	19.59651	1749.9835



For the data tables of the following simulation experiments, refer to the 5.7.4.







5.6.3 Discussion

From Figure 5.7.1, the number of passengers in the downstream of the connecting line increases and its variance keeps same. Then the total benefit will go down. Compared with Figure 5.7.2, while the average number of passengers in feeder bus increases with the same variance, the total benefit will increase. This result is consistent with the objective function analysis.

With Figure 5.7.3, while the average number of passengers in downstream along the connecting line keeps same, but the variance increases, then the total benefit goes down. From Figure 5.7.4, while the average number of passengers in feeder bus keeps same, but the variance increases, then the total benefit goes down as well. This indicates that any more uncertainty about the number of passengers will result more negative evaluation in total benefit. How to have accurate real time information about the number of passengers (either total passengers number or number of transferring passengers) will have a large influence on the real time bus dispatching result.

5.6.4 Data Tables

These are the output data tables for each of the above experiments in Figure 5.7.2, 5.7.3, and 5.7.4 respectively.

Feeder, Conne	Total Benefit	Holding	Delay	s_Rate	g/s_Rate	Pass Cost	Pass Benef	Pass Zero	avg_cost	avg_benef
[25, 10],[30, 10]	414951	3987	4933	0.948052	0.909091	9075	364	8833	24.4676	1749.98
[35, 20], [30, 10]	591394	4418	6272	0.972222	0.958333	10414	542	14798	33.7838	1740.26
[45, 30], [30, 10]	774353	4799	7660	0.918919	0.905405	12035	740	20464	42.3138	1734.59

Feeder, Conn	Total Benefit	Holding	Delay	s_Rate	g/s_Rate	Pass Cost	Pass Benef	PassZero	avg_cost	avg_benef
[25,10], [30,10]	414951	3987	4933	0.94805	0.909091	9075	364	8833	24.4676	1749.9835
[25,10], [35, 5]	363756	3984	5225	0.97222	0.930556	8862	332	9005	24.3636	1745.9819
[25,10], [40, 0]	362366	3984	5225	0.97222	0.930556	8812	332	9005	24.6596	1745.9819

Feeder, Conne	Total Benefit	Holding	Delay	s_Rate	g/s_Rate	Pass Cost	Pass Benef	Pass Zero	avg_cost	avg_benef
[25,10], [30, 10]	414951	3987	4933	0.948052	0.909091	9075	364	8833	24.46755	1749.9835
[30, 5], [30, 10]	321233	3968	5249	0.972222	0.847222	8922	312	9016	25.09437	1747.1955
[35, 0], [30, 10]	322613	4010	5441	0.972222	0.777778	9041	325	9095	27.01383	1744.1385

5.7 Bus Route Schedule Coordination Impact

5.7.1 Introduction

From the original research proposal, we do not need timed transfer as requirement to dispatch transfer buses in real time. However, this schedule coordination does have impact on the simulation results, so we will discuss some relevant issues in this section.

First of all, almost all the previous simulation experiments have the same schedule coordination degree. This is done by predefining the arrival time of the first feeder bus at Orleans. For those impacted by coordination issue, we have the explanation in each section.

The following is an example of how the schedule coordination affects the simulation experiments. Assuming the headway of feeder line is 4 minutes and the headway of connecting line is 30 minutes, the first bus of feeder line 1 arrives at Orleans at time 14' and the first bus of feeder line 2 arrives at Orleans at time 15'. The connecting line start from Orlean at time 0 and next departure time is scheduled at 30.

Feeder1	Connecting	Feeder2
14	0	15
18		19
22		23
26		27
30 *	30	31 *
34		35
38		39
42		43
46		47
50		51
54		55
58		59 *
62 *	60	63
66		67
70		71
74		75

78		79
82		83
86		87
90 *	90	91 *
94		95
98		99

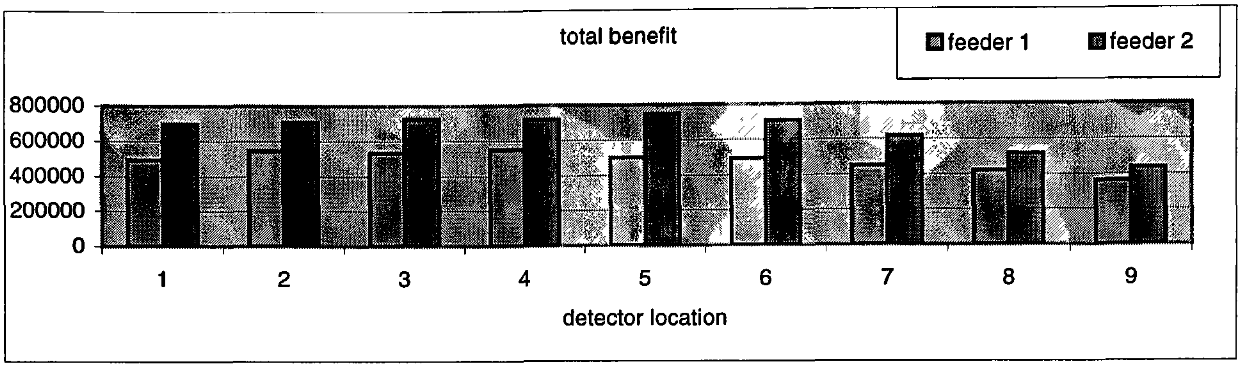
* shows which one is to be the coordinated feeder bus.

From the above table, we can see that for feeder line 1, the coordination pattern is one feeder on time, one feeder late for 2 minutes. For feeder line 2, the coordination pattern is one feeder early for 1 minute and one feeder late for 1 minute. The last column in the right of the table shows another feeder line arrival time, but it has the same coordination pattern which is equivalent to that of feeder line 1.

From previous analysis, we can easily know that the best coordination pattern is feeder line 2. It is because, the connecting bus will have more chance to catch the occational "late" feeder bus within the threshold and get positive control benefit. If the feeder line and the connecting line are totally coordinated, the feeder bus has 50% probability to arrive before sdt of the connecting bus, and this is considered to be 0 benefit. The worst case is late or early 2 minutes and above, since the probability that feeder buses are frequently 2 minutes later or earlier than schedule is very small.

5.7.2 Simulation Experiments and Discussion

Based on the above table of schedule coordination data, we run the simulation and have the following results.



As expected, feeder line 2 has better result than feeder line 1.

We run another simulation experiment with the following data. The feeder line headway is 3 minutes and the headway of the connecting line is 30 minutes. We run 3 feeder lines with different schedule as in the following table.

Feeder 1	Connecting	Feeder 2	Connecting	Feeder 3
14	0	**15**	0	**16**
17		18		19
20		21		22
23		24		25
26		27		28
29 *		30 *	30	31 *
32	30	33		34
35		36		37
38		39		40
41		42		43
44		45		46
47		48		49
50		51		52
53		54		55
56		57		58
59 *	60	60 *	60	61*
62		63		64
65		66		67
68		69		70
71		72		73
74		75		76

77		78		79
80		81		82
83		84		85
86		87		88
89 *	90	90 *	90	91 *
92		93		94

* shows which one is to be the coordinated feeder bus.

The feeder line 1 has the coordination pattern: feeder is always early 1 minute. The feeder line 2 has the pattern: feeder is always on time. The feeder line 3 has the pattern: feeder is always late 1 minute.

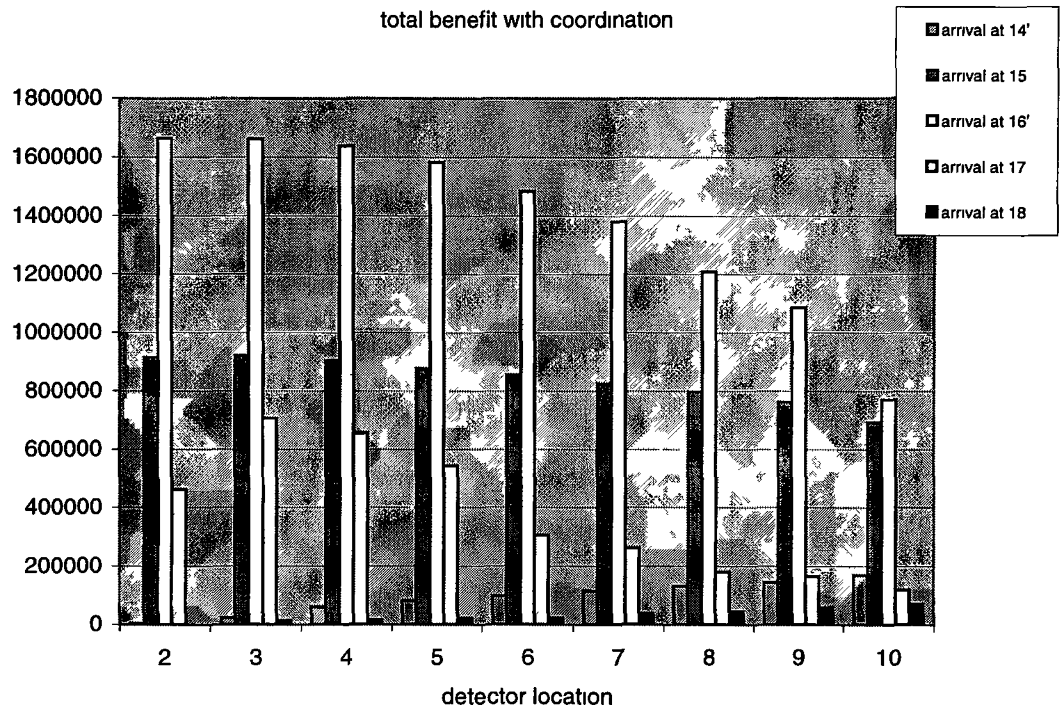
We can see the impact of the schedule coordination is very significant.

Lastly, we run another simulation with the following data. The headway of the connecting line is still 30 minutes and the feeder line headway is 5 minutes. The coordination is listed in the following table.

Feeder1	Connect	Feeder2	Connecti	Feeder3	Connecting	Feeder4	Connecting	Feeder5	*Same as 1*
14	0	15		16		17		18	19
19		20		21		22		23	24
24		25		26		27		*28	29
29*	30	30*	30	31*	30	32*	30	33	34
34		35		36		37		38	39
39		40		41		42		43	44
44		45		46		47		48	49
49		50		51		52		53	54
54		55		56		57		58*	59
59*	60	60*	60	61*	60	62*	60	63	64
64		65		66		67		68	69
69		70		71		72		73	74
74		75		76		77		78	79
79		80		81		82		83	84
84		85		86		87		88*	89
89*	90	90*	90	91*	90	92*	90	93	94
94		95		96		97		98	99
99		100		101		102		103	104

* shows which one is to be the coordinated feeder bus.

In this case, the feeder line 1 has the coordination pattern: feeder is always early 1 minute. The feeder line 2 has the pattern: feeder is always on time. The feeder line 3 has the pattern: feeder is always late 1 minute. The feeder line 4 has the pattern: feeder is always late 2 minutes. The feeder line 5 has the pattern: feeder is always early 2 minutes. The other cases are duplicated in the coordination pattern.



From the simulation results, we can see that feeder line 3 has the best benefit due to its best coordination.

Chapter 6, Conclusions and Future Research

This chapter will discuss the knowledge and experience gained from this research work, and research extensions to this research work.

6.1 Conclusions

Through this thesis research, we have studied real time bus dispatching to improve bus transfer in an environment with the following characteristics: transfer occurs from a high frequency route to a low frequency route, no timed transfer implemented in advance and the threshold is used to limit the maximum bus holding time.

For the bus transferring from high frequency to low frequency, real time bus control can improve the service level. GPS technology has better performance than AVL detector based monitoring technologies, because GPS can provide the latest real time information allowing for the minimization of the prediction variance. But the AVL technology is more economical and is in fact what is currently being used at OCTranspo which provides the case study for this work. On the other hand, any control actions similar to "Wait Next", or even similar to "Wait First" will not improve the service level, and even degrade the system performance.

Feeder line headway, detector location and the threshold are important factors to the system performance. For a given feeder route, the headway is normally defined, and the detector location will determine the real time information availability by the following:

- If detector is installed less than feeder line headway, the real time information provided by the AVL detector will not always be available.

However this could be used to prevent the connecting bus from excessive holding, which could result in a negative benefit. From our experiments, this is useful when the threshold is defined too large and can not limit the connecting bus holding time. The advantage is that the real time information will be relatively accurate.

- If the detector is installed equal to or greater than the feeder line headway, the information is quite available but the real time information accuracy is relatively bad.

Threshold has a significant effect on the system performance. When the threshold is too small, it can not accommodate the travel time variance and the connecting bus is held rarely. When the threshold is too large, it will not be an effective holding time limit. If the control action is totally depending on the threshold and the threshold is not defined well, the system performance can be largely degraded, even worse than no control due to excessive holding. The threshold can be defined very large, but the actual bus holding time will not increase a lot, since implicitly we only wait for the next feeder bus, so the expected waiting time is about half the headway of the feeder line.

Bus schedule coordination issue still exists and has significant impact in out simulation, although for some experiments we set the coordination degree to the same level. However, ideally a high frequency bus route such as No. 95 in our simulation, should operate under "headway control" strategy, instead of the current "schedule control" strategy. This is a conclusion taken from the work of previous researchers who were looking at a "bus route control strategy study" on the same project. "Headway control" strategy basically aims at maintaining an even headway among the buses along the route, but not necessarily adhere to the time table. In this case, the "timed transfer" or "schedule coordination" will not work, since the time table is totally not predefined. On the other hand, if we know the 2 bus routes that have a certain degree of coordination, it is worth to check the impact of this coordination before implementing the real time dispatching system. If the feeder line is often a little late than the connecting line, the real time control will expect a very good improvement, however, if the coordination could make most transfers successful, we should discuss how meaningful to implement real time control.

6.2 Future Research

Our bus transfer model can be improved in the following aspects:

- A more sophisticated model of the number of passengers could be designed to include a time dependent transfer rate (percentage of passengers transferring): the closer we get to having feeder arrive around s.d.t, the longer the percentage of passengers seeking a transfer to the connecting line. We could also model other passengers coming from other sources than from the feeder.
- A more comprehensive traveling time model could be designed. For example, it can be associated with the number of passengers on board. It also can be related to the traffic conditions, so that the traveling time of two consecutive feeder buses, for example, not be totally independent: both should see generally the same traffic conditions, and be affected by traffic in the same way.
- A new model should be able to work with multiple routes transferring: from multiple feeder lines to multiple connecting lines. This makes sense to central bus stations with many bus routes connected to each other.
- Design and test more complex control action, such as multiple or comprehensive objective functions.
- Implement dynamic control action selection based on time of the day or the day of the week, which means for each transfer, we can choose different control actions in the different time of a day, or in the different season.
- Model multiple detectors provide real time information to the control center. This is expected to improve the AVL detector system with the real time information, and will involve some work on manipulating the real time data.

- Model more complex and realistic passengers arrival processes at the transfer stop.

Another way to improve the research work is to improve the data quality. First is to improve and test the simulation against real data. Although we tried our best to keep this work realistic, the limitation still exists and we had to make some assumptions. For example, there could be a better way to model the passengers arrival pattern at the transfer stop, but we ignored some details for simplicity. Generally, we think 2 important details can be modeled: first is to consider the other passengers sources, such as shopping mall and residence. Second is the number of transfer passengers from the feeder line is not evenly distributed. We would like to expect more transfer passengers coming in when it is close to the schedule departure time of the connecting bus. Today the public transit information is published quickly and broadly, so it is feasible for an individual passenger to have a rough time estimate for the transfer.

References

[1] Operational Feasibility of Timed Transfer in Transit Systems,
Journal of Transportation Engineering, Vo 113, No.2, 1987,
Mark Abkowitz et al,

[2] Real -Time Dispatching Control for Coordinated in Transit Terminals,
Transportation Research Record 1433, 1994,
Kurt Ker-Tsung Lee and Paul M. Schonfeld

[3] Distribution of Bus Transit On-time Performance,
Transportation Research Record 1202,
Richard P. Guenchner and Kasimin Hamat

[4] Timed Transfer: An Evaluation of its Structure, Performance and Cost
Systan Inc., 1983

[5] Transfer Optimization in a Transit Network,
Transportation Science, Vol. 26, No.2, May, 1992,
James H. Bookbinder and Alain Desilets,

[6] Planning and Design Model for Transit Route Networks with Coordinate
Operation, Transportation Research Record 1623, 1998, pp16-23,
Mao-Chang Shih Hani S.Mahmassani, and M.Hadi Baaj,

[7] Public Transit in America: Findings from the 1995 Nationwide Personal
Transportation Survey, September 1998, *Center for Urban Transportation Research University of South Florida.*

[8] DOT (Department of Transportation, the United States)
web site: http://scitech.dot.gov/reeng/sensmsrm/sensmsrm.html

[9] Study of Bus Control Strategies by Computer Simulation,
1994, Master's Thesis, University of Ottawa, Ottawa, Ontario,
Lule Chen

[10] Using Headway Control to Improve High-Frequency Bus Service Reliability,
Working Paper, 1996, 96-04, University of Ottawa, ISSN 0701-3086,
Francois M. Julien, Lule Chen