

**VEHICLE ROUTING AND  
SCHEDULING FOR THE  
ULTRA SHORT HAUL  
TRANSPORTATION SYSTEM**

**Barry C. Smith**

**Report FTL-R79-3  
July 1979**

**MIT**

**DEPARTMENT  
OF  
AERONAUTICS  
&  
ASTRONAUTICS**

**FLIGHT TRANSPORTATION  
LABORATORY  
Cambridge, Mass. 02139**

**VEHICLE ROUTING AND SCHEDULING  
FOR THE ULTRA SHORT HAUL  
TRANSPORTATION SYSTEM**

**Barry C. Smith**

**FTL Report R79-3**

**May 1979**

**Flight Transportation Laboratory  
Massachusetts Institute of Technology  
Cambridge, Massachusetts**

## ABSTRACT

A method of vehicle routing and scheduling for an air based intraurban transportation system is developed. The maximization of level of service to passengers in a system operating under time varying demand is considered on both the optimal and heuristic levels. It is shown that while the determination of an optimal schedule is mathematically feasible, it is computationally impractical. Heuristic vehicle control algorithms are developed and tested using computer simulation. It is shown that, as compared to fixed routing strategies, dynamic vehicle routing strategies provide a greater level of service to passengers while substantially reducing the direct operating costs of the system.

## TABLE OF CONTENTS

Chapter I - Introduction	5
1.1 Motivation for Research	5
1.2 Summary	7
1.3 Terminology	9
1.4 The USH System	13
Chapter II - Simulation Model for the USH System	18
2.1 GASP	19
2.2 USH Simulation	22
2.2.1 Passenger Simulation	24
2.2.2 Simulation of Passenger Arrivals	28
2.2.3 Vehicle Simulation	40
Chapter III - Simulation Investigation of USH performance	42
3.1 The Operating Scenario	42
3.1.1 The Demand	44
3.1.2 The Vehicles	48
3.2 Fixed Control Strategies	49
3.2.1 Allstop Routing	51
3.2.1.1 Simple Controls for Allstop Routing	61
3.2.2 Nonstop Routing	68
3.3 Dynamic Vehicle Controls	73
3.3.1 Dynamic Vehicle Routing	75
3.3.2 Vehicle holding Strategies	89
3.3.2.1 Minimum headway Controls	90
3.3.2.2 Holding based on Marginal Cost	93
3.3.2.3 Holding based on waiting and operating Costs	100
3.4 Summary of Results	107

Chapter IV - Suggested Topics for Further Research	114
4.1 Immediate Applications of Models	114
4.2 Modifications to Models	115
4.3 Fixed vs. Dynamic Scheduling	116
4.4 LOS	118
Appendix A - Optimal Scheduling for the USH System	126
Appendix B - SMUSH User's Guide	135
Appendix C - Summary of future V/STOL Operating Costs	180
References	195

## Chapter 1

INTRODUCTION1.1 Motivation for Research

The Ultra Short Haul (USH) system is an air-based intraurban transportation system. The combination of advances in V/STOL technology, and a degradation in the quality of surface travel, has made USH a potentially attractive alternative transportation system in some urban areas. The USH system has significant advantages over ground-based systems in the urban environment. Since its only ground facilities are terminals, the USH system requires a fraction of the land necessary for a comparable ground-based system. The resulting community impact (noise, exposure, housing and business displacement) is also far less. Since the vehicles do not travel on fixed tracks or guideways, vehicle routings are not built into a USH system as they are in ground-based systems. This makes the air-based system more adaptable to changing transportation patterns of the area it serves.<sup>1</sup>

To become operational, the USH system must break into a market against established, subsidized competition. The system's greatest attraction for passengers is high level of service (low travel times). But this system is relatively expensive to operate, and the passengers receive this better service at the expense of higher fares. While there is a market for this type of service, the fraction of travelers with a high enough value of time to use the system regularly is probably small.

In order to attract enough demand to maintain operations, the USH system must offer a high level of service (LOS) to all potential customers. The system's ability to offer this high LOS depends on good overall design, and effective use of resources during operation. System design involves the selection of an operating area, and the determination of network layout and fleet requirements. Operations deal with the routing and scheduling of vehicles, the goal of which is to offer the maximum LOS to passengers. The operations of the USH system are complicated by the nature of the demand, a significant portion of which comes from the journey-to-work market. This demand is highly directional and concentrated in the morning and evening rush hours. Therefore, the feasibility of the USH system, to a large extent, depends on its ability to operate efficiently under significantly varying demand patterns during the day.

A total system analysis and optimization was conducted by Mann, and a system to serve the Long Island area was designed.<sup>2,3</sup> The aircraft requirements and schedules were determined using a fleet assignment model FA-4.5<sup>4</sup>. This model assumes that daily demand for a particular origin-destination pair is strictly a function of frequency of service. That is, for a given number of departures, there is a unique demand. This arises from the assumption that average wait time, and hence average trip time, depend only on headway. For a system with relatively constant demand over time, and evenly spaced departures, this is a reasonable model. But in a system with significant time variations in demand, the LOS offered to a particular passenger depends not only on

the total number of daily departures, but also on their times, and on the time at which service is requested. This aspect of the USH system makes the applicability of FA-4.5 questionable.

A review of existing vehicle routing and scheduling methods concluded that no techniques truly applicable to the USH operating environment are available. It is the lack of such a design tool which provides the motivation for this study. The goal of this work is to develop a method of vehicle routing and scheduling to be used as part of a total USH system design.

## 1.2 Summary

The anticipated result of this study is a methodology to be used in USH system analyses, the usefulness of which depends on its applicability to future USH systems. Since methods are developed for use with, and tested on particular systems, it is essential that these testing scenarios be representative of realistic USH systems. The last section of this chapter is dedicated to identifying such an environment. This is done in two parts. First, the characteristics common to all USH systems are identified. Next, those factors unique to each operating scenario, which significantly affect system performance are considered. These factors vary widely between different areas, but the range of values is narrowed by restricting consideration to those areas highly conducive to successful USH operations. The factors determined in the actual system design, such as fleet size, vehicle characteristics, are treated as freely varying parameters.



The USH scheduling problem is first approached on the optimal level. The objective is to minimize the average time spent in the system by all passengers. The optimal schedule can be obtained as the solution of a shortest path problem. While this formulation assures the existence of an optimal schedule, it is computationally impractical. A brief discussion of optimal scheduling for the USH system appears in Appendix A.

The scheduling problem is next approached on the heuristic level. A simulation model for the USH system is developed and is used as a testbed to compare various vehicle control algorithms. This model is described in Chapter 2. The development and testing the control algorithms is described in Chapter 3. Two classes of controls (fixed and dynamic) are tested. The fixed controls are used to check out the simulation model, and to estimate the performance of the USH system when it is operating in a manner similar to existing transportation systems. The dynamic controls route the vehicles and schedule departures according to actual system conditions. Two types of dynamic control strategies are investigated, these are routing and holding controls. Dynamic vehicle routing enhances the system's ability to cope with varying demand patterns, and reduces the sensitivity of system performance to demand peaking and directionality. Holding controls are used to adjust the pace of the system to changing magnitude of demand. Three holding strategies are investigated. The first two, minimum headway, and an extension of the routing strategy, proved to be ineffective. The third strategy, based on system operating cost, offers significant reductions in operating cost with only a small degradation in LOS. The results of this

investigation indicate that system performance is maximized when the vehicles are routed dynamically and are held according to this final strategy.

Three areas for further research related to the USH system are suggested in Chapter 4. These areas are: 1) Application of the simulation model and control algorithms to USH sensitivity analysis; 2) Modification of the algorithms to more fully exploit the capabilities of the USH system; 3) Investigation of basic transportation questions, such as fixed versus dynamic scheduling, and the determination of a good measure of LOS.

### 1.3 Terminology

To reduce the possibility of ambiguity in this study, we now state definitions for several commonly used terms.

The time required for a passenger to travel from his origin (home or business) to the nearest system terminal is access time. Egress time is defined to be the time spent from deboarding the vehicle and leaving the system until the final destination is reached. For the remainder of this study, the terms origin and destination refer to the terminal where a passenger enters and leaves the USH system. Arrival time is defined to be the time at which the passenger enters the system. The time spent in the vehicle is ride time. Travel time is defined to be the sum of the access, wait, ride and egress times, or the total time spent in transit. The time the passenger spends in the system is the trip time, the sum of ride and wait times (see Fig. 1.1). Level of service, LOS, is a measure of the cost of travel time to the passenger.

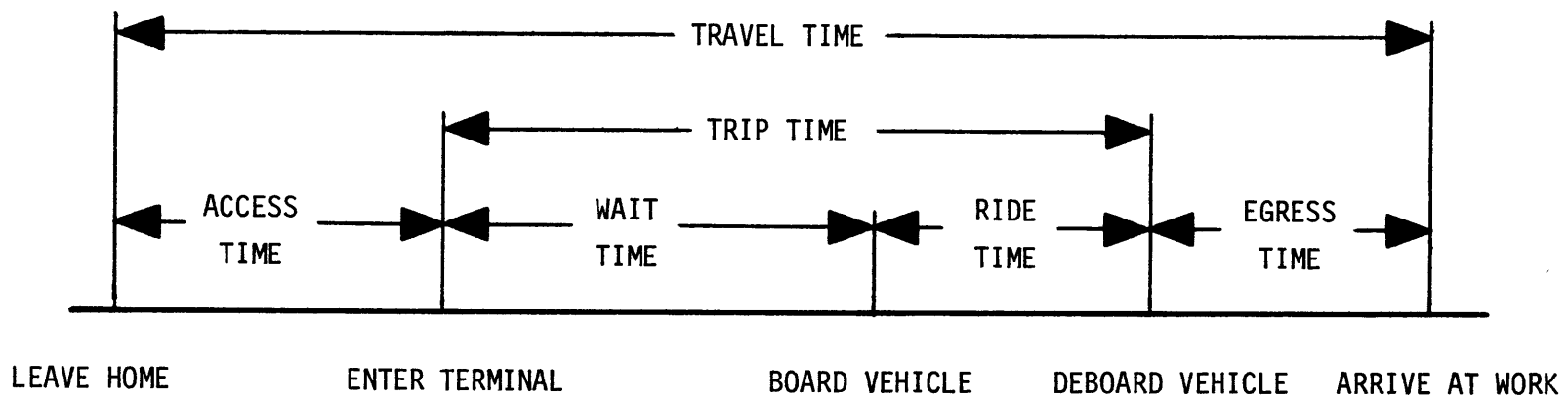


Figure 1.1 - Timeline for journey to work trip.

In general, LOS refers to average travel time. But in the urban environment, there is evidence that the uncertainty (or variance) of travel time is also a significant factor.<sup>5</sup> While this can be justified intuitively, a precise definition of LOS is not offered here. The realization that both the mean and variance of travel time are involved must suffice. Travel time depends on access and egress times and hence on system design (number and locations of terminals) which is beyond the scope of this study. Therefore, LOS will refer only to trip time. The direct operating cost, DOC, for each vehicle is the total cost incurred by flying the aircraft. This cost includes crew, fuel, depreciation, maintenance, and insurance costs. The system DOC is the sum of the costs for each vehicle. System performance is a measure which relates system costs and LOS offered to passengers. To a first approximation, performance relates average trip time and system DOC. Consider a system operating under three different policies (Figure 1.2). Clearly, maximum performance is not achieved at B, since the same LOS can be offered for less cost (A), and better service can be offered at the same cost (C). But without more information, we cannot establish whether the maximum performance is achieved at A or C. In the example, A and C comprise the set of Pareto Optimal points. More precisely,  $P_i = (E(T)_i, DOC_i)$ , is a Pareto optimal point, if  $E(T)_i \leq E(T)_j$ , and, or  $DOC_i \leq DOC_j$ , for all  $i$  and  $j$ . In a system design, many operating configurations are investigated. While the best operating point depends on system objectives (max profit, max market share, etc.), it is obvious that the system should be operated at a point on the Pareto optimal

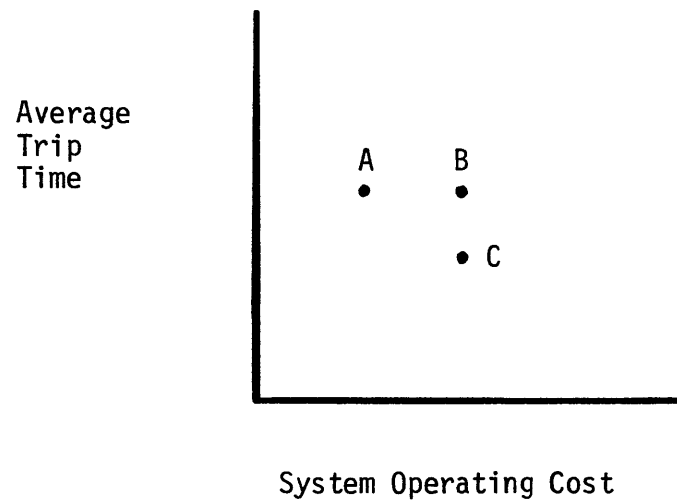


Figure 1.2 - Illustration of system performance measure.

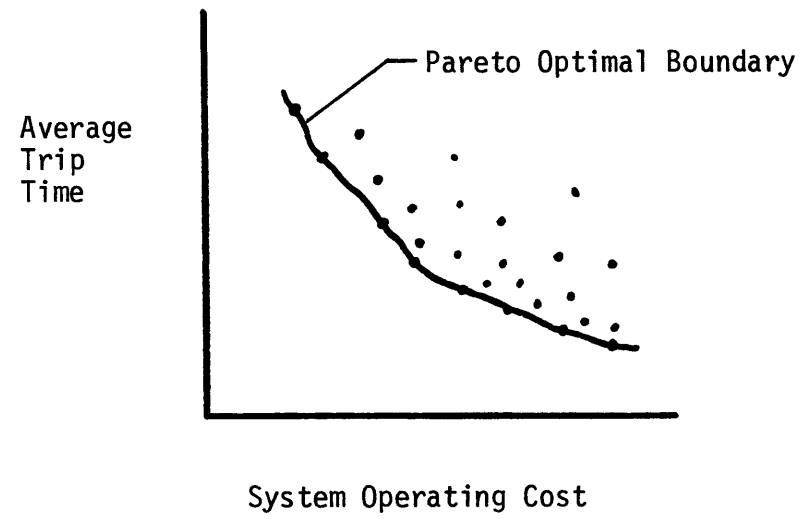


Figure 1.3 - The Pareto Optimal Boundary.

boundary (see Figure 1.3). The goal of this study, in this context, is to develop methods to establish the Pareto optimal boundary over a wide range of system configurations and operating costs.

#### 1.4 The USH System

In this section, the USH system and its operating environment are described. The purpose here is to establish guidelines for the development of operating policies and to determine the basic characteristics of the scenario in which these policies will be tested.

The USH system operates in an urban environment, which can be divided into a suburban region and a central business district (CBD). As indicated before, the fraction of all travelers who place a great enough value on their time to regularly patronize the USH system is likely to be small. Only in an urban area is the population large enough to insure a sufficient level of patronage to maintain operations. In this environment, ground transportation systems are more prone to congestion. By degrading the quality of surface travel, this congestion makes the air-based system more attractive to individual passengers. The urban environment is essential to USH operations for both reasons.

USH demand is comprised of commuter and general travelers. The demand from the commuter, or journey-to-work market is strongly time-dependent and highly directional. All requests for service from this group are made during the morning and evening rush hours. During the morning rush hours, almost all commuter trips are made from the

suburbs to the CBD, while in the evening, almost all requests are for trips from the CBD back to the suburbs. The total number of trips made by this group in the morning and evening rush hours is roughly equal. The demand from the general travel group displays neither time dependence nor directionality. There are three types of trips made by general travelers. These are: CBD to suburbs; suburbs to CBD; and suburbs to suburbs. The total demand for each type of trip will be assumed roughly equal, and the rate at which requests for service, for each type, arise will be assumed to stay constant throughout the day.

Passenger arrivals at the USH terminals are random. There are many factors which affect the actual arrival times of individual passengers. Each passenger is expected to act and be affected by them independently. Under these conditions, the level of uncertainty is such that knowledge of past arrivals is worthless in predicting the exact times of future arrivals. For these reasons, the arrival of passengers is modeled here as a Poisson process. The general passengers arrive according to a homogeneous Poisson process. The commuter passengers arrive according to a nonhomogeneous Poisson process, where the average arrival rate, (or "intensity function") depends on the time of day. For a particular OD pair, serving both passenger types, the total average arrival rate at a given time is the sum of the average rates for the two passenger types.

Due to the directional nature of the demand in the USH system, three types of nodes (or stations) can be identified. There are morning

(evening) commuter origins (destinations), which are located in the suburbs. Also, there are the morning (evening) commuter destinations (origins), located in the CBD. Both of these types of nodes display not only time-varying demand, but also unbalanced demand during the rush hours. The third type of node displays some daily variation in the level of demand, but the demand is balanced. That is, the number of passengers entering and leaving the system are roughly equal throughout the day. This is the pattern of demand at, say, airports.

The USH network is a complete graph. That is, a potential direct route exists between every pair of nodes. This property yields a tremendous amount of routing flexibility to the system, and makes the USH system truly unique among urban transportation systems.

For a particular system, the geography of the operating area plays a major role in determining the relative attractiveness of air and ground-based travel. Natural barriers, such as water and mountains can degrade the quality of surface travel, and hence reduce its market share. Bottlenecks at bridges and tunnels affect the quality of surface travel due to congestion, and the cost of tolls. These barriers also constrain suburban growth, the effect of which is to force a longer commuting distance, since the suburbs must grow away from the CBD at a rate greater than in a physically unconstrained area. For example, in a coastal city where the suburbs are forced to grow to one side of the CBD, the maximum commuting distance is about 40% longer than in a non-coastal city of the same area. In a city on an island or in the



mountains, the suburbs may be constrained to grow along a corridor, which could easily double the maximum commuting distance.

The effect of greater commuting distance on travel time is much more profound on surface than air travel.<sup>6</sup> This can be illustrated by comparing a plot of time versus distance traveled for the two modes (see Figure 1.4). The air-based system has higher zero length costs due to greater access and egress times, a result of having fewer terminals. As distance increases, the air system's speed becomes a significant factor, making this mode increasingly attractive. In summary, the USH system captures its largest market share in areas where natural barriers and congestion make surface travel long and difficult.

Also affecting system operations is the fraction of total demand which comes from the commuter market. This fraction determines the level of demand-peaking, which can have a profound effect on system efficiency. If fleet requirements are determined on the basis of the off-peak periods, which prevail for the greater part of the day, congestion during the peak periods can cause a drastic reduction in LOS to passengers served during the rush. This variation in LOS during the day can cause a lack of confidence in the system, and a significant reduction in overall demand. If, on the other hand, the system is designed to handle the peak demand, then for most of the day, it is underutilized.

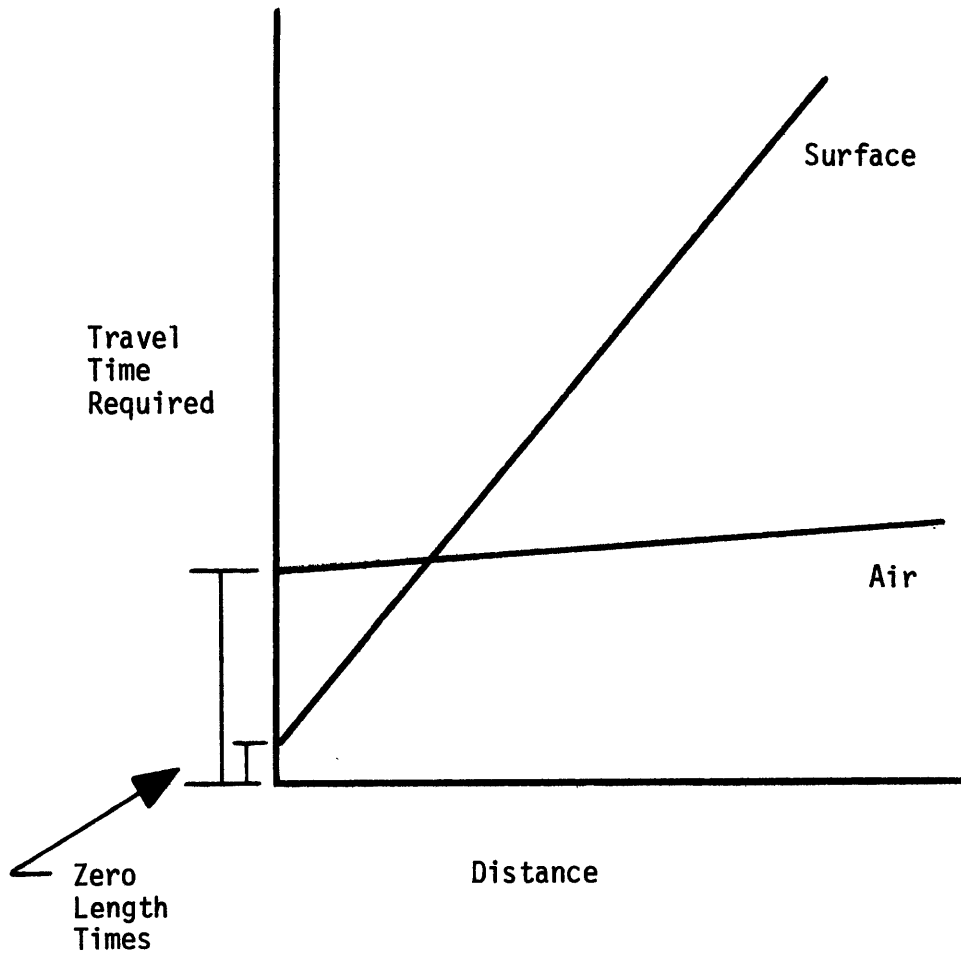


Figure 1.4 - Comparison of surface and air travel times.

SIMULATION MODEL FOR THE USH SYSTEM

In this section, the basic characteristics of the Simulation Model for the Ultra Short Haul transportation system (SMUSH) are described. SMUSH was developed specifically to analyze the performance of the USH system operating under various vehicle routing and scheduling strategies. A user's manual, containing a source list and sample input, is presented in Appendix B.

SMUSH is written in the GASP IIA simulation language. GASP is a collection of FORTRAN subroutines which provide the general framework for event paced simulation. A model written in GASP is a FORTRAN program, consisting of GASP and user provided subroutines. Due to this structure, the simulation methodology and program code are completely accessible to the user. Only a basic understanding of FORTRAN and simulation techniques is required to implement and modify the language. Also a GASP model can be run on any FORTRAN compiler. These are the primary reasons that GASP was chosen for this model.

SMUSH is composed of around 20 subroutines of which about half are standard GASP routines. The GASP portion of SMUSH provides the mechanism to store and retrieve data, advance the simulation clock, and collect statistics. The non-GASP portion is dedicated to the input, output, and handling of data unique to the USH system.

SMUSH is structured such that the following information is

specified by the user.

- Network layout
- Fleet size, and vehicle parameters
- Route structure
- Average passenger arrival rates, as a function of time of day

The output of SMUSH consists of the following:

- Vehicle schedule map
- Passenger trip time statistics
- Direct operating costs
- Vehicle load factors

## 2.1 GASP

GASP is a general framework for discrete event simulation.

In event paced simulation, system parameters are considered and updated at discrete points in time. These points correspond to the occurrence of events, such as passenger arrival, and vehicle departure events. The versatility of GASP stems from the fact that the user defines and provides the logic for all events.

GASP controls the progress of the simulated system by maintaining a list of future, or pending, events. Each entry in the list consists of an event code and time of occurrence. This list is ordered chronologically, with the most pending events first. Upon completion of an event, GASP advances the simulation clock to the time of occurrence of the next item on the list. System parameters are then

updated in a manner corresponding to this event code.

System parameters are stored in a series of files. File 1 is always the event file, and contains the information corresponding to the list of pending events. Files 2 and beyond are for the storage of user defined system parameters. (In SMUSH, for example, File 2 is used to maintain information about passengers waiting at the nodes in the USH network.) Each entry in a file contains a set of system attributes. For example, an entry in the event file corresponding to a vehicle departure contains the following information.

- Event code for vehicle departure
- Time of departure
- Station number from which vehicle is leaving
- Station number at which the vehicle will next arrive
- Route number

All file entries, and their associated attributes, are stored in two one-dimensional arrays (NSET and QSET). Integer valued parameters (such as event code) are stored in NSET, real-valued parameters are stored in QSET. (For simplicity, let NSET refer to both NSET and QSET.) The attributes associated with an individual entry are stored together as a block in NSET. But in order to save storage space in NSET, entries in each file are not stored sequentially, or necessarily together. When a new file entry is made, it is loaded into the first available block in NSET. To keep track of the files, GASP stores (in common) the addresses of the first and last entries of each file. And for each

entry, GASP stores the addresses of the predecessor and successor entries, in that file, along with the associated attributes. So for the previous example of the event file entry, the list of attributes is increased by two, and the following is added.

- Location in NSET of the event just prior to this one.
- Location in NSET of the event immediately after this one

Obviously, care must be taken in manipulating entries of these files to maintain order. Fortunately, the GASP language provides the algorithms to locate, add, and delete entries, and to update the file pointers.

For example, the algorithm to locate a file entry (Subroutine Locat) is as follows:

1. Find address of first file entry.
2. Go to this address in NSET.
3. If this is the desired entry STOP.
4. Find address of successor entry, go to 2.

To remove an entry from a file, once the location is known, Subroutine REMOVE is used. REMOVE deletes the entry, calls Subroutine SET to update the file pointers, and loads the attributes of this entry into two buffer arrays, ATRIB (real values) and JTRIB (integer values). These buffer arrays are the method of communicating information between NSET and the rest of the program. To add an entry to a file, the attributes must be loaded into the buffer arrays, then Subroutine FILEM is called to make the addition, and update the file pointers.

Simulation time is controlled by the repeated application of the following algorithm.

1. Locate and remove the first entry in the event file.
2. Advance the simulation clock to the time of occurrence of this event.
3. Call the subroutine corresponding to this event.

For any simulation variable statistics can be collected for either sample values or time weighted values of the variable. This is done by calling subroutines COLCT (sample values) and TMST (time weighted values) during the simulation. These routines maintain the following parameter statistics.

- minimum observed value
- maximum observed value
- sum of values
- sum of the squares of the values
- total number of observations

A flow chart for a general GASP model is shown in Figure 2.1.

## 2.2 USH Simulation

Using the GASP framework, SMUSH models the USH system by simulating the movements of passengers and vehicles. These movements are controlled by the use of the following five events:

- System Initialization
- Passenger Generation
- Passenger Arrival
- Vehicle Departure
- Vehicle Arrival

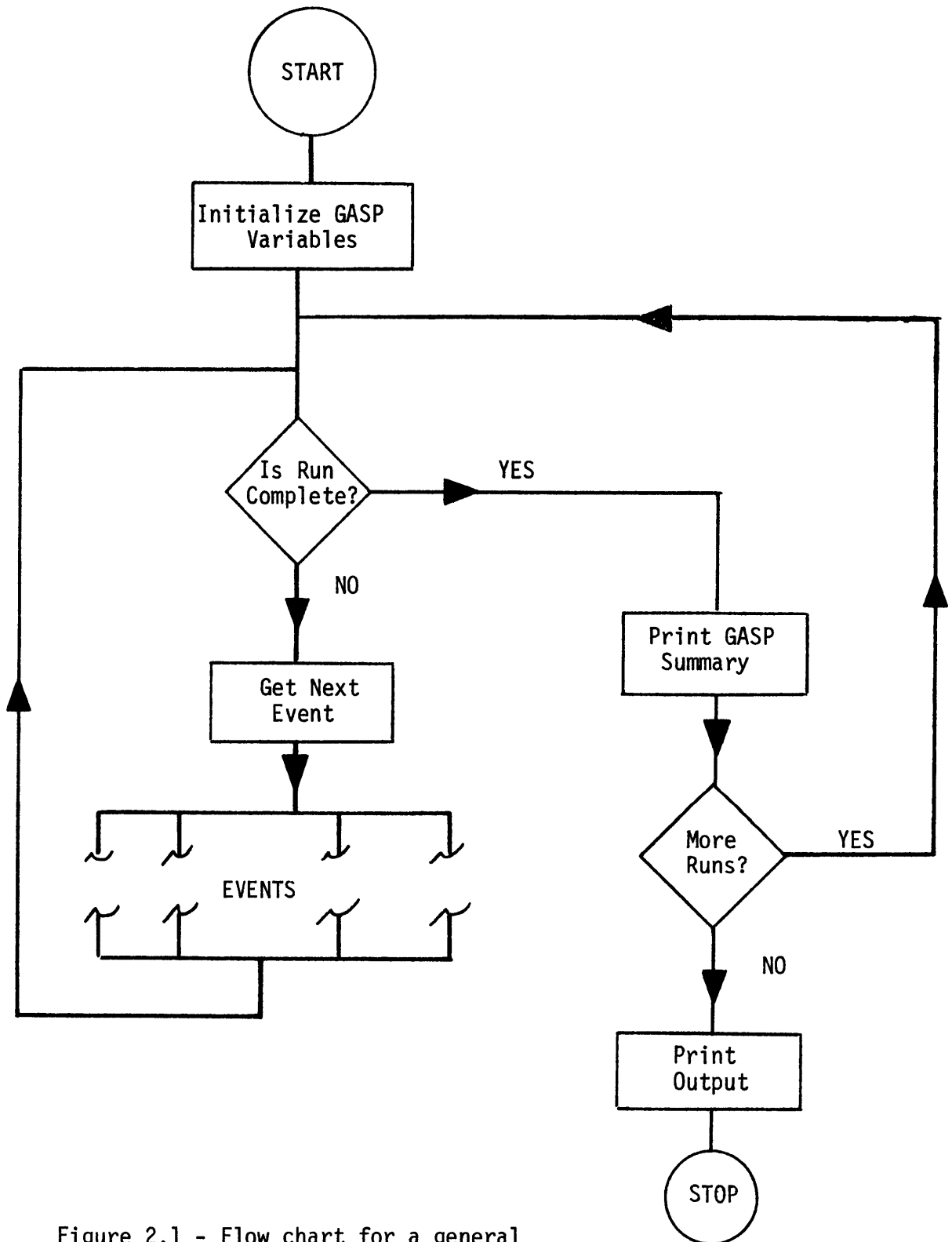


Figure 2.1 - Flow chart for a general GASP model.



Corresponding to each event is a subroutine in SMUSH, which is called when the event occurs. These subroutines update system parameters and set up future events as necessary.

The movements of vehicles and passengers are handled separately in SMUSH. The actual simulation process can be illustrated by considering first the handling of passengers, and then the control of vehicle movements through the network.

### 2.2.1 Passenger Simulation

All passengers and their arrival times are generated using the algorithm which is outlined in Section 2.2.2. At the time of generation, an entry into the event file is made for each passenger. This entry corresponds to a future passenger arrival event. The arrivals of all passengers can be generated at the start of simulation, but this would require an entry to be stored simultaneously for all passenger arrivals. To conserve storage space and to avoid the possibility of overflowing NSET in a long run, SMUSH generates passenger arrivals in half hour blocks. For example, at the start of simulation SMUSH generates and stores arrival events for all passengers who will arrive in the interval 0 to .5 hours. Another passenger generation event is then stored to occur at .5 hours into the run.

We will now trace a single passenger through the system. At the time of generation, the following attributes are stored in the event file.

- Event code for passenger arrival

- Time of passenger arrival
- Passenger's origin node number
- Passenger's destination node number
- Passenger type

Suppose this passenger is traveling from  $i$  to  $j$  and has an arrival event scheduled to occur at time  $t_a$ . When the simulation clock is advanced to  $t_a$ , this entry in the event file is removed, the attributes are placed in the buffer arrays, and the subroutine corresponding to a passenger arrival event is called. The passenger arrival event updates the number of passengers waiting at node  $i$  and stores the passenger's attributes in File 2. File 2 is a list of the attributes of all passengers waiting at the nodes in the network. This passenger remains in File 2 until a vehicle departure event occurs, which satisfies all the following conditions.

- Vehicle is leaving from node  $i$
- The vehicle will stop at node  $j$ , on its present route
- There is a seat available on this vehicle

If these conditions are satisfied, the passenger's attributes are removed from File 2, and the following entry is made to File 3. (It should be noted that the entries in File 2 are maintained in the order in which they are made. When File 2 is scanned for passengers to board, the oldest entries are removed first. Hence, the passenger loading discipline is FIFO.)

- Passenger type
- Origin ( $i$ )

- Destination (j)
- Vehicle number
- Time of arrival ( $t_a$ )
- Time of boarding ( $t_b$ )

At this point, the number waiting at node i, and the number of seats available on this vehicle are both decreased by one. The passenger then remains in File 3 until a vehicle arrival event occurs which satisfies the following conditions.

- Vehicle number corresponds to that which the passenger boarded
- Vehicle is arriving at j

If these conditions are met, the passenger is removed from File 3, and the following statistics are collected.

$$t_{\text{wait}} = t_b - t_a$$

$$t_{\text{ride}} = t_{\text{now}} - t_b$$

$$t_{\text{trip}} = t_{\text{wait}} + t_{\text{ride}}$$

It should be noted that as the passenger's attributes are transferred from one file to the next, the space left behind is cleared, and is used by future passengers. And once the passenger's entry is removed from File 3, and the final statistics are collected, SMUSH is done with him. Table 2.1 illustrates the handling of information for a single trip.

	Passenger Generated	Arrives at Origin	Boards Vehicle	Arrives at Destination
File	1	2	3	-
ATRIB(1)	$t_a$	$t_a$	$t_a$	-
ATRIB(2)	-	-	$t_b$	-
JTRIB(1)	2*	PTYPE	PTYPE	-
JTRIB(2)	ORIGIN	ORIGIN	ORIGIN	-
JTRIB(3)	DEST.	DEST.	DEST.	-
JTRIB(4)	PTYPE	-	VEH. #	-

$t_a$  = time of arrival

$t_b$  = time of boarding

PTYPE - passenger type

1) commuter

2) general

\*Event code for passenger arrival

Table 2.1 - Flow of passenger data in SMUSH.

### 2.2.2 Simulation of Passenger Arrivals

The arrival of passengers into the USH system is stochastic. To model this, SMUSH generates individual passengers according to a time varying Poisson process. Generation of the appropriate exponentially distributed interarrival times is a common method of determining arrival times in a homogeneous Poisson process. When an arrival occurs, the time of the next arrival is calculated as the sum of the present time and a random deviate corresponding to the interarrival time. The system does nothing until this next arrival occurs, and then repeats the process. Consider a non-homogeneous arrival pattern, such as the one shown in Figure 2.2. Suppose an arrival occurs at  $t_1$ . An interarrival time with parameter  $\lambda_1$  is then generated ( $\Delta t$ ). The algorithm then sits and waits until  $t_2$ . By this time, the arrival rate can change significantly, but the system is still operating under the conditions at  $t_1$ . One would expect a time lag and a significant difference between the expected and actual arrivals using this algorithm in the nonhomogeneous case. Suppose that when the system is initialized, the first arrival is calculated from  $t_0$ . At this point,  $\lambda = 0$ , and the first interarrival time is infinitely large. In this pathological case, no arrivals are generated at all. Obviously, this method of generating arrivals is not applicable to the USH simulation.

Consider that portion of the passenger arrival rate curve between  $t_1$  and  $t_2$ . SMUSH generates arrivals in this interval using the following algorithm.

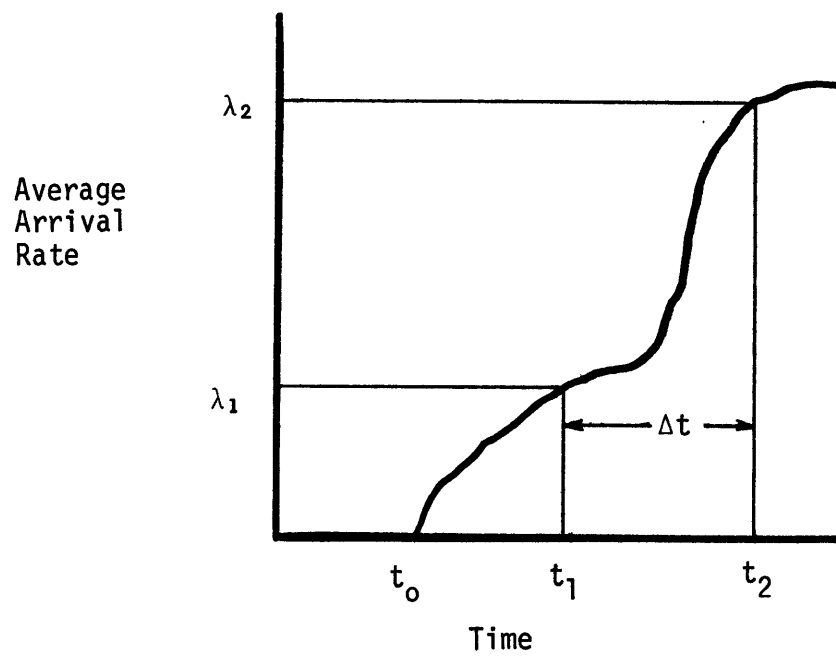
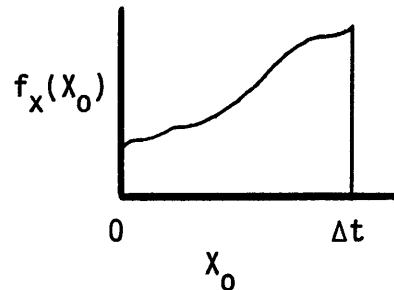


Figure 2.2 - Time varying average arrival rate of passengers.

1. Calculate the mean average arrival rate,  $\bar{\lambda}$ , in this interval.
2. Construct a pdf (figure 2.3) by normalizing the area under the arrival rate curve, between  $t_1$  and  $t_2$  to one.

Figure 2.3



3. Use the probability integral transformation method to generate random deviate,  $R$ , having exponential distribution with parameter  $\bar{\lambda}$ .
4. If  $\sum R_1 > t_2$  go to step 8.
5. Using probability integral transform of the pdf constructed in step 2, generate the random deviate  $R_2$ .
6. Passenger arrival time =  $t_1 + R_2$ .
7. Go to step 3.
8. STOP.

For each O-D pair, the SMUSH user inputs values of average arrival rates, of individual passengers, versus time. SMUSH assumes that the rate between the data points is linear. The calculation of  $\bar{\lambda}$  and the probability integral transform (in step 5) can be done in terms of the values of  $\lambda$  at the endpoints of each linear segment. This algorithm is invoked and completed automatically in SMUSH.

Due to the randomness of passenger arrivals, the performance of the real (as well as the simulated) system, is subject to daily (single run) fluctuations. To get a good estimate of system performance, it is necessary to look at data averaged over several days (or several runs). For this reason, SMUSH is set up as a Monte Carlo model. In a single run of SMUSH, the entire system can be simulated several times (the number of times is specified by the user). The output of SMUSH is the average of these runs. Ten is a reasonable number of runs for convergence of the results. But even a single run provides sufficient accuracy for the investigation of general trends.

We will now show that the proposed passenger generation algorithm preserves the fundamental properties of the Poisson type process. In a Poisson arrival process, the probability of exactly  $n$  arrivals occurring in the interval  $(0, t_0 + \Delta t)$ ,  $P_n(t_0 + \Delta t)$ , is given by Equation (2.1).

$$P_n(t_0 + \Delta t) = P_n(t_0) \Pr(\text{no arrivals in } (t_0, t_0 + \Delta t)) \\ + P_{n-1}(t_0) \Pr(\text{one arrival in } (t_0, t_0 + \Delta t)) \quad (2.1)$$

By definition of the Poisson process, the probability of exactly one arrival in the interval  $(t_0, t_0 + \Delta t)$  is equal to  $\lambda(t_0)\Delta t$ , the probability of more than one arrival is negligible, and hence the probability of zero arrivals is  $1 - \lambda(t_0)\Delta t$ . Substituting this into Equation (2.1) yields the following.

$$P_n(t_0 + \Delta t) = P_n(t_0)[1 - \lambda(t_0)\Delta t] + P_{n-1}(t_0)[\lambda(t_0)\Delta t] \quad (2.2)$$



Rearranging and taking the limit as  $\Delta t \rightarrow 0$  results in the following differential equation

$$\frac{dP_n(t_0)}{dt} = \lambda(t_0)[P_{n-1}(t_0) - P_n(t_0)] \quad (2.3)$$

Clearly, the distribution of  $n$  depends on  $\lambda(t)$  in the interval. The algorithm proposed suggests that for a given interval  $(0, t_0)$ ,  $n$  is distributed as a Poisson random variable with parameter  $\bar{\lambda}(t_0)$ , where  $\bar{\lambda}(t_0)$  is the average value of  $\lambda(t_0)$  in the interval, and is defined as follows

$$\bar{\lambda}(t_0) = \frac{1}{t_0} \int_0^{t_0} \lambda(x) dx \quad (2.4)$$

$$P_n(t_0) = \frac{(\bar{\lambda}(t_0)t_0)^n e^{-\bar{\lambda}(t_0)t_0}}{n!} \quad \begin{array}{l} t_0 \geq 0 \\ n = 0, 1, \dots \end{array} \quad (2.5)$$

If this does represent the correct distribution of the number of arrivals in the interval, then  $n$  and its assumed distribution should satisfy the differential equation (2.3). Substituting (2.4) and (2.5) into (2.3) yields

$$\begin{aligned} \frac{dP_n(t_0)}{dt} &= \frac{n}{n!} \bar{\lambda}(t_0)^{n-1} t_0^n e^{-\bar{\lambda}(t_0)t_0} \frac{\partial \bar{\lambda}(t_0)}{\partial t} \\ &+ \frac{n}{n!} \bar{\lambda}(t_0)^n t_0^{n-1} e^{-\bar{\lambda}(t_0)t_0} \\ &+ \frac{1}{n!} \bar{\lambda}^n(t_0) t_0^n e^{-\bar{\lambda}(t_0)t_0} (-\bar{\lambda}(t_0) - t_0 \frac{\partial \bar{\lambda}(t_0)}{\partial t}) \end{aligned} \quad (2.6)$$

Note that

$$\begin{aligned}\frac{\partial \bar{\lambda}(t_0)}{\partial t} &= -\frac{1}{t_0^2} \int_0^{t_0} \lambda(x) dx + \frac{1}{t_0} \lambda(t_0) \\ &= \frac{1}{t_0} [\lambda(t_0) - \bar{\lambda}(t_0)]\end{aligned}\quad (2.7)$$

Substituting (2.7) into (2.6) yields,

$$\begin{aligned}\frac{dP_n(t_0)}{dt_0} &= \frac{n}{n!} \bar{\lambda}(t_0)^{n-1} t_0^{n-1} e^{-\bar{\lambda}(t_0)t_0} \left[ \lambda(t_0) - \bar{\lambda}(t_0) + \bar{\lambda}(t_0) \right. \\ &\quad \left. + \frac{\bar{\lambda}(t_0)t_0}{n} (-\lambda(t_0) - (\lambda(t) - \bar{\lambda}(t))) \right] \\ &= \frac{n}{n!} \lambda(t_0)^{n-1} t_0^{n-1} e^{-\bar{\lambda}(t_0)t_0} \left[ \lambda(t_0) - \frac{\bar{\lambda}(t_0) \lambda(t_0) t_0}{n} \right] \\ &= \lambda(t_0) \left[ \frac{(\lambda(t_0) t_0)^{n-1} e^{-\bar{\lambda}(t_0)t_0}}{(n-1)!} - \frac{(\bar{\lambda}(t_0) t_0)^n e^{-\bar{\lambda}(t_0)t_0}}{n!} \right] \\ &= \lambda(t_0) [P_{n-1}(t_0) - P_n(t_0)]\end{aligned}\quad (2.8)$$

Therefore, the assumed distribution of  $n$  is valid for the nonhomogeneous case. This is a very significant result; it means that the total number of arrivals in an interval for any Poisson process can be simulated, if the mean arrival rate in the interval is known.

Now, consider the interval  $(t_i, t_j)$ . Recall that the proposed algorithm calls for a pdf to be constructed from the function of  $\lambda(t)$

in this interval. This is done by normalizing the area under the curve to one and then shifting the curve so it begins at zero (Figure 2.4). We will define  $x_0 = t_0 - t_i$ , and  $A \equiv \int_{t_i}^{t_j} \lambda(t) dt$ . The probability that a given arrival occurs in the interval  $(t_0, t_0 + dt)$ , corresponds to the area of the strip  $(x_0, x_0 + dx)$  which equals  $f_X(x_0) = \lambda(t_0)/A dt$ . Since there are  $n$  independent arrivals in this interval, the probability of any one arrival occurring in  $(t_0, t_0 + dt)$  is:

$$\Pr(\text{arrival in } (t_0, t_0 + dt)) = (n\lambda(t_0)/A)dt$$

This probability can be found as follows:

$$\begin{aligned} \Pr(\text{arrival in } (t_0, t_0 + dt)) &= \sum_{n_0=0}^{\infty} P_N(n_0) \Pr \left( \begin{array}{l} \text{arrival in } (t_0, t_0 + dt), \\ \text{given there are } n_0 \\ \text{arrivals in the} \\ \text{entire interval } (0, t) \end{array} \right) \\ &= \sum_{n_0=0}^{\infty} P_N(n_0) \cdot \frac{n_0 \lambda(t)}{A} dt \\ &= \frac{\lambda(t)}{A} dt \sum_{n_0=0}^{\infty} n_0 P_n(n_0) \\ &= \frac{\lambda(t)}{A} dt E(n) \end{aligned} \quad (2.9)$$

It was just shown that  $n \sim P(\bar{\lambda})$ , so the expected value of  $n$  is as follows.

$$\begin{aligned} E(n) &= \bar{\lambda}(t_0)(t_j - t_i) \\ &= (t_j - t_i) \left[ \frac{1}{(t_j - t_i)} \int_{t_i}^{t_j} \lambda(x) dx \right] \\ &= A \quad \text{substitution into 2.9 yields,} \end{aligned} \quad (2.10)$$

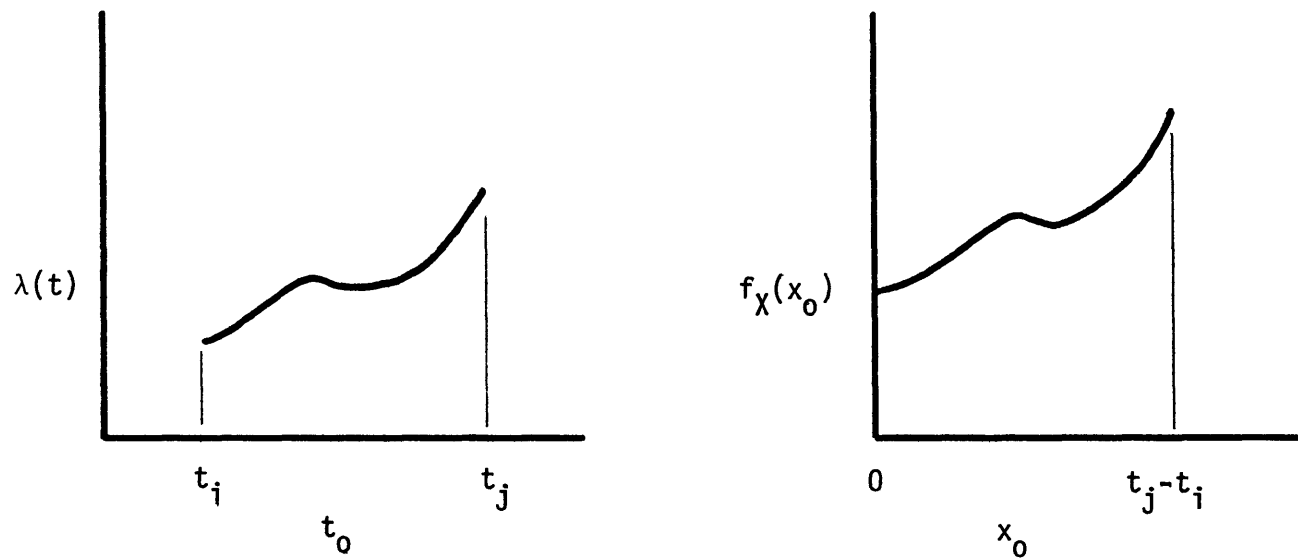


Figure 2.4 - The Construction of the PDF From the Arrival Rate in an Interval

$$\Pr[\text{arrival in } (t_0, t_0 + dt)] = \lambda(t_0)dt \quad (2.11)$$

Recall that for each passenger arrival, a random deviate is calculated using the probability density integral transform method. These are independent, identically distributed random variables, therefore arrivals are independent.

In summary, the algorithm generates independent arrivals, and the probability of an arrival in a given interval  $(t_0, t_0 + dt)$  is  $\lambda(t_0)dt$ . These are the basic characteristics of a Poisson arrival process. Hence, the algorithm in SMUSH does generate Poisson arrivals.

The average arrival rate for each OD passenger type is input at discrete points in time. SMUSH assumes the arrival rate to be linear between these points (Figure 2.5). When the passenger generator is called, the passengers arriving in the next half hour are generated. While this half hour may contain several data points, and several linear segments, the generation algorithm is applied to each linear segment separately. So,  $\lambda(t)$  in the interval is linear.

During simulation, the actual number of arrivals is determined by generating exponentially distributed (with parameter  $\bar{\lambda}$ ) interarrival times,  $R_1$ . The value of  $n$  is determined as the largest integer satisfying the following.

$$\sum_1^n R_1 < t_j - t_i$$

It is now necessary to determine the placement in  $I_{ij}$  of these  $n$  arrivals. The following pdf is constructed from the curve of  $\lambda(t)$  in  $I_{ij}$  (see figure 2.6).

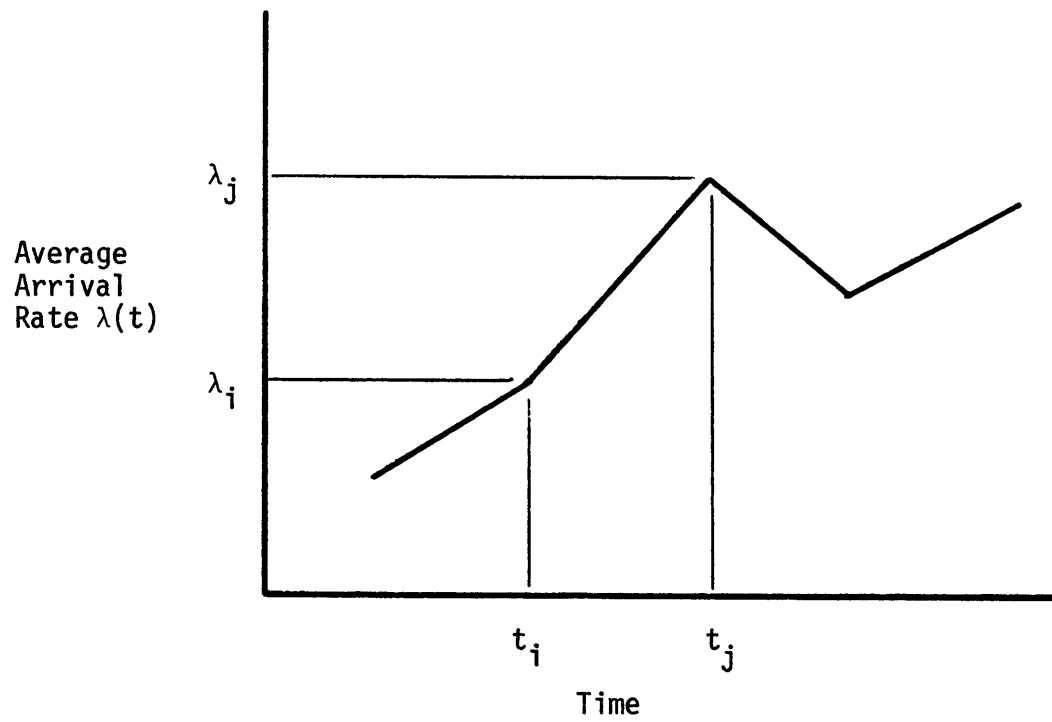


Figure 2.5 - Piecewise Linear Arrival Rate Used in SMUSH

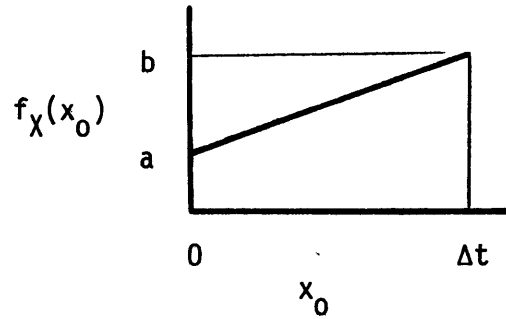


Figure 2.6 - The PDF Constructed from  $I_{ij}$

where  $a = \lambda_i / (\Delta t \bar{\lambda})$ ,  $b = \lambda_j / (\Delta t \bar{\lambda})$ , and where  $\Delta t = (t_j - t_i)$ .

$$f_X(x_0) = \begin{cases} a + \frac{b-a}{\Delta t} x_0, & 0 < x_0 < \Delta t \\ 0 & \text{elsewhere} \end{cases}$$

To generate random deviates according to this pdf, the probability-integral-transform method is used.

1) Determine cumulative distribution of  $X_0$ ,  $F_X(x_0)$

$$F_X(x_0) = \int_0^{x_0} f_X(t_0) dt_0 = \int_0^{x_0} a + \left(\frac{b-a}{\Delta t}\right) t_0 dt_0 \quad (2.13)$$

$$F_X(x_0) = \begin{cases} 0, & x_0 < 0 \\ ax_0 + \frac{b-a}{2\Delta t} x_0^2, & 0 \leq x_0 < \Delta t \\ 1 & x_0 \geq \Delta t \end{cases} \quad (2.14)$$

2) Set  $R_2 \sim U(0, 1)$  equal to  $F_X(x_0)$ , solve for  $x_0$ .

$$R_2 = ax_0 + (b - a)/2 \Delta t x_0^2 \quad (2.15)$$

$$\text{If } a = b \rightarrow R_2 = ax_0$$

$$x_0 = R_2/a = \Delta t R_2$$

If  $a \neq b$ , substitute for  $a$  &  $b$ ,

$$R_2 = \frac{2\lambda_i}{\Delta t(\lambda_i + \lambda_j)} x_0 + \frac{2(\lambda_j - \lambda_i)}{\Delta t(\lambda_j + \lambda_i)} \frac{x_0^2}{2\Delta t}$$

$$R_2 = \frac{1}{\lambda_i + \lambda_j} \left[ \frac{x_0}{\Delta t} 2\lambda_i + \left(\frac{x_0}{\Delta t}\right)^2 (\lambda_j - \lambda_i) \right] \quad (2.16)$$

Solving (2.16) for  $x_0$  yields:

$$x_0 = \frac{\Delta t}{\lambda_j - \lambda_i} \left[ \sqrt{\lambda_i^2 (1 - R_2) + \lambda_j^2 R_2} - \lambda_i \right] \quad (2.17)$$

For each of the  $n$  arriving passengers in this interval,  $R_2$  is drawn and  $x_0$  is calculated. The arrival time of an individual passenger is then,

$$\text{Time of arrival} = t_a = t_i + x_0$$

When the passenger generator is called, this algorithm is carried out for each linear segment between  $T_{\text{now}}$  and  $T_{\text{now}} + .5$ , for every OD pair. Another passenger generation event is then stored to occur at  $T_{\text{now}} + .5$ .



### 2.2.3 Vehicle Simulation

SMUSH controls the vehicles with a simple routing scheme. Every vehicle is assigned to a specific route. Each route has a successor route. When a vehicle completes its assigned route, it immediately begins the completed route's successor. To invoke a fixed control, all routes and their successors are specified, and each vehicle is assigned an initial route, before the simulation begins. In this manner, the complete routing for each vehicle is determined. If dynamic controls are invoked, SMUSH constructs routes and determines their successors for each vehicle during simulation.

The actual control of vehicle movements is also straightforward. Basically, a vehicle departure causes a future vehicle arrival event to be stored, and vice versa.

During system initialization, SMUSH stores vehicle departure information on File 1. The attributes associated with each departure event are as follows:

- Event code
- Vehicle number
- Route number
- Stop index (this is the stop number on the route, first, second...)
- Time of occurrence

A vehicle departure event invokes the following:

- Search File 2 for passengers to load
- Transfer from File 2 to 3 those passengers which meet boarding requirements
- Store an arrival event at the next stop on the route

The attributes associated with a vehicle arrival event are the same as those for the vehicle departure. A vehicle arrival invokes the following:

- Search File 3 for passengers to unload
- Remove passengers meeting the unloading conditions
- If this is the last stop on the route, determine next route
- Store a departure event.

The last two items carried out during a vehicle arrival (determine next route, and schedule a departure) are simply a restatement of the vehicle control strategy. Although the logic to make these routing decisions may be complex, its addition to SMUSH is simple.

## Chapter 3

SIMULATION INVESTIGATION OF USH PERFORMANCE

Completion of the simulation model marks the end of the preliminary work. We can now begin the actual evaluation of various vehicle routing and scheduling strategies. This evaluation, based on system performance, is carried out in two phases. First, the system operating under simple fixed routing strategies is simulated. Next, the system is simulated while operating under dynamic controls.

### 3.1 The Operating Scenario

The operating scenario is based on the Long Island area. The population density and the presence of natural barriers to surface transportation make this area conducive to successful USH operations. Mann's study provides an estimate of daily demand volume and patterns from this area. The overall characteristics of the testing scenario are as follows:

1. Total demand is 4000 one-way trips per day
2. The ground facilities consist of 6 terminals, 2 located in the CBD, and 4 in the suburbs
3. The general layout is as shown in Figure 3.1.

#### 3.1.1 Heliport Location

In an unconstrained system, the heliports would be located such that the expected access/egress time for passengers is minimized. But heliports are not the best of neighbors (although they could be

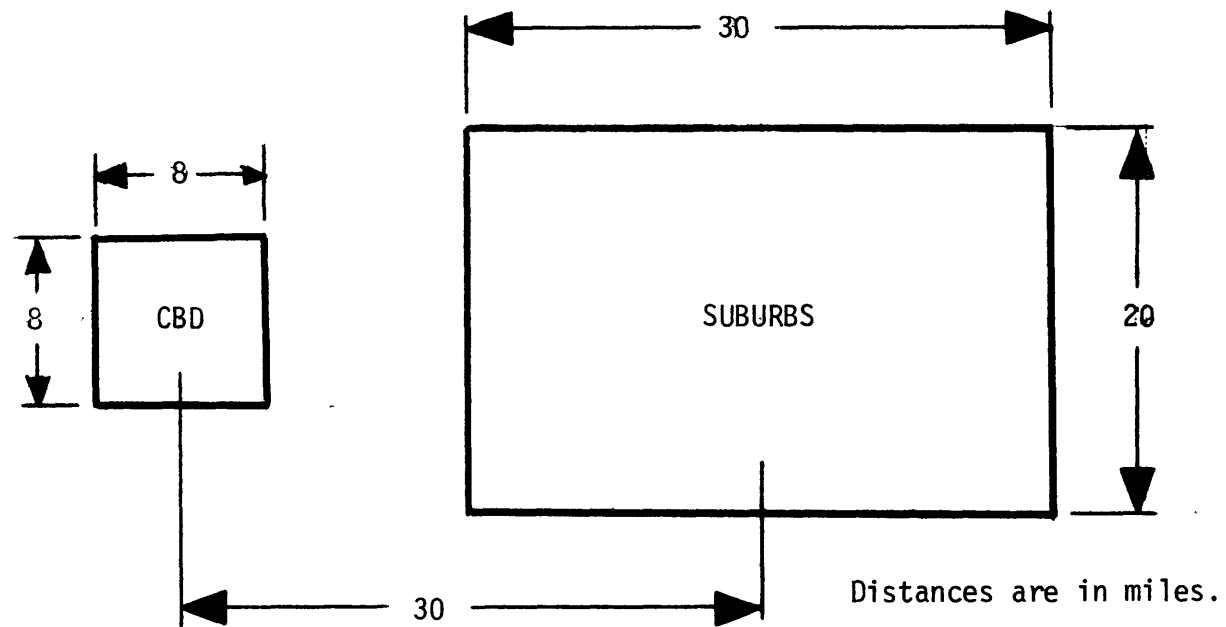


Figure 3.1 - Study Network Layout

made so at a cost<sup>8</sup>). One would expect restrictions to be placed on terminal location due to noise and safety considerations. In general, the terminals would be located as follows. First, each area is divided into single terminal regions. Then, in each region, the terminal is located according to local restrictions (independently of all other terminals). With no prior knowledge of local restrictions, there is considerable uncertainty as to the location of each terminal. In fact, the location may be considered a random vector, distributed uniformly over the region. This is the approach used to simulate the location of the ground facilities in this study, see Figure 3.2.

### 3.1.2 The Demand

The market for USH operations is made up of the commuter and the general travel group. Mann estimated that the total trip volume for each group is equal. (From each group, there are 2000 one-way trips per 12 hour day). During morning operations, all commuter trips are made from the suburbs to the CBD, and occur between 6:30 and 9:30. The total magnitude of commuter traffic versus time of day is shown in Figure 3.3a. (Since morning and evening operations are roughly symmetric, only the period from 6 a.m. to 12 a.m. is simulated.) The trips generated by the general travel group are distributed evenly over time. There are three types of trips taken by this group, suburbs to CBD, CBD to suburbs, and suburbs to suburbs. The total number of trips of each type is equal to 1/3 of the total. The rate at which trips of each type is generated is about 60 per hour (see Figure 3.3b). A schematic view of the daily demand is shown in Figure 3.4a and b.

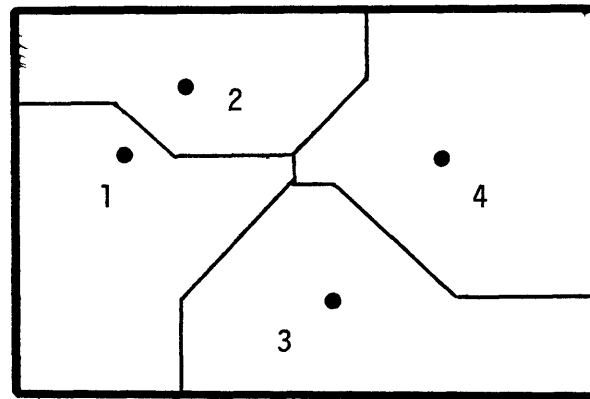
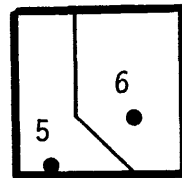
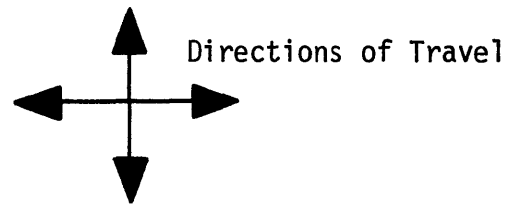
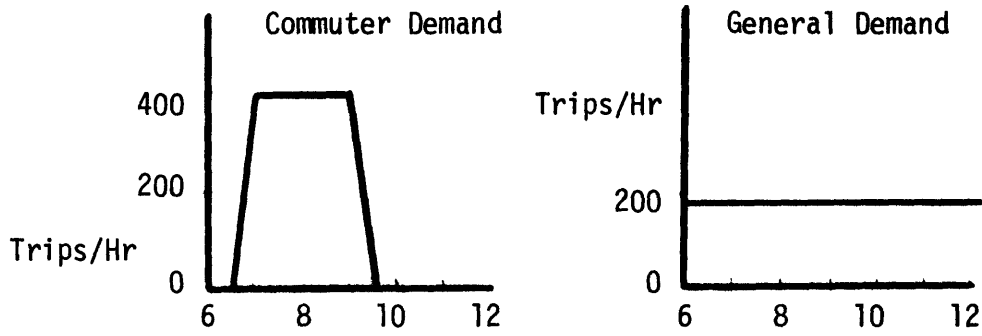
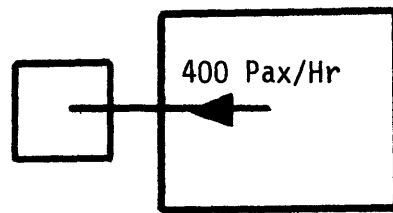


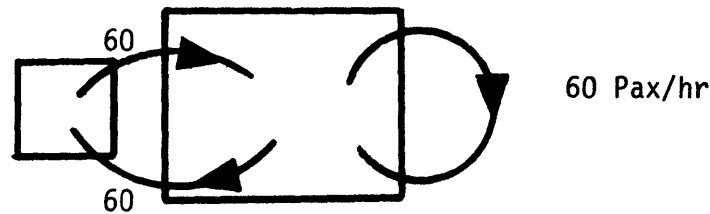
Figure 3.2 - Station Location and Catchment Areas



Figures 3.3 a,b - Time Variations of Demand



Commuter Demand Pattern During the Morning Peak



General Demand Pattern

Figure 3.4c - Spatial Demand Patterns

If the demand density (trips/mile<sup>2</sup>), for each trip type, is assumed to be uniform throughout the suburbs and also CBD, and if all passengers enter (leave) the USH system at the station closest to their origin (destination), then the actual demand rate for each OD pair (origin and destination pair) is proportional to the catchment areas of the origin and destination stations. These assumptions allow the demand rates to be broken down into OD demand data, necessary for the simulation. The actual OD demand rates are shown below, in Table 3.1.

AVERAGE ARRIVAL RATES TRIPS/HR													
Origin/ Destina- tion	Commuters 7-9 a.m.						General						
	1	2	3	4	5	6	1	2	3	4	5	6	
	1	0.	0.	0.	0.	29.	71.	1	0.	3.	4.	6.	4.
2	0.	0.	0.	0.	22.	54.	2	3.	0.	3.	4.	3.	8.
3	0.	0.	0.	0.	31.	77.	3	4.	3.	0.	6.	4.	11.
4	0.	0.	0.	0.	34.	82.	4	6.	4.	6.	0.	5.	12.
5	0.	0.	0.	0.	0.	0.	5	4.	3.	4.	5.	0.	0.
6	0.	0.	0.	0.	0.	0.	6	10.	8.	11.	12.	0.	0.

Table 3.1

The actual arrival of passengers follows a Poisson process. This process is simulated using the algorithm discussed in 2.2.2 in conjunction with a pseudo random number generator. The pseudo random number generator calculates a string of numbers recursively and although these numbers appear random, they are not. This same string can be reproduced exactly, if desired. So, while the simulated arrivals of passengers approximate a Poisson process, it can be reproduced also.. In this study, the same arrival process is used for each simulation



run. By exploiting this reproducibility, all control algorithms are compared on the basis of operations during a single day. Of course, there are random fluctuations in the performance of a single day. But since the purpose of the study is the investigation of general trends rather than precise performance, the results based on a single day of operation provide sufficient accuracy.

### 3.1.1 The Vehicles

The aircraft type used in this study is typical of the next generation of helicopters. It is similar to the tandem rotor vehicle which is described in Appendix C. The vehicle characteristics are as follows:

Capacity: 50 seats

Cruise speed: 100 m/hr.

DOC: 600\$/flight hour

100\$/cycle

Take-off/landing cycle time:

Max: 9 minutes

Min: 3 minutes

The take-off and landing cycle time is comprised of the time required for deceleration and descent from cruise, ground maneuvers, ascent and acceleration to cruise. The minimum cycle time corresponds to the time required for the air maneuvers. The difference between minimum and maximum time represents the time to load the vehicle to capacity. It is assumed that this time is a linear function of the number of

passengers loaded (see Figure 3.5).

$$\text{Cycle time} = \text{min} + (\text{max} - \text{min}) \frac{\text{number loaded}}{\text{capacity}}$$

So, for this vehicle, cycle time is found as follows:

$$\text{Cycle time} = 3 + 6 (\text{number loaded})/50 \text{ (minutes)}$$

### 3.2 Fixed Control Strategies

Two general classes of fixed vehicle routing strategies are considered in this section, these are allstop and nonstop. The allstop strategy is similar to that used by bus and subway lines. On a single trip through the network, each vehicle stops at every station, in a prescribed order. The nonstop strategy is a point-to-point routing. There is no mechanism by which the vehicle routing can adjust to meet changes in the system conditions (varying demand patterns, or congestion). These are very simple strategies, since no decisions are made during system operations, and no system information is necessary.

These two strategies provide an estimate of system performance in two extreme service situations. Each OD pair is served with a maximum frequency in the allstop case, but the average time required to complete service (ride time) is also a maximum. In the nonstop case, both frequency of service and average ride time are at a minimum. This is an interesting trade-off to be investigated.

The effect of demand peaking on the performance of the system, operating under each strategy, is also investigated. Demand peaking

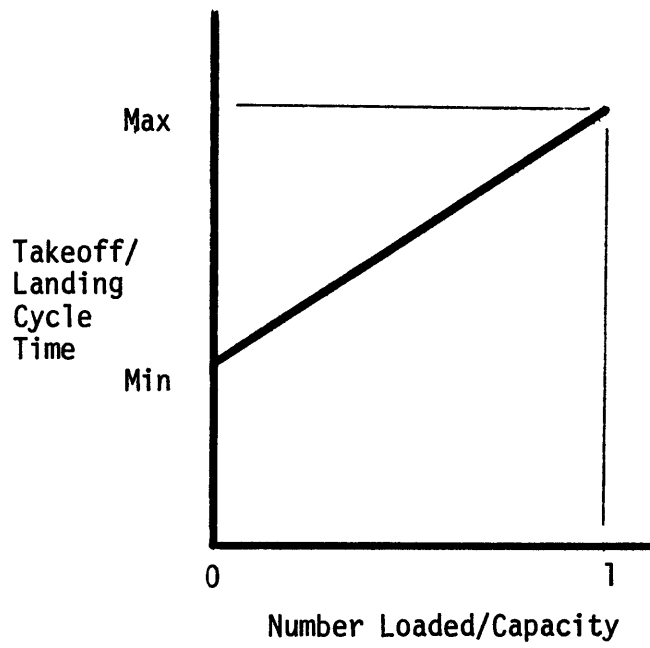


Figure 3.5 - Takeoff and Landing Cycle Time

causes a reduction in the probability that a given passenger will be able to board the next departure from her origin, which leads to congestion. The system can be designed for the peak demand period such that congestion never occurs, but this is an expensive way to go. A trade-off to be investigated is, how much congestion is tolerable for a given system and by what amount is operating cost reduced by allowing this amount of congestion. Also, are there any simple controls, within the guidelines of this section, which can reduce the effects of congestion on LOS?

### 3.2.1 Allstop Routing

The allstop routing strategy is similar to that used in uncontrolled linear networks (bus and subway systems). All vehicles travel the same route, and on this route stop at every station. In this study, the route is chosen such that all stations are visited with a minimum total distance traveled (the traveling salesman route). The routing is shown in Figure 3.6.

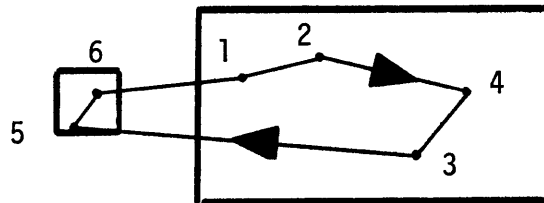


Figure 3.6 - Allstop Vehicle Routing

To avoid congestion in this network, the flow of seats available must be greater than or equal to the demand flow on each arc.

In this system, the most heavily traveled arc is obviously 35 (the arc connecting nodes 3 and 5). During the morning rush, all the commuters must travel across this arc. The maximum average commuter flow is 400 trips/hr, the average general traffic brings the total expected flow to 483 trips/hr. A vehicle can complete one trip around the network in about 1.5 hrs, and each vehicle has a capacity of 50 passengers. Therefore, the contribution of each aircraft to system capacity is around 33 seats/hr across each arc. So, to avoid congestion, the number of vehicles needed is  $483/33 \approx 15$ .

The morning operation of this USH system is simulated using the allstop routing with 15 vehicles. The average systemwide travel time is about 1 hour, which is pretty mediocre for such an expensive system. By referring to the load factor histogram (Figure 3.7), it can be seen that there is some congestion, as evidenced by the 60 full flights. As expected, these are departures from 3 during the peak hours. So, there was no serious congestion problem. A most shocking result though is that nearly 200 (of 400) flights are made empty. This is due to the effect of vehicle clumping.

Clumping is a phenomenon demonstrated daily by uncontrolled linear networks. Clumping results from a tendency for headways which are shortened by perturbations to the system to become shorter still, until the headway goes to zero, and the vehicles travel together in a clump. The mechanism which makes the system of headways unstable is the dependence of cycle time on the number of passengers loaded and

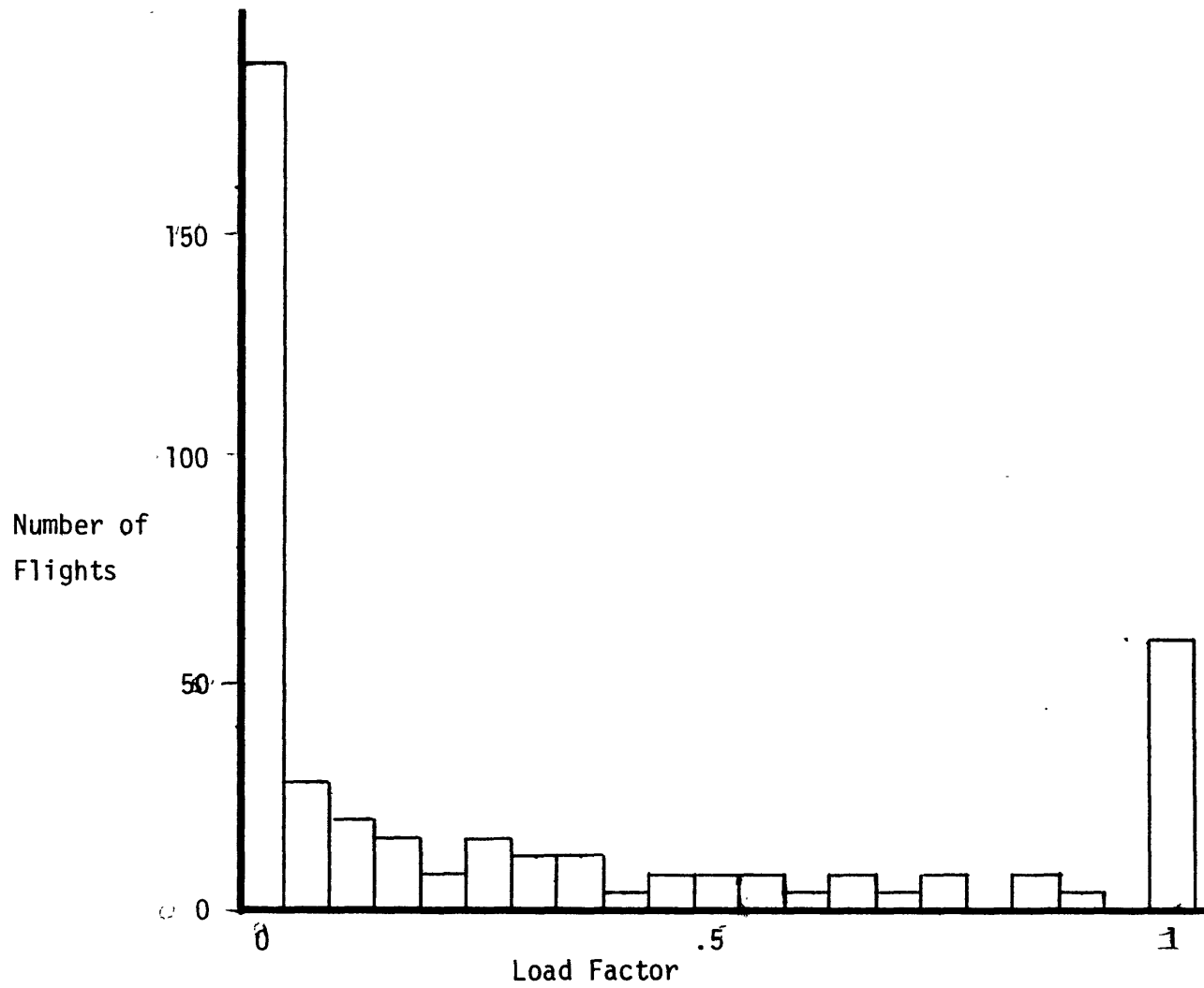


Figure 3.7 - Load Factor Distribution, Allstop Routing, 15 Vehicles, No Controls

unloaded. In an actual system, for example, a bus line, vehicles are spaced evenly at the start of a run, randomness in traffic conditions provides the perturbations to begin clumping. The evenly spaced vehicles tend to form pairs, these pairs form quads, etc... Although vehicles in the USH system are not affected by traffic conditions, the randomness in passenger arrivals, and hence cycle time, produces the perturbations necessary to start the clumping process. Figure 3.8 illustrates what happens to vehicle spacing in the 15 aircraft case. The inner circle indicates the location of the vehicles (small circles) in the network. The light and shaded regions represent ground and flight time, respectively. Initially, the vehicles are spaced evenly around the network. After 4 hours of operation, 4 clumps have formed, and after 8 hours, all the vehicles are in two massive clumps of whirling rotors. Before making any rash judgments to the effect that helicopters are more prone to clumping than buses, it should be remembered that these vehicles ran uncontrolled for 8 hours, where buses run for about 30 minutes before they are respaced.

This clumping has a pronounced effect on LOS over the course of the day. Initially, the headways are all .1 hours, thus the average passenger wait is .05. By the end of the run, the average headway is .75 hours, and the average wait is now .37. This represents a 30% increase in average trip time.

The effects of clumping and congestion on LOS are illustrated in the Figure 3.9. Average travel time increases gradually during the

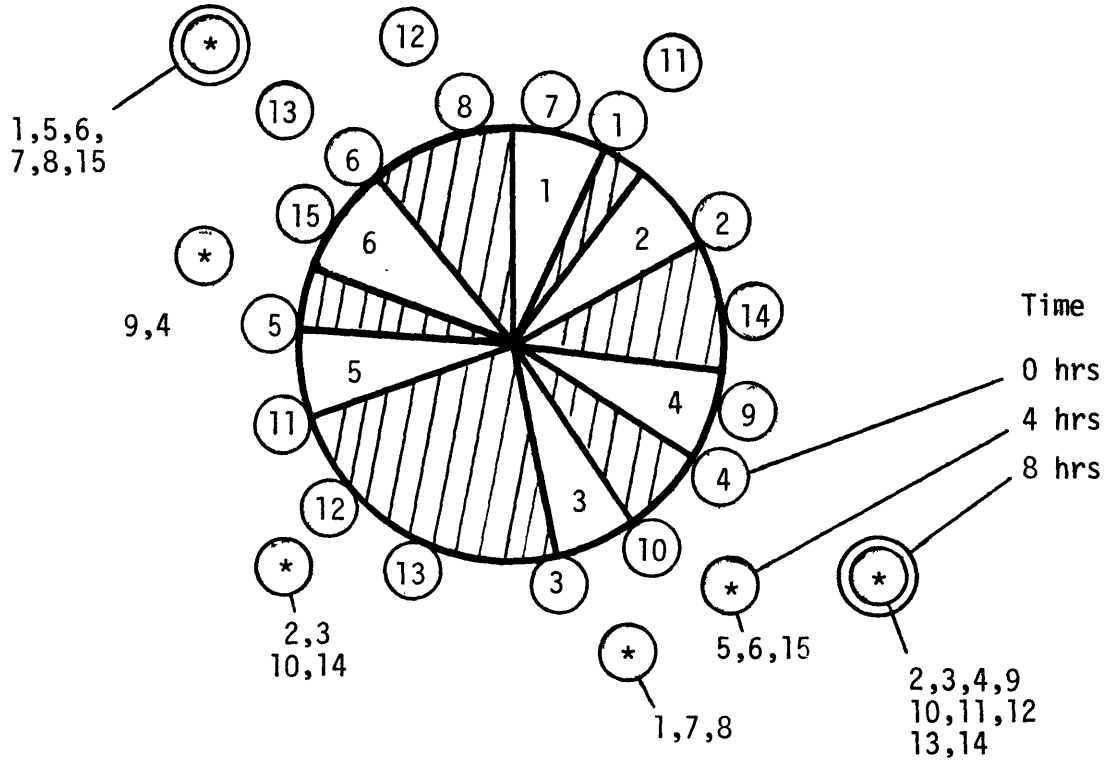


Figure 3.8 - Clumping in the 15 Vehicle Case,  
With no Minimum Headway Controls



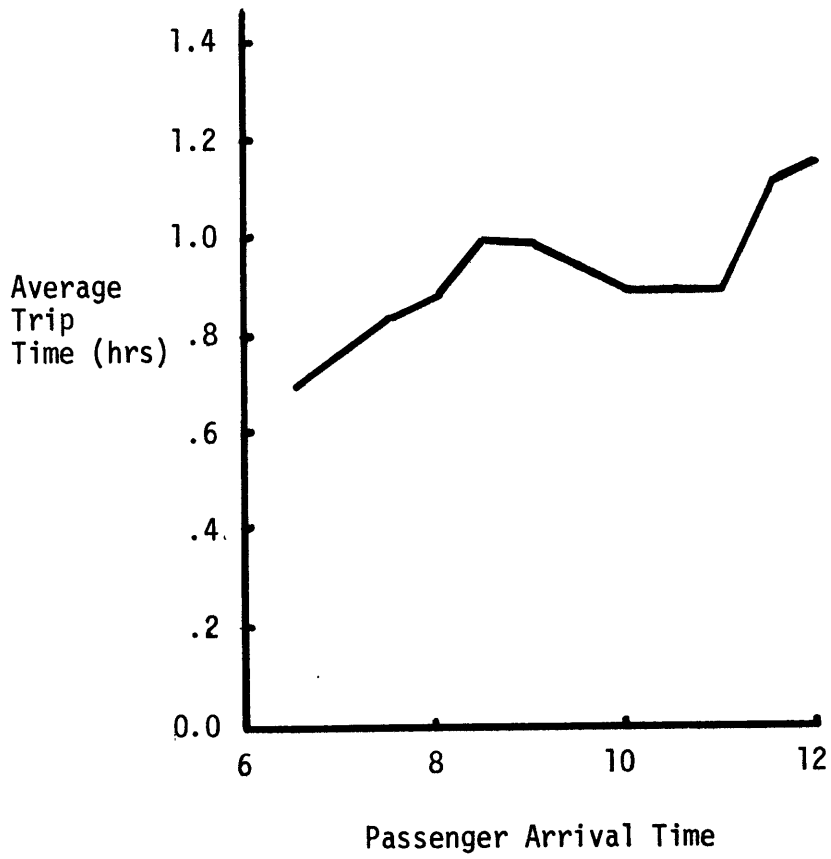


Figure 3.9 - Variation of LOS With Time of Day,  
15 Vehicle, Allstop Routing

rush, due to congestion, and then drops back off, at the end of the rush. At around 11:00, clumping shows its effect, by driving average time back up by about .25 hours.

Another result of clumping is a tremendous waste of vehicle resources. After waiting 20 minutes for a bus, it is not uncommon to have a caravan of 4 buses arrive, in which the first bus is full, and the trailing 3 are empty. This effect also appears in the already under utilized USH system. The LF Histogram, Figure 3.10, illustrates this effect. After 4 hours into the run, this particular vehicle did not pick up a single passenger. Unfortunately for the USH system, this is the rule, not the exception. In fact, by the end of the 8 hour run, 2 vehicles (one in each clump) are picking up what few passengers there are, at each stop. The problem of clumping and some possible controls is addressed in a later section.

Having witnessed the mediocre service in the 15 vehicle case, its time to plunge into the realm of the poor and terrible service as the system is simulated with 10 and then 6 vehicles. This illustrates the horrendous effects of congestion on LOS, and the system's inability to recover from overloading. At first glance, the results of the simulation for these cases do not look so bad; average trip times are 1.37 and 1.65 hours for the 10 and 6 vehicle runs, respectively. But these average values belie the terrible service provided to passengers in both cases. Shortly after the rush begins, the 35 arc becomes congested. This congestion then starts to spread back upstream

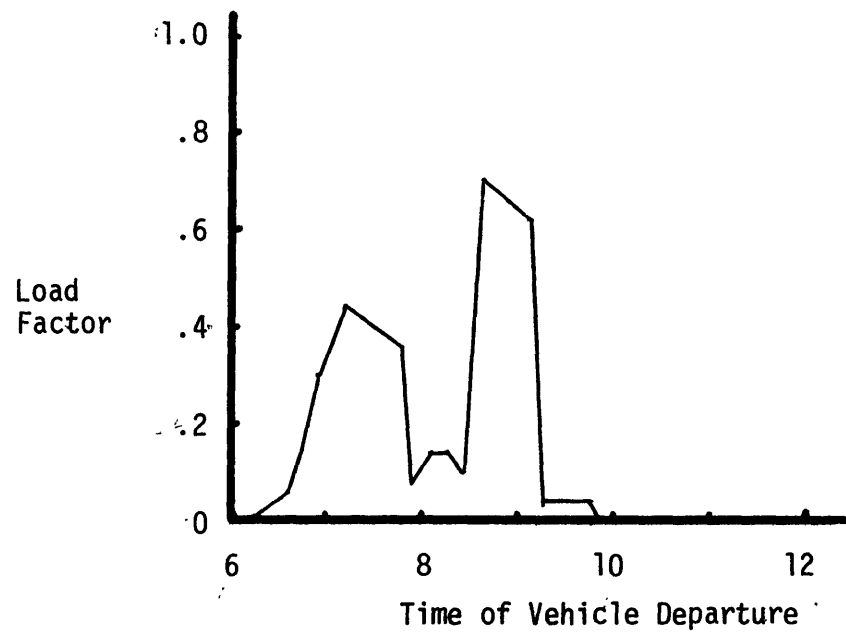


Figure 3.10 - The Variation of Load Factor for a Vehicle During a 15 Vehicle, Allstop Routing Case With No Cont

through the suburbs to station 2. This means that not only are the flights from the suburbs to the CBD full, but the vehicles arriving at 3 and 4 are full also. This creates tremendous passenger queues at 3 and 4, since the only seats available to passengers entering the system are created by passengers leaving at these stations. This leads to the following observation. When the allstop system is running at capacity into a station, there is a conservation of flow at this station. The number of passengers allowed to board is equal to the number leaving at this station. During the rush, severe congestion occurs due to the strong directionality of the demand. For example, between 7:00 and 9:00, station 4 is the origin for 150 passengers per hour, but 4 is the destination for only 30 passengers per hour. So, when vehicles arrive at 4 full, passengers back up at a rate of 120 per hour. The result of this effect is illustrated in the figure 3.11. There are a few observations to be made from this figure. First, in the 6 vehicle case, there are still 600 passengers waiting at the end of the run. Because of this, the actual system average trip time should be adjusted upward; a reasonable estimate for the trip time is 2.3 hours. In this same case, the number waiting at both 3 and 4 is nearly equal to the total number of passengers which have arrived so far. This indicates that the system became congested at these stations almost immediately after the rush began. Notice that in the 10 vehicle case, the congestion did not reach 4 until 8:30 and ended at 9:30, while at 3, the congestion is as bad in the 10 vehicle case as in the 6. This illustrates the progressive nature of the congestion (recall that 4 is upstream of 3). Even more critical

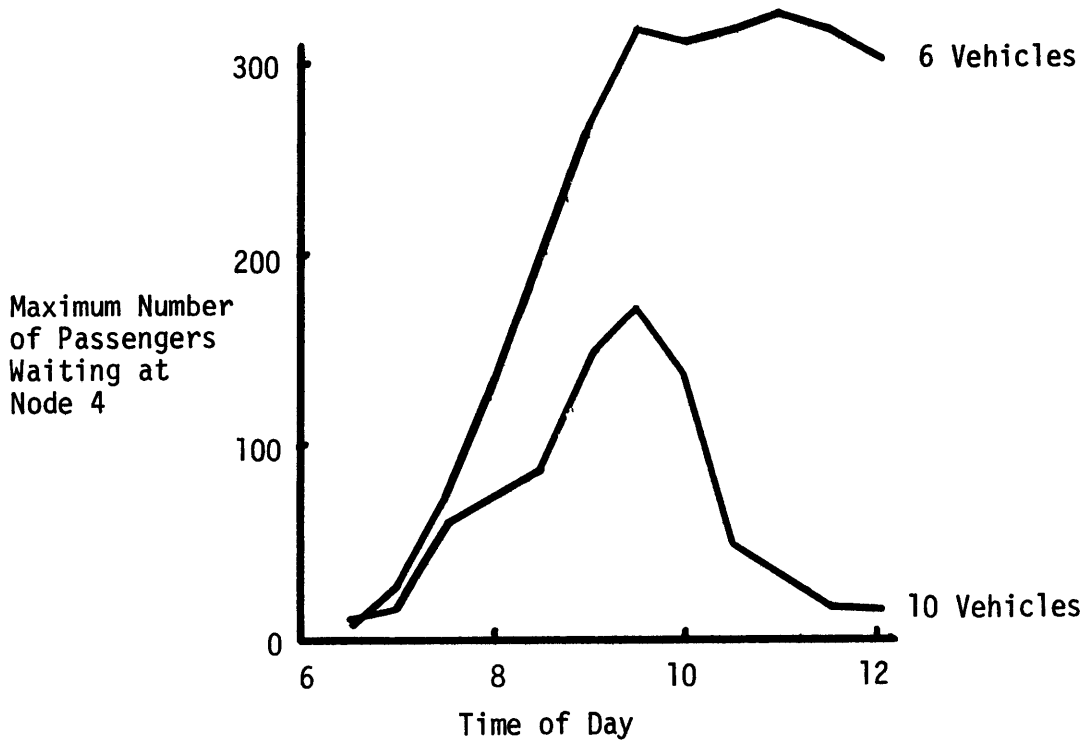
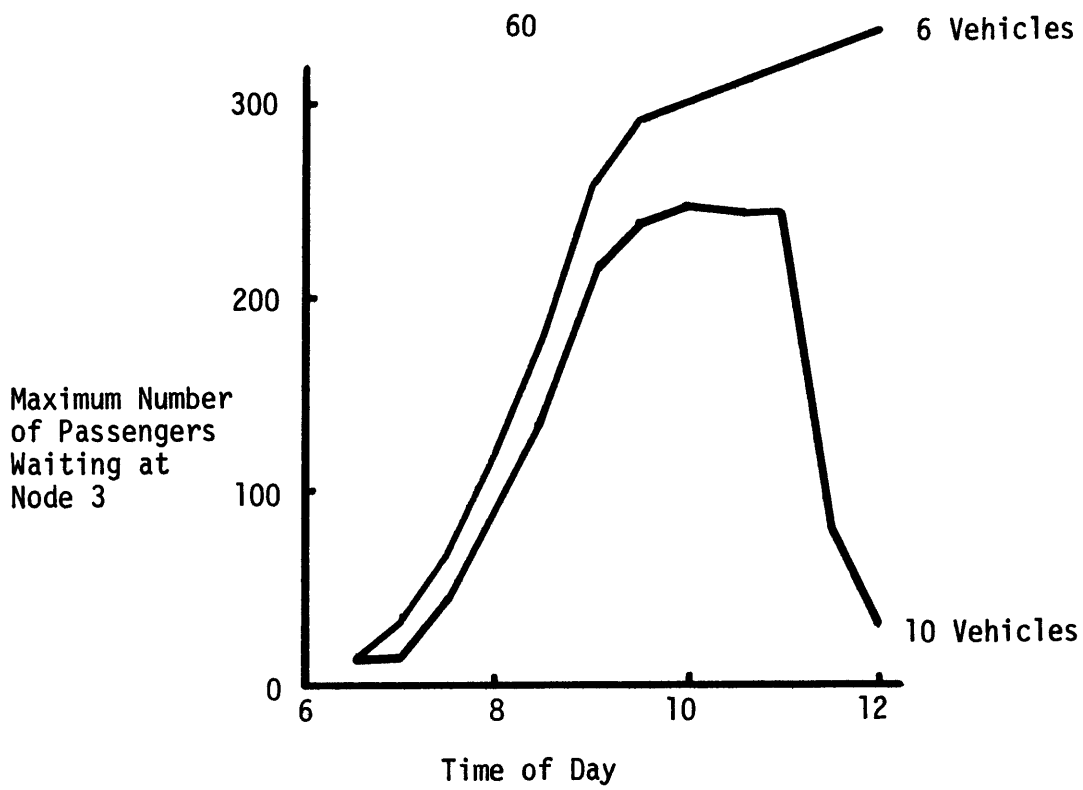


Figure 3.11 - Maximum Number Waiting, 6 and 10 Vehicles, Allstop

is the progressive manner in which congestion clears, moving back downstream. Notice that the number waiting at station 4 at 11:00 (10 vehicle case) has just gotten below 50 (vehicle capacity). At this time, there is a dramatic clearing at node 3, since vehicles are no longer arriving full. The progressive nature of the buildup and clearing of congestion creates an inequity in LOS throughout the network. While passengers are served FIFO at each station, the passengers at downstream stations are served only after all the upstream passengers are served. This effect can be seen by comparing the average trip times for up and downstream OD pairs (see Table 3.2).

OD Pair	AVERAGE TRIP TIME (hrs)	
	10 Vehicles	15 Vehicles
1-6	1.44	1.44
2-6	1.33	1.29
3-6	2.17	0.82
4-6	1.44	1.06

Table 3.2

Obviously, this is no way to run an airline. Service is being eroded by congestion and clumping. Simple solutions to these problems are investigated in the following section.

### 3.2.1.1 Simple Controls for Allstop Routing

A dispatching rule which provides a minimum spacing between

the vehicles is used to reduce the effects of clumping. To implement this control, the time of the last departure from each station is recorded. When a new departure is scheduled, the headway between this and the most recent departure is determined. If this proposed headway is greater than the minimum, the departure is made as scheduled. If the headway is less than the minimum, the vehicle is held until the minimum headway is achieved.

Congestion is controlled by limiting the number of commuters boarded at stations 1, 2 and 4. This limits the tendency for vehicles to arrive at 3 and 4 full, during the rush period. Ideally, only the 35 arc will run at capacity, and the number of each commuter type on the arc should be proportional to the total number requesting service. This policy will control the crowding at 3 and 4, and should provide a more equitable LOS to passengers in the suburbs.

The system is simulated using these controls for the 15 and 10 vehicle cases (the 6 vehicle case is dropped since LOS is so poor). The system-wide trip times are shown below in Table 3.3

Number of Vehicles	10	15
Average Trip Time	1.47	0.97

Table 3.3

These results are somewhat underwhelming. LOS seems to be degraded in the 10 vehicle case, and there seems to be no difference in the 15 vehicle case. But there are positive effects of these controls. In the 15 vehicle

case, there is an obvious (but insignificant) change in the distribution of load factor (see Figure 3.12). The total number of empty flights is reduced to 40, from 190 in the uncontrolled case. This and the fact that the number of 2 passenger flights increased are evidence that the minimum headway rule is controlling clumping. The minimum headway was set at .05 hrs. While this eliminates vehicle clumps, it still does not provide even spacing. At the end of 8 hours, the vehicles travel in a large bunch, and the headway distribution is as follows:

$$h = \begin{cases} .05 \text{ hr, with probability } 14/15 \\ .75 \text{ hr, with probability } 1/15 \end{cases}$$

The resulting average wait is .2 hours. Clumping is still taking its toll, but not as badly as in the uncontrolled case where the average wait is .37 hours. The effect of the controls are shown in Figure 3.13. By the end of the simulation, there is a significant difference in LOS between the controlled and uncontrolled cases. The effects of the inequity of service of the controls is shown in Table 3.4. As expected, the quota has little effect on the 15 vehicle case, since congestion is not a problem here. But in the 10 vehicle case, the quota works as planned. The difference in LOS for the various OD pairs is reduced, but the price of this smoothing is increased systemwide trip times.



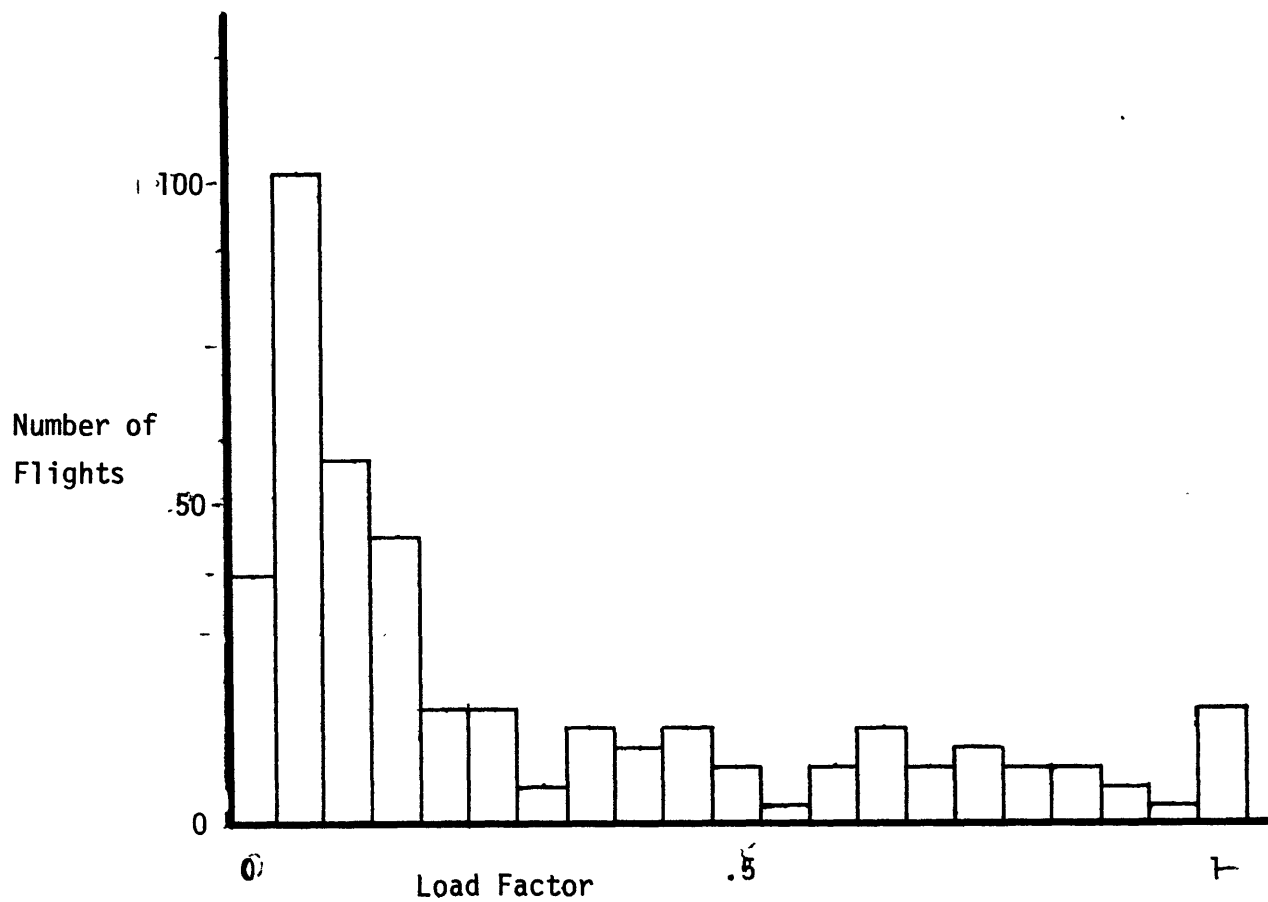


Figure 3.12 - Load Factor Distribution, 15 Vehicles, Allstop Routing With Controls

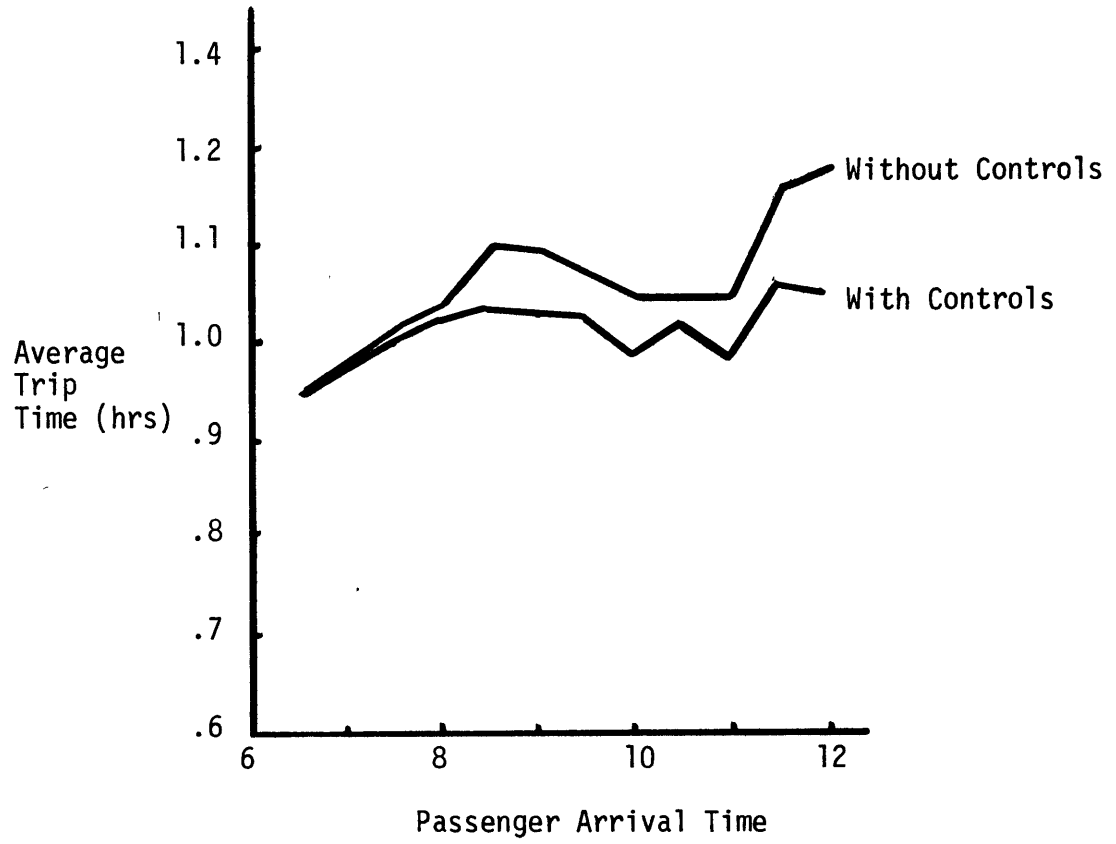


Figure 3.13- LOS versus Time of Day, 15 Vehicles,  
Allstop Routing with and without Controls

## AVERAGE TRIP TIME (hrs)

OD Pair	10 Vehicles	15 Vehicles
1-6	1.89	1.41
2-6	1.95	1.25
3-6	1.80	0.86
4-6	1.64	0.99

Table 3.4

The performance of the USH system using the allstop routing scheme is poor. The system's sensitivity to demand peaking is illustrated in the Figure 3.14. With 15 vehicles, equilibrium during the peak loading is achieved, average trip time leveled out during the rush. In the 10 vehicle case, this does not happen. Recall that passengers queued up at station 3 (at a rate of 80 per hour) during the rush. These passengers are cleared out only after the rush ended. Because of this congestion, the average travel time more than doubled for the passengers entering during the rush. In the 6 vehicle case, not only do passengers back up during the rush, they're still waiting at noon. The reason that the observed average trip time drops after the peak at 8:00 is that only the lucky passengers who received service are counted.

Another problem with the allstop strategy is the interdependence of service between different OD pairs. This effect makes it difficult to improve the service to one OD pair without degrading service to others.

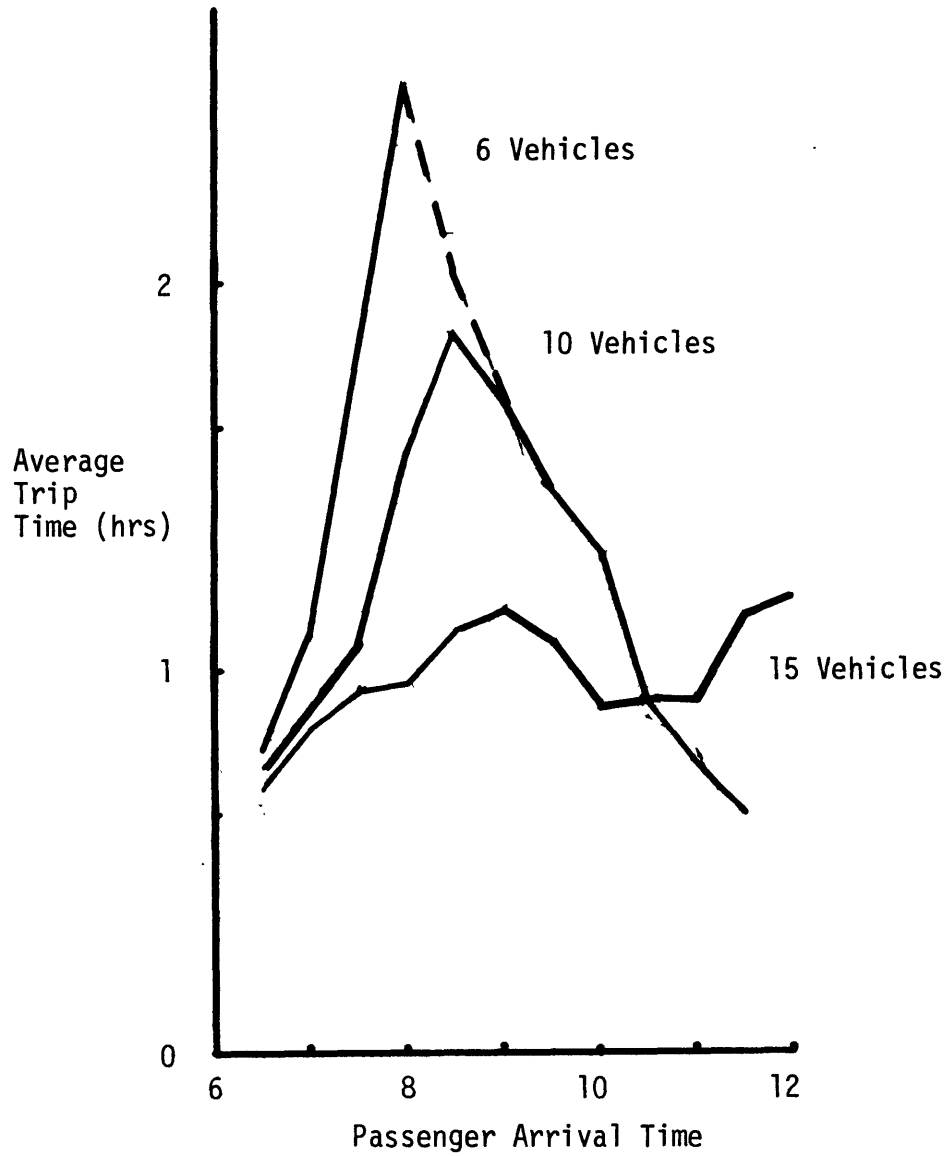


Figure 3.14 - LOS vs. Time of Day, Allstop Routing with No Controls

The last point to be made is that this service is expensive. The variation of average travel time with operating cost is shown in Figure 3.15.

### 3.2.2 Nonstop Routing

We now consider a simple nonstop strategy. As in the allstop case, vehicle routing is fixed. Under the nonstop strategy, individual vehicles are assigned separate routes, and serve only a few pairs (under allstop controls, each vehicle serves all OD pairs). For example, a vehicle may be assigned the route 6-2-3-6 which it flies continuously. OD pairs 62, 23 and 36 are served exclusively by this vehicle.

Nonstop routing has several advantages over the allstop strategy. Since each OD pair is served by only one vehicle flying a closed loop, and the time required to complete a loop is nearly constant throughout the day, even headways are maintained. Also, only one OD pair is served at a time, so when a vehicle lands, everybody gets off, which insures the availability of 50 empty seats on each departure. Thus, the problems of clumping and progressive congestion are avoided.

In the test scenario, there are 28 OD pairs requiring service. To simulate the operation of the system using the nonstop strategy, routing patterns are determined such that all OD pairs are served with reasonable frequency. This is done by assigning each OD pair to a particular vehicle. Figure 3.16 shows the assignments for the 6, 10 and 15 vehicle cases. The goal of these assignments is to provide equitable service to all passengers throughout the system. This is done

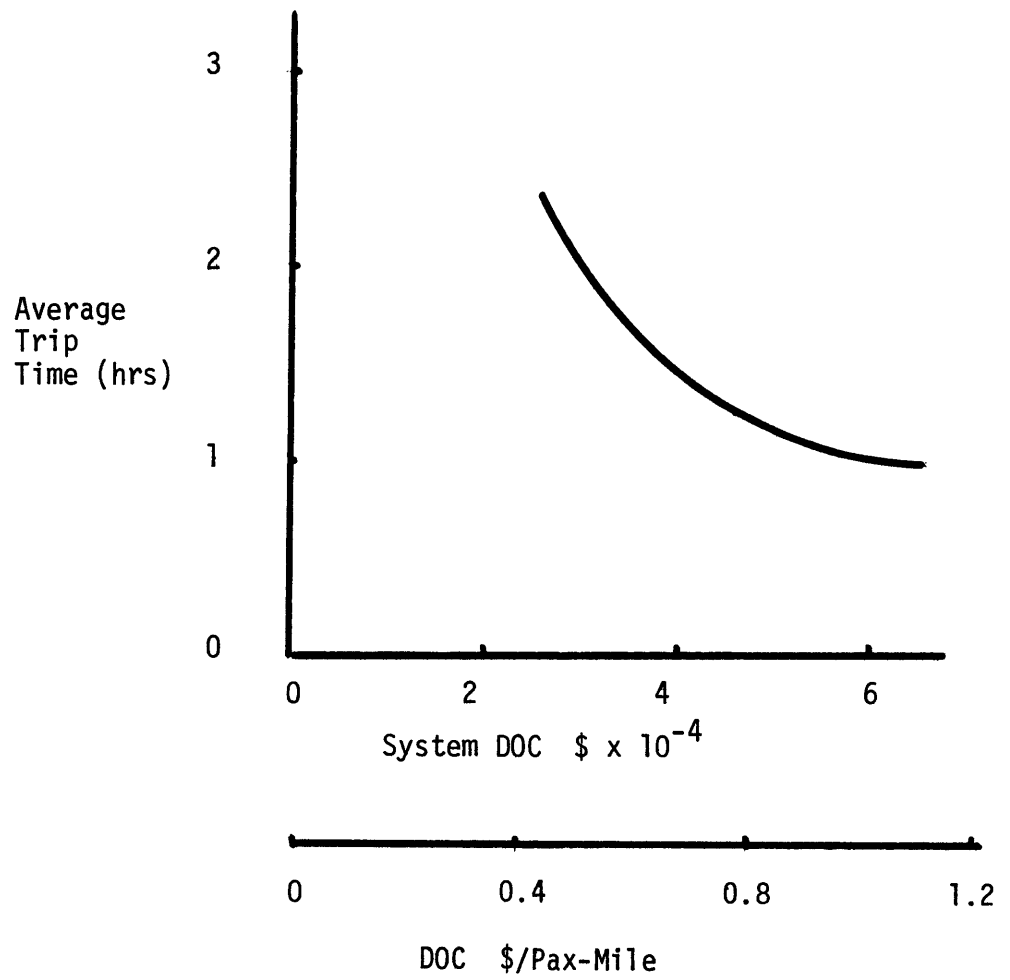


Figure 3.15 - Trip Time vs. DOC, Allstop Routing

70

6 Vehicles

Destination

		1	2	3	4	5	6
Origin	1	-	1	5	2	5	2
	2	1	-	6	4	6	1
	3	5	6	-	3	6	3
	4	2	4	3	-	5	4
	5	5	6	6	5	-	-
	6	2	1	3	4	-	-

10 Vehicles

Destination

		1	2	3	4	5	6
Origin	1	-	5	10	10	6	1
	2	6	-	9	9	5	2
	3	10	9	-	7	8	3
	4	10	9	8	-	7	4
	5	5	6	7	8	-	-
	6	1	2	3	4	-	-

15 Vehicles

Destination

		1	2	3	4	5	6
Origin	1	-	15	13	12	5	1/9
	2	12	-	15	14	6	2/9
	3	15	14	-	13	1/11	3/10
	4	13	12	14	-	8/11	4/10
	5	5	6	7/11	8/11	-	-
	6	1/9	2/9	3/10	4/10	-	-

Figure 3.16 - Nonstop Routing Vehicle Assignments

by inspection, no attempt at optimization, or use of a formal algorithm is made.

The results of three simulation runs are shown below in Table 3.5.

	AVERAGE TRIP TIME (hrs)		
	15	10	6
Nonstop	0.63	0.89	1.37
Allstop	1.02	1.37	2.3

Table 3.5

The results are quite interesting; the nonstop cases with 6 and 10 vehicles do as well as the allstop cases with 10 and 15 vehicles, respectively. There are several reasons for this. As mentioned before, the nonstop routing avoids the congestion of the allstop scheme. The effect of congestion in the allstop cases is to reduce the number of available seats on each departure. So, even though the service is less frequent in the nonstop case, the rate at which available seats are leaving some stations (3 and 4) is greater than in the allstop case. There is also a significant difference in ride time between these two cases. In the allstop case, the average ride time is .8 hours, in the nonstop, it is .3 hours. Although average wait time decreases with the number of vehicles, this ride time is constant. So, no matter how many vehicles are flying, the average trip time is always greater than the ride time. Figure 3.17 summarizes the results.



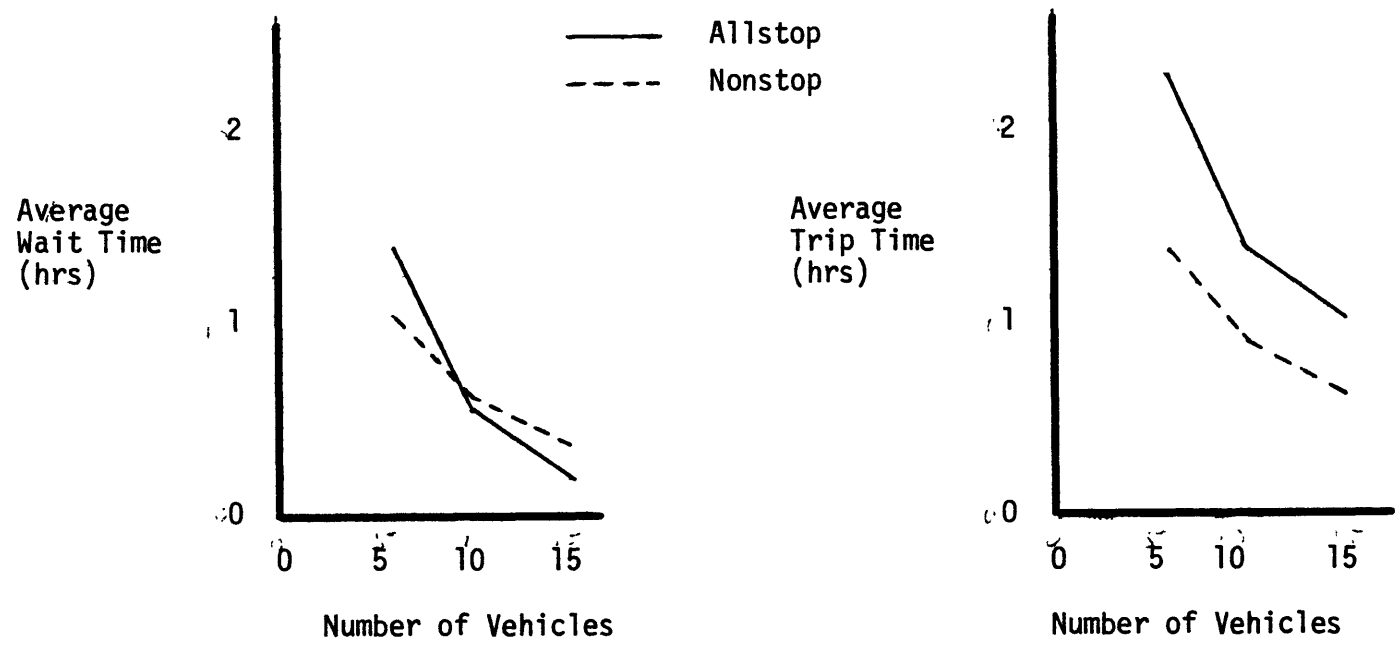


Figure 3.17 - Average Wait and Trip Times vs. Number of Vehicles, Allstop and Nonstop Routing

The performance of the system operating under the nonstop strategy is clearly greater than under the allstop strategy. This is due to the reduced sensitivity of LOS to overloading during the rush hours. Even though the nonstop strategy represents a tremendous improvement in system performance, average travel time still varies significantly during the rush hours. In fact, only the 15 vehicle case achieved equilibrium during the rush hours (see Figure 3.18).

### 3.3 Dynamic Vehicle Controls

In the previous sections, it is shown that the USH system, operating under a fixed routing policy, needs a surprising number of vehicles to provide reasonable service to all passengers. This results from the fact that while the routing is fixed, the system conditions vary substantially during the day. A fixed route system designed to handle the peak demand is tremendously under utilized during most of the day. But a system which allows some overloading delivers degraded service to rush hour passengers. A goal of this section is to develop a system which provides good, efficient service throughout the day. To achieve this goal, the system is controlled by a strategy which reacts to changes in the demand pattern and to system conditions. The control algorithm must decide which passengers to service, when and what type of service is to be provided, and how the vehicles should be routed to provide this service. To make this a tractable problem, the decisions are simplified. The problem considered here is: given a vehicle which has just finished service to a particular OD pair, which

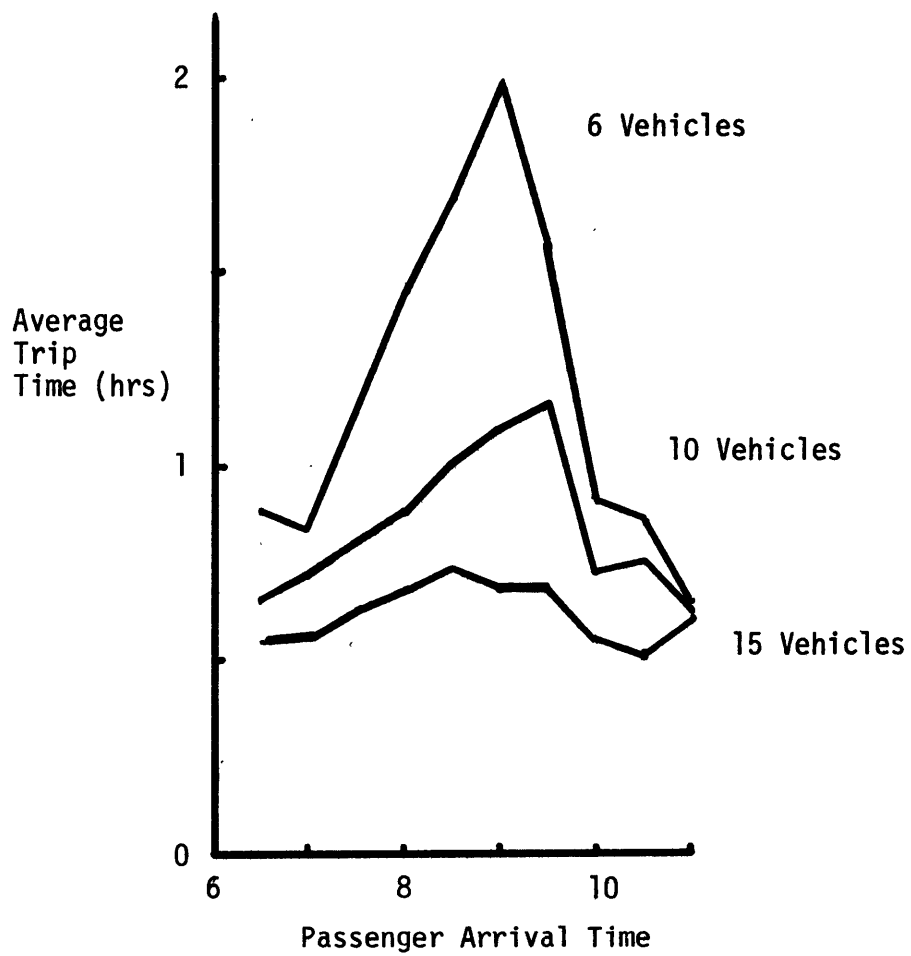


Figure 3.18 - LOS vs. Time of Day, Nonstop Routing with No Controls

OD pair should be served next by this vehicle? When the decision is made, nonstop service to these passengers begins immediately. Later, the control is relaxed to allow some onestop service. Finally, vehicle holding strategies are introduced, to add some freedom in the decision as to when the service takes place.

### 3.3.1 Dynamic Vehicle Routing

In developing vehicle control strategies, some simple rules come to mind. These are service criteria such as serve an OD pair if the number waiting is greater than a certain level, or serve an OD pair if the maximum wait for this group is greater than a certain time limit. Although these rules are reasonable intuitively, they have some serious problems, particularly in the USH environment. First, the determination of the limit values is critical. It is unlikely that limits designed to provide good service during the rush will work well during the slack period. As demand patterns vary during the day, so also must the limit values. Also, these rules are simply constraints on service. It seems more reasonable to use criteria such as serve the OD pair with the most passengers waiting, or serve the pair with the maximum waiting time. The first rule should work well except during the rush. Since the OD pairs with the most passengers waiting for service will almost always be the commuter pairs, the general passengers are given virtually no service during the rush hours. The second rule is more equitable for the general travelers, but it tends to slow down the larger group of commuters. This rule would serve a single passenger who has waited for

1 hour before 100 passengers who have waited for 59 minutes. It seems that a viable rule should consider both, the number of passengers waiting and the maximum wait in each group, to choose the OD pair for service. This indicates a need to establish a cost \$/hr, for passenger waiting time. The interest in controlling the long waits suggests that this cost per unit time is increasing with the length of wait. The proposed model of passenger waiting cost is illustrated in Figure 3.19. If a passenger waits  $w$  hours, the total cost to the system is,

$$\int_0^w (a + bx)dx = aw + \frac{b}{2} w^2 \quad (3.1)$$

An optimal strategy is one which minimizes the total passenger waiting cost to the system. This is no easy task on the optimal level, but the following heuristic is proposed with the same objective. The OD pair to receive service is chosen on the basis of the rate at which passenger cost is accumulating. This rate is equivalent to the marginal cost, MC, of delaying service to the OD pair. The OD pair with the highest MC is the most attractive to serve. This marginal cost is the change in total cost, TC, per unit time, for the OD pair.

$$MC = \frac{\partial TC}{\partial t} \quad (3.2)$$

It is necessary to devise a method to calculate, or at least estimate the marginal cost for each OD pair. Consider an OD pair where passengers begin arriving at  $t = 0$ , at an average arrival rate of  $\lambda(t)$ . The total cost to the system at the time of the next

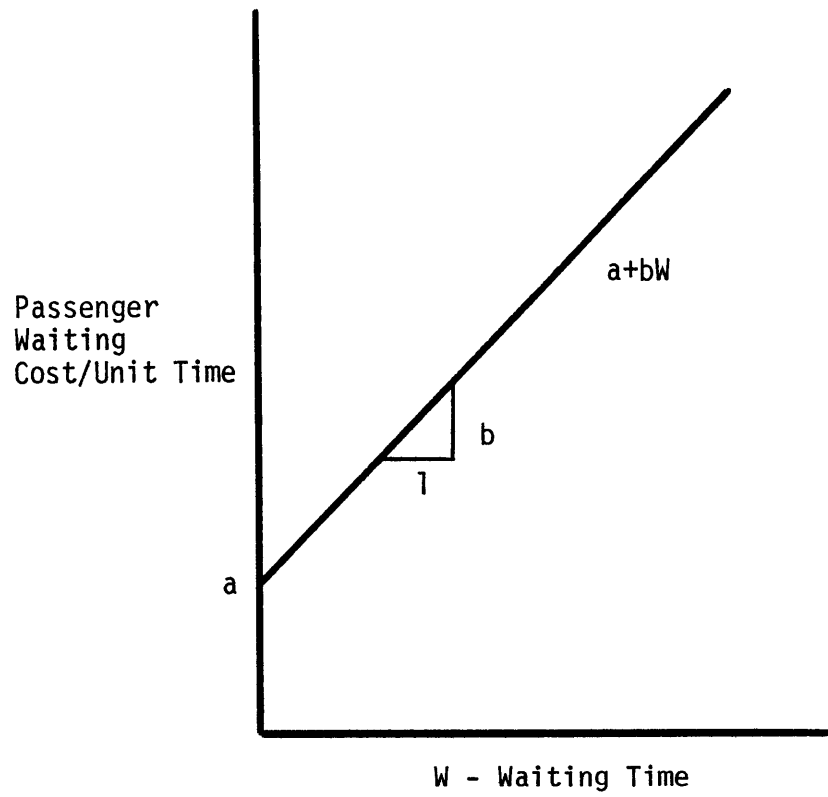


Figure 3.19 - Passenger Waiting Cost Function

departure,  $t_d$ , is the sum of the cost to each passenger arriving in this headway. The individual passenger cost depends on the actual time of arrival. Since passenger arrivals are uncertain, the total cost to the system is a random variable and cannot be determined in advance. But we can determine the expected total cost. The cost to a passenger arriving in the interval  $(s, s+dt)$  is  $a(t_d - s) + \frac{b}{2} (t_d - s)^2$ . The probability of an arrival in this interval is  $\lambda(s)$ . So, the expected total cost to the system given a departure occurring at  $t_d$  is

$$E(TC) = \int_0^{t_d} [a(t_d - s) + \frac{b}{2} (t_d - s)^2] \lambda(s) ds \quad (3.3)$$

The expected marginal cost of delaying service is then,

$$E(MC) = \frac{\partial}{\partial t} E(TC) = \frac{\partial}{\partial t} \int_0^{t_d} [a(t_d - s) + \frac{b}{2} (t_d - s)^2] \lambda(s) ds \quad (3.4)$$

Integrating yields:

$$E(MC(t_d)) = a[L(t_d) - L(0)] + b[(t_d) - (0)]$$

where

$$L(t_d) = \int_0^{t_d} \lambda(x) dx$$

and

$$\mathcal{L}(t_d) = \int_0^{t_d} L(x) dx$$

so

$$E(MC(t_d)) = aL(t_d) + b\mathcal{L}(t_d) \quad (3.5)$$

where  $a$  and  $b$  are the intercept and slope of the individual waiting cost curve.

This expression for expected marginal cost can be simplified by making the following observations. The parameter  $L(t_d)$  is the expected number of passengers to arrive in this headway  $E(n)$ , and  $\mathcal{L}(t_d)$  is the expected number of passenger hours which have accumulated in the headway. If the average rate of passenger arrival is constant (a reasonable assumption for short headways), the expected number of passenger hours is simply the expected number of arrivals times the average wait,  $t_d/2$ .

$$E[\mathcal{L}(t_d)] = E[N] t_d/2 \quad (3.6)$$

Substituting into Eq. (3.5) yields,

$$\begin{aligned} E(\text{MC}) &= aE(N) + bE(N) t_d/2 \\ &= E(N) \left[ a + \frac{b}{2} t_d \right] \end{aligned} \quad (3.7)$$

where

$$E(N) = \int_0^{t_d} \lambda(x) dx$$

This results in an expression for expected marginal cost which is much simpler than before.

Since the control decisions are made during system operation, the algorithm has access to the actual number of passengers waiting. MC can be estimated directly by replacing the expected number waiting



$E(N)$ , with the actual number,  $N$ . More care is necessary in replacing  $t$ . It must be remembered that  $t_d$  is not the time since the last departure, but rather the time since passengers of this group started arriving. Clearly,  $t_d$  is the maximum wait,  $w$ , for this group. So, MC is estimated as follows

$$MC = N(a + \frac{b}{2} w) \quad (3.8)$$

which is the key result for this section. This expression for MC is precisely of the form motivated earlier. The marginal cost of not serving the OD, or the attractiveness of serving the pair, depends on both the number waiting and the maximum wait. The two factors which intuitively should appear do, and are combined in a simple way by the passenger cost parameters. Possibly the most important aspect of this result is that the information required to calculate MC is minimal. For each OD pair, 2 values are required. In a real system, these values can be maintained by recording passenger arrivals. In the simulation model, these values are readily available.

The effect of the parameter  $b$  can be seen from this expression for MC. If  $b = 0$  (constant cost function per unit time) MC is proportional to the number waiting. Recall that this is one of the service criteria suggested earlier. In this case, OD pairs are chosen to minimize systemwide average travel time. But as indicated earlier, there is an inequity in LOS to passengers of low demand rate OD pairs. Consider two OD pairs with average arrival rates 10 and 100 passengers/hour. If

the OD pair to be served is chosen only on the basis of the number waiting, the high arrival rate pair is served ten times as often as the low rate pair. During a 2-hour period, the low arrival rate passengers most likely receive no service at all. This is the situation faced by the general passengers during the rush hour in the USH system. The effect of increasing  $b$  is to reduce this difference in service frequency. In fact, for  $a = 1$ ,  $b = 10$ , the frequency of service to the low rate passengers is increased from  $1/10$  to  $1/4$  the frequency of the high rate passengers. The result is a decrease in the uncertainty of travel time throughout the system. But for  $b$  greater than zero, the system objective is no longer the minimization of average travel time. The objective becomes the minimization of a function of both average and maximum travel time. As  $b$  increases, average travel time does also. This relationship between the average travel time and the variance of travel time can not be avoided, but the parameter  $b$  allows the tradeoff to be controlled.

Now, how should this expression for MC be used to control vehicle routing? Consider a vehicle which is finishing service to an OD pair, the system must now decide which pair to serve next with this vehicle. The MC for several OD pairs is determined, and at first it seems reasonable that the OD pair with the greatest value of MC should be served next. But the vehicle is a scarce resource in the system, and the time it is tied up serving one OD pair, making it unavailable to the rest of the system, needs to be considered. It has been shown that in priority queues, with certain cost functions, the optimal

ordering by which customers are served is determined by the ratio of waiting cost to service time.<sup>9</sup> Using the same reasoning here, the most attractive OD pair to serve is the one with the largest value for the ratio,

$$MC_{ij}/S_{ij}$$

where  $MC_{ij}$  and  $S_{ij}$  are the marginal cost and service time, respectively, for OD pair  $i$  to  $j$ . Note that if the vehicle is at  $i$ , and it is decided to serve OD pair  $jk$ , the time required to complete service is  $S_{ij} + S_{jk}$ , since the vehicle must fly from  $i$  to  $j$  and then to  $k$  to complete the service. Obviously, this system favors short flights, and service to OD pairs originating at the vehicle's present position.

Now that it has been determined how to identify the OD pair to serve, we must decide which pairs to consider and when the routing decision is to be made. Once again, a simple strategy comes to mind. When the vehicle lands at  $i$ , choose the pair  $ij$ , which maximizes  $MC_{ij}/S_{ij}$ , fly to  $j$  immediately. In a system with demand distributed uniformly between all OD pairs, this rule may work. But in a system with strong one-way flow, problems develop. Consider a commuter origin during the rush, there is heavy traffic out of this station, but light traffic in. A situation can occur where there are many passengers waiting to leave this station, but there is no demand into the station. This strategy only considers service to OD pairs which originate at the present vehicle locations. Since there is no demand into this station, and hence no vehicles arriving here, the algorithm never "sees" the commuters waiting

to leave. Service in this case is somewhat haphazard, since service to the heavy traffic out of some stations depends on the light traffic into these stations. This myopia can be avoided with a two-look ahead approach. Consider the following algorithm. When the vehicle lands at  $i$ , choose pairs  $ij$  and  $jk$  which maximize  $MC_{ij}/S_{ij} + MC_{jk}/S_{jk}$ , fly to  $j$  immediately. So a vehicle at station  $i$  considers the value of serving the combination  $ij$  and  $jk$ . In this way, all OD pairs in the system are considered. There is a subtle problem here though, vehicle interference. This results from a lack of communication between the vehicles. Suppose there are many  $jk$  passengers waiting for service, and a vehicle initially at  $i$  flies to  $j$  to serve the  $jk$  passengers. At the same time, another vehicle has seen the need for  $jk$  service, and flies from  $i'$  to  $j$ . In the meantime, the first vehicle arrives and serves the  $jk$  passengers, and when the second vehicle arrives,  $jk$  service is no longer very attractive. If there are other OD pairs leaving  $j$ , the trip by the second vehicle wasn't made completely in vain, but obviously, this duplication of effort reduces system performance. Consider the same situation, but this time the first vehicle flags the  $jk$  pair as having a service pending. The flag in effect sets  $MC_{jk}$  to zero until service is begun. This discourages other vehicles from flying to  $j$ , in hope of serving the  $jk$  passengers. Of course, we can't have the vehicle flag  $jk$ , fly to  $j$ , look around and then decide to fly to  $k'$  (leaving the  $jk$  passengers in limbo). So, once the vehicle chooses to fly  $ij$  and  $jk$ , it is obligated to complete the flight to  $k$ . This algorithm (summarized

below) is invoked at every second step, since things are fixed at  $j$ .

When the vehicle lands at  $i$ , choose the pairs  $ij$ , and  $jk$  which maximize the following.

$$MC_{ij}/S_{ij} + MC_{jk}/S_{jk}$$

fly to  $j$  immediately. Flag the OD pair  $jk$ , such that it is not considered for service (by other vehicles) until this vehicle reaches  $j$ . Fly to  $k$ .

SMUSH was modified to incorporate this algorithm. The system was simulated using fleet sizes of 6 and 10 vehicles. There is nothing unexpected in the results of the simulation. The most surprising aspect is that this simple strategy works so well. Table 3.6 summarizes the results.

		Total Trip Time		
		Average Wait	Average	Standard Deviation
6 Vehicles	a = 1, b = 0	0.84	1.17	.77
	a = 1, b = 10	0.89	1.20	.69
	Fixed Route	1.04	1.77	.81
10 Vehicles	a = 1, b = 0	0.42	.73	.43
	a = 1, b = 10	0.44	.76	.33
	Fixed Route	0.59	.89	.54

Table 3.6

Considering that the original nonstop case has a reasonable route structure, it is remarkable that LOS was improved so much, on the first cut with this simple strategy. In the 10 vehicle case, average waiting time is reduced 25% with the dynamic routing algorithm. As

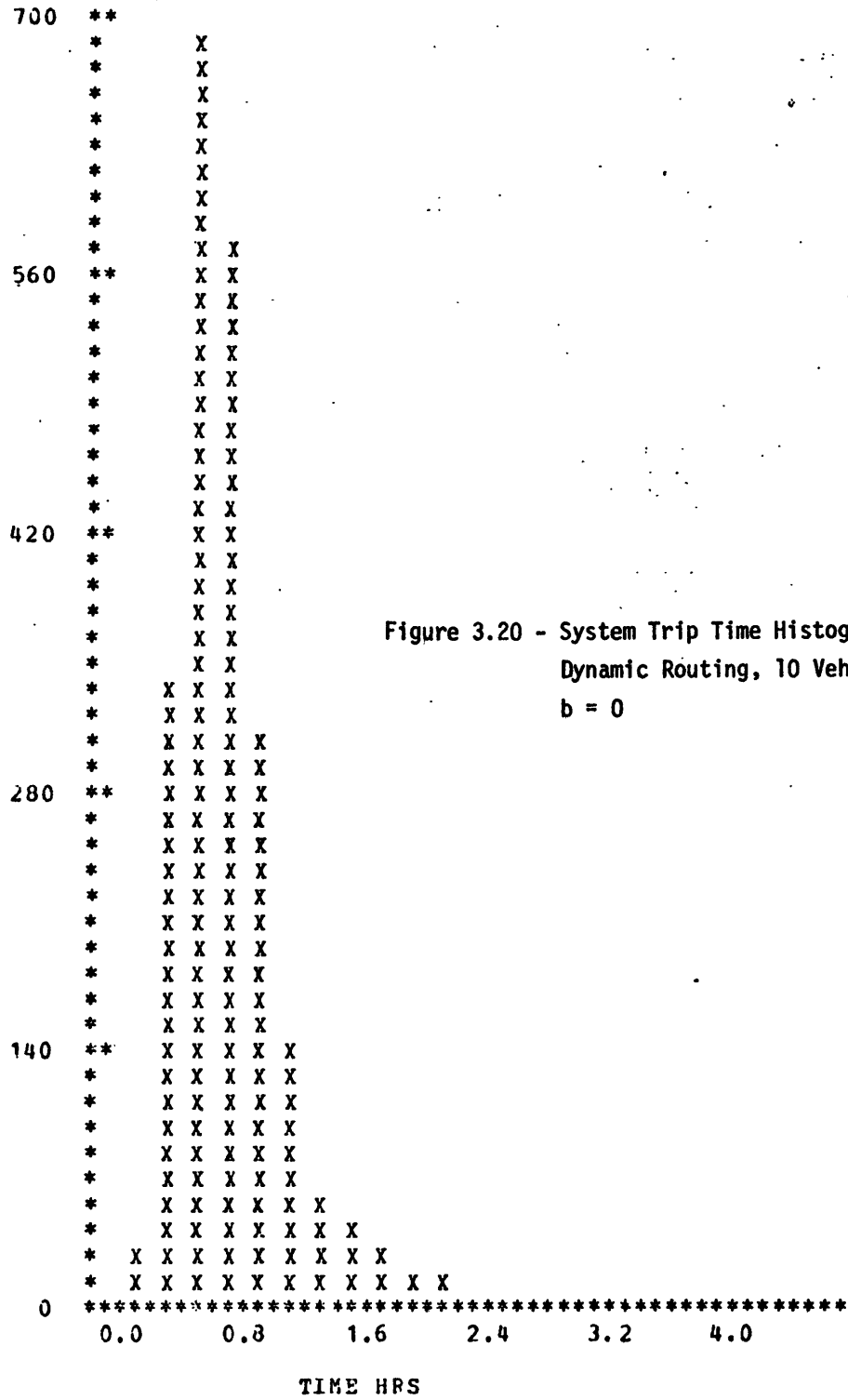
anticipated when  $b$  is increased, the uncertainty in trip time is reduced, and average trip time increases. What is surprising is that a substantial decrease in standard deviation is achieved with only a slight increase in average time. The effect of the reduced uncertainty can be more dramatically seen in the total system trip time histograms (Figures 3.20, 3.21). Notice that when  $b$  is increased to 10, both tails of the distribution are brought in; there are few very short and very long trips made.

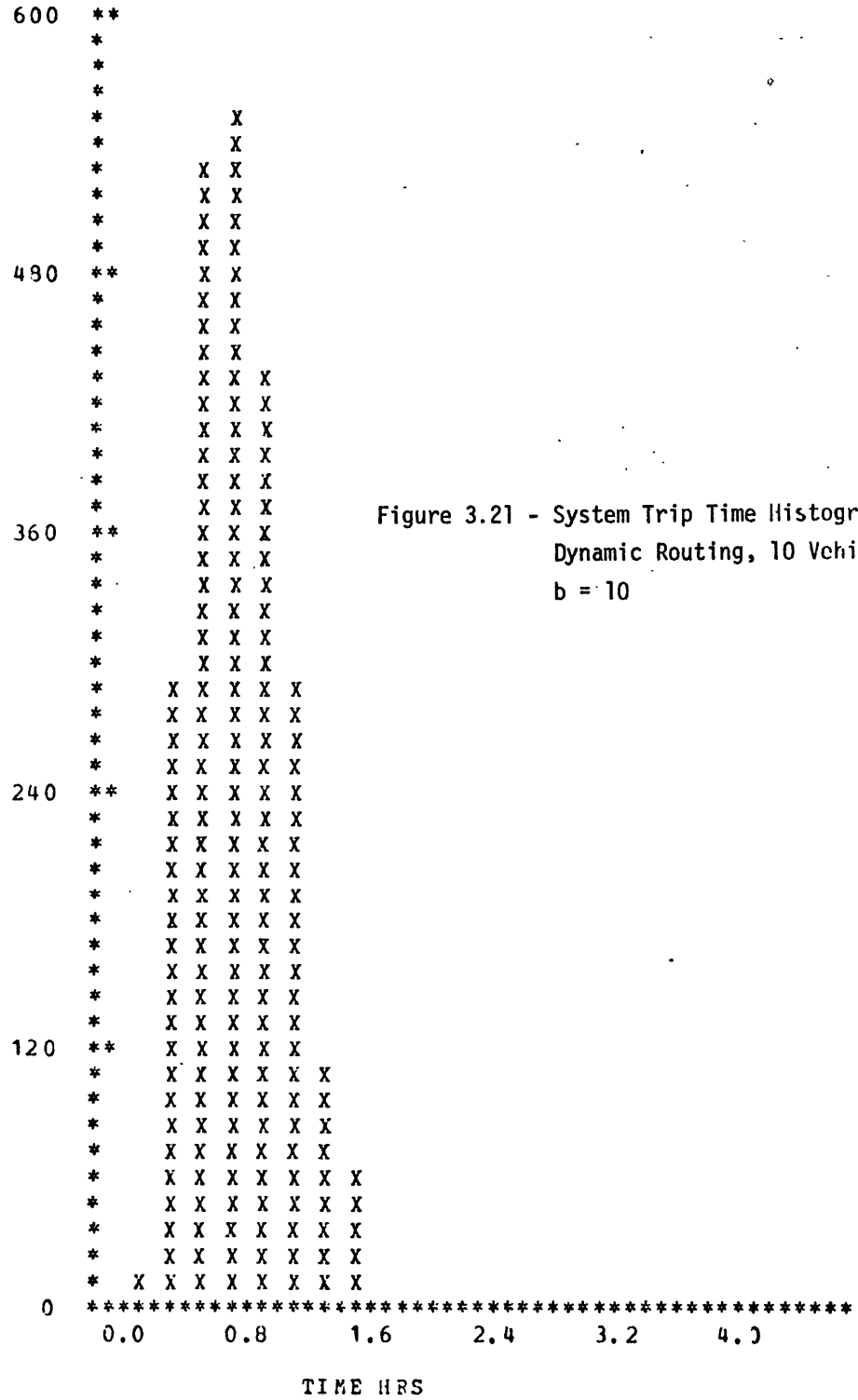
Recall that when the vehicle chooses a route, the next two stops  $j$  and  $k$  are determined. Because of this, some passengers at  $i$ , going to  $k$ , can be boarded; they receive one-stop service. A modification was made to SMUSH to allow this one-stop option. Again, the system is simulated with 6 and 10 vehicles using this one-stop service policy. This strategy yields an additional increase in LOS over the previous cases. The results are summarized in Table 3.7.

		Total Trip		
		Average Wait	Average	Standard Deviation
6 Vehicles	$a = 1, b = 10$	.82	1.16	.62
10 Vehicles	$a = 1, b = 10$	.39	.75	.32

Table 3.7

The system's enhanced ability to cope with the demand peaking is illustrated in the plot of average travel time vs. arrival time (Figure 3.22). Note that the 10 vehicle case achieved stability during







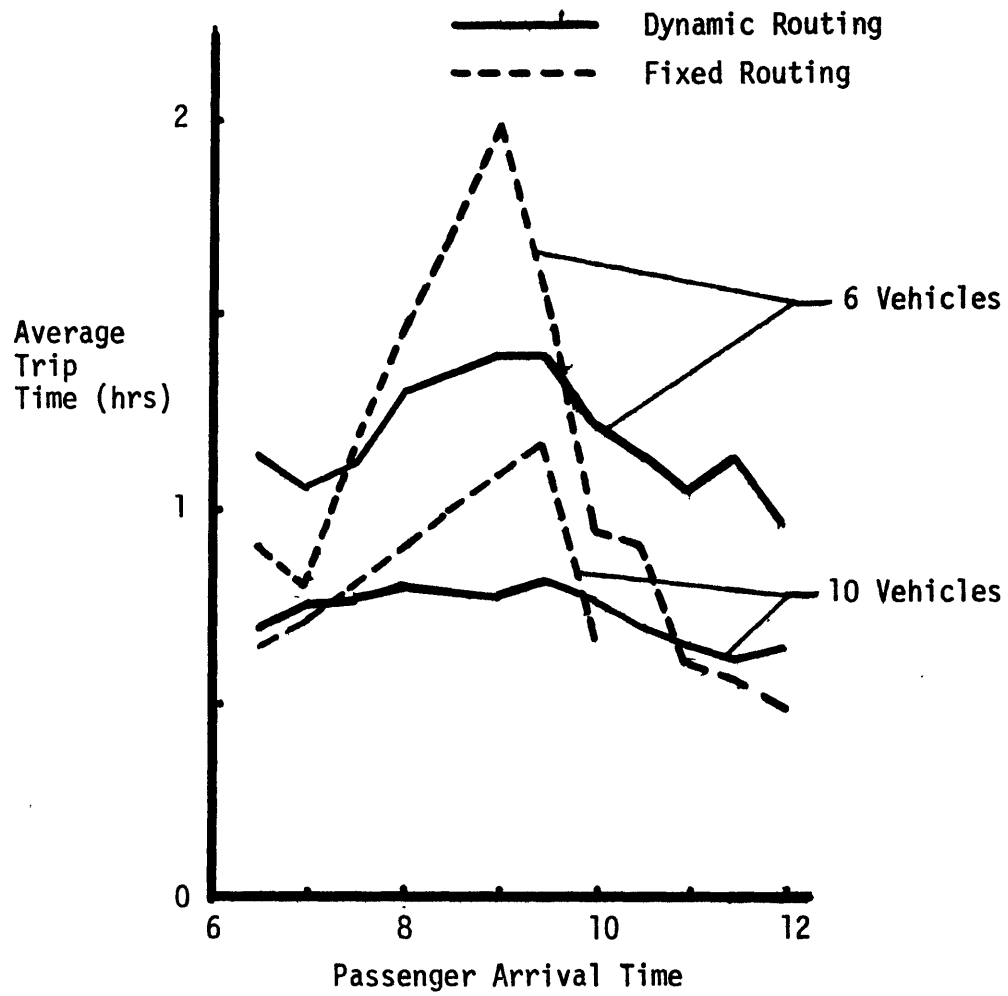


Figure 3.22 - The Variation of LOS With Time of Day, Dynamic and Fixed Routing

the rush. And although there is some congestion in the 6 vehicle case, the effects are substantially reduced by this strategy. This route choice algorithm has tremendously reduced the system sensitivity to demand peaking and overloading.

The method developed here is significant because it is applicable to any USH system. Recall that for the original nonstop case, a routing structure had to be determined before the system could be operated. To intelligently determine a set of routes, some information about the individual system is necessary. With the method described here, no prior information is required to set up the system. Once the system starts, it essentially runs itself. Therefore, this strategy can be applied to any USH network without modification.

### 3.3.2 Vehicle Holding Strategies

Vehicle holding is employed to control the pace of the system and to reduce operating costs (and in some cases to improve LOS). In the previous sections, the aircraft fleet was sized such that the peak demand could be met without serious congestion. Although the demand drops off after the rush, the system continues to run at a high rate. Algorithms are now introduced to judiciously slow the system in the off-peak hours. Of course, in doing this, LOS in the slack periods drops off slightly.

Three holding strategies are considered. The first, a minimum headway strategy, is employed to increase LOS (the reduction in operating cost is a side benefit). The second algorithm is an extension of the

route choice scheme. The last strategy considers both passenger waiting cost and system operating costs, and allows the tradeoff between operating cost and LOS to be controlled during system operation.

### 3.3.2.1 Minimum headway controls

This strategy improves LOS provided to passengers by controlling the randomness in service. More precisely, a lower bound on headways is maintained by holding vehicles. This strategy was developed by Osuna and Newell.<sup>10</sup>

Consider passengers of a particular OD pair waiting to be served. For simplicity, assume that passengers arrive deterministically at a rate of 1 per minute. Suppose the headway for this pair is 5 minutes. It is a simple matter to calculate a waiting cost,  $C$ , for this group. Suppose the service is delayed one minute. All the passengers already waiting must wait an extra minute in which time one new passenger will arrive. This delay will increase the total cost of this group by some  $\Delta C$ . The change in cost due to the one minute delay can be found by summing up the extra cost to each passenger, or by making the following observation. The two groups are identical, except the delayed group now has one passenger which must wait 6 minutes. This one passenger forms a new subgroup placed on the long end of the original group. The waiting cost to this subgroup is  $\Delta C$ . In general, the effect of delaying service to an OD pair is the creation of a subgroup such as the one described here. The extra cost of this delay is simply the cost to the passengers

in this subgroup.

It was shown that for a particular OD pair that the following strategy minimizes average passenger cost. Hold the vehicle if the cost to the subgroup created by the lengthened headway is less than average passenger cost. Using the headway distribution for this OD pair, the average passenger cost is calculated using random incidence arguments. The passenger cost function shown in Figure 3.19 is again used. The cost to a passenger waiting  $t_0$  is

$$at_0 + \frac{b}{2} t_0^2$$

The expected cost per passenger in a headway of length  $h_0$  is

$$\begin{aligned} E(c | h_0) &= \int_0^{h_0} a(h_0 - s) + \frac{b}{2} (h_0 - s)^2 ds \\ &= \frac{a}{2} h_0 + \frac{b}{6} h_0^2 \end{aligned} \quad (3.9)$$

Let  $w$  be the length of the headway entered by random incidence.

$$E(w) = \frac{E(h^2)}{E(h)} \quad (3.10)$$

So, the expected cost for a passenger entering the system at random is

$$E(\text{cost/passenger}) = \frac{E(w)}{2} [a + \frac{b}{2} E(w)] \quad (3.11)$$

The expected cost to the subgroup created by extending the headway from  $h_0$  to  $h_0 + dh$  is

$$ah_0 + \frac{b}{2} h_0^2 \quad (3.12)$$

So, the optimal holding strategy becomes, hold if

$$ah_0 + \frac{b}{2} h_0^2 < \frac{E(h_0^2)}{2E(h_0)} \left[ a + \frac{b}{3} \frac{E(h_0^2)}{E(h_0)} \right] \quad (3.13)$$

If  $a \gg b$ , this becomes

$$\text{hold if:} \quad h_0 < \frac{E(h^2)}{2E(h)} \quad \text{This is the classic strategy developed by Osuma \& Newell}$$

If  $b \gg a$ , this becomes:

$$\text{hold if:} \quad h_0 < \frac{E(h^2)}{\sqrt{3} E(h)}$$

So, as long waits become more costly, the holding strategy becomes more stringent, forcing more headways close to the expected value.

To determine the impact of this strategy on system performance, information about the headway distribution for each OD pair is necessary. Fortunately, this information is available from the previous simulation runs. But which case or cases should be considered? A minimum headway rule was already applied to the allstop routing case. This strategy would have no effect on the fixed nonstop routing case, since the headways are nearly constant. The only situation for which this strategy is assessed for the 10 vehicle case with  $a = 1$  and  $b = 10$ . Only two OD pairs are considered, a commuter pair 36, and a general travel pair 34. Headway data is collected and the distribution parameters are estimated as shown in Table 3.8.

	3-4	3-6	
$\bar{H}$	.87	.54	
S.D.	.51	.31	$E(h) \approx \bar{H}$
$E(h)$	.87	.54	$E(h^2) \approx (S.D.)^2 + (\bar{H})^2$
$E(h^2)$	1.02	.39	

Table 3.8

The control rule reduces to a quadratic in  $h_{min}$ , the minimum headway.

$$3-4 \quad h_{min} = .67 \text{ hrs.}$$

$$3-6 \quad h_{min} = .41 \text{ hrs.}$$

Based on the amount of time the vehicles would have been held, it is concluded that the effect of this strategy, if implemented, is minimal. Although the effects on LOS and operating cost are small, they are positive. In some systems, this strategy can improve service (such as the allstop case). But the results here suggest that the vehicle controls are operating the system in such a way as to make this holding strategy superfluous.

### 3.3.2.2 Holding based on the marginal cost

The holding strategy discussed in this section is an extension of the vehicle routing algorithm described in 3.3.1. Recall that this algorithm chooses the OD pair to serve next, based on a value to the system for service, MC/S. The OD pair with the maximum value of this ratio is the most attractive to serve. The same idea is used here, but

the choice is not between different OD pairs, but rather between serving an OD pair now or waiting some  $\Delta t$  for service to begin. The strategy is this, given that an OD pair,  $ij$ , has been selected for service, hold the vehicle at  $i$  if

$$\frac{MC(t_{\text{now}})}{S} < \frac{MC(t_{\text{now}} + \Delta t)}{S + \Delta t} \quad (3.14)$$

and the number waiting  $<$  capacity.

Recall that the marginal cost can be estimated as

$$MC = N(a + bw/2)$$

where  $N$  = number waiting

$w$  = maximum wait in this group

$a, b$  = parameters of the passenger waiting cost function.

Substituting into (3.14) yields

$$\frac{N(a + \frac{b}{2} w)}{S} < \frac{(N + \lambda \Delta t)(a + \frac{b}{2} (w + \Delta t))}{S + \Delta t}$$

Upon simplification, the rule becomes:

$$\text{Hold if: } N < \lambda_s \left[ \frac{a + \frac{b}{2} w}{a + \frac{b}{2} (w - S)} \right] \equiv N_{\text{req}} \quad (3.15)$$

and

$$N < \text{Capacity}$$

There are some intriguing and perplexing aspects to this rule. Obviously, the rule is simple. At the time of a departure, this is a go or hold rule based on the number of passengers waiting and the maximum

waiting time. The decision is illustrated in Figure 3.23. For specified  $a$ ,  $b$ ,  $s$ , and  $\lambda(t)$ , the holding region is shown in terms of number waiting and the maximum wait. An interesting aspect of the rule is that it gives a clear-cut criterion for holding the vehicle until full,  $\lambda s > \text{capacity}$ . Intuitively, this is reasonable, since it suggests that the vehicle should be held to achieve higher loading if either arrival rate is high, or service takes a long time.

When  $b = 0$ , the rule becomes: hold if  $N < \lambda(t)S$ , or if  $\frac{N}{S} < \lambda(t)$ . This says hold if the service rate is less than the arrival rate. This is reminiscent of the stability condition for a queuing system, but there is no intuitive justification for the relationship here. This rule reacts to varying demand rate, since  $\lambda(t)$  appears. Also, the holding criterion is completely independent of headway distribution, which means no prior information is necessary to implement the rule.

A most disturbing aspect of this dispatching rule is the appearance of the term  $(w - S)$  in the denominator. This is not only difficult to justify intuitively, it is disastrous mathematically. For ranges of  $S$  and  $w$ , the number required for dispatch can be either infinite or negative.

The dependence on  $b$  of  $N_{\text{req}}$  is also troublesome. Strangely enough, as  $b$  increases, so does  $N_{\text{req}}$  (Figure 3.24). A possible effect of this dependence is that the algorithm may manufacture extra delay costs, by holding some passengers if their individual costs increase rapidly enough. Of course, this self-serving characteristic of the



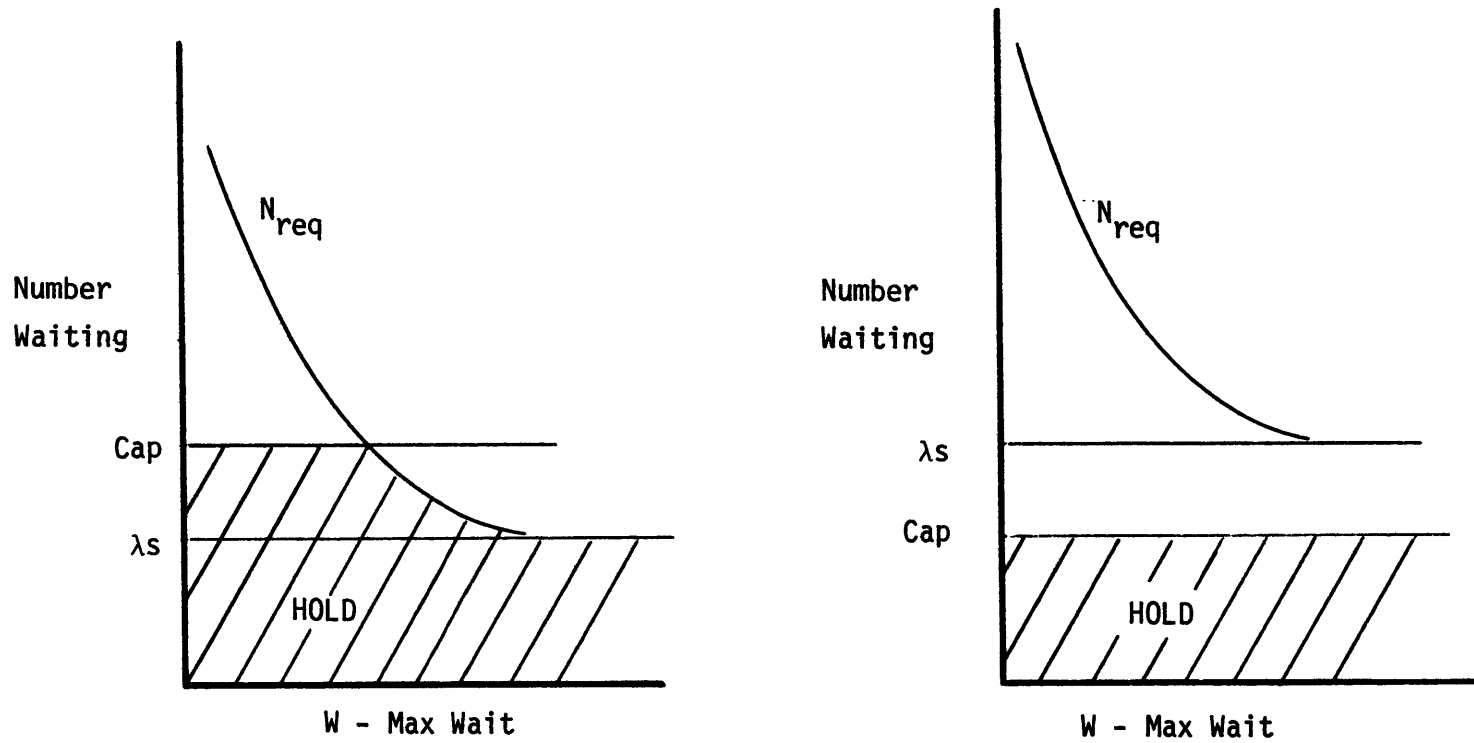


Figure 3.23 - The Decision to Hold, in Terms of Number Waiting and Max Wait

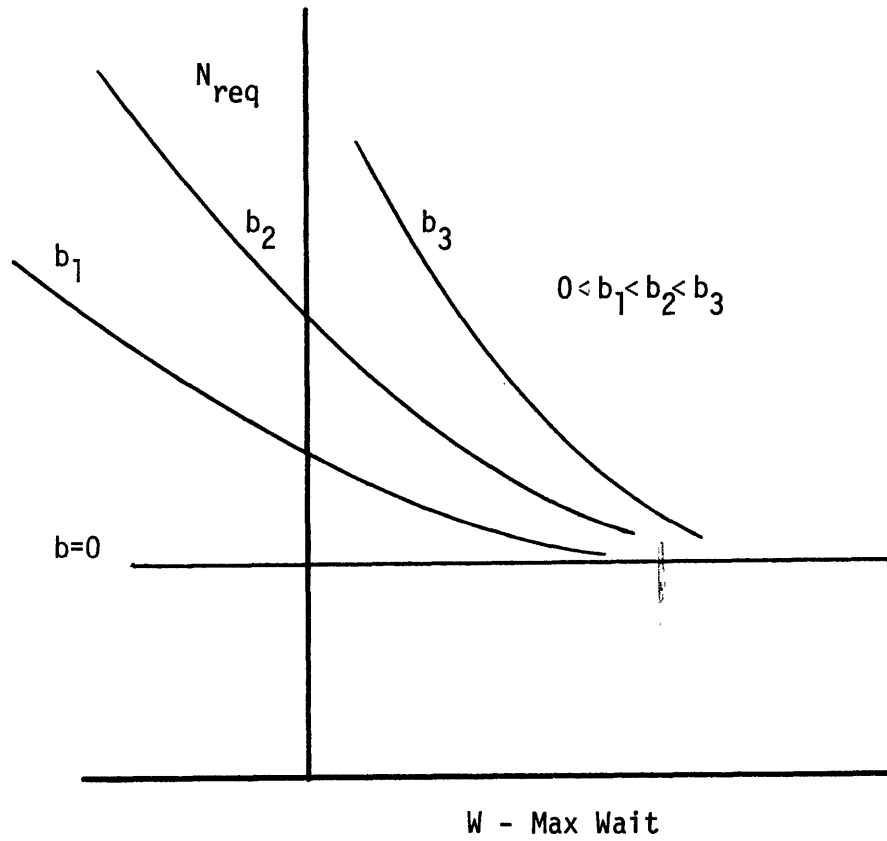


Figure 3.24 - Illustration of the Dependence of  $N_{req}$  on  $W$  and  $b$

rule helps no one.

SMUSH was modified to incorporate this holding rule. If at the time of departure, the number waiting is less than the calculated value for  $N_{req}$ , the departure is postponed for 3 minutes. This cycle is repeated until the departure occurs. The morning operations are simulated using 10 vehicles under the control of this rule used in conjunction with the dynamic route choice algorithm. The results are disappointing, but not surprising. Where it is expected that vehicles are held only during the off-peak hours, under this rule, vehicles are held almost uniformly throughout the day. The holding does reduce operating cost somewhat, and travel time increases. The overall effect of this rule is to slide the system up the LOS-cost curve (see Fig. 3.25).

One reason for the disappointing results is the myopic character of this rule. The holding at a station depends only on the conditions of a single OD pair. The rule is completely oblivious to changes occurring in the rest of the system. This leads to an explanation for vehicles being held during the rush period. Suppose a vehicle is in the CBD and it chooses to fly to the suburbs and then back to the CBD to serve commuters during the rush. This choice is made primarily on the basis of the large MC for the OD pair coming into the CBD. Now when the vehicle tries to leave the CBD, the rule sees that there is not much demand out of this station and decides that there is no need to hurry. So, the vehicle is held on the first leg of the trip even though

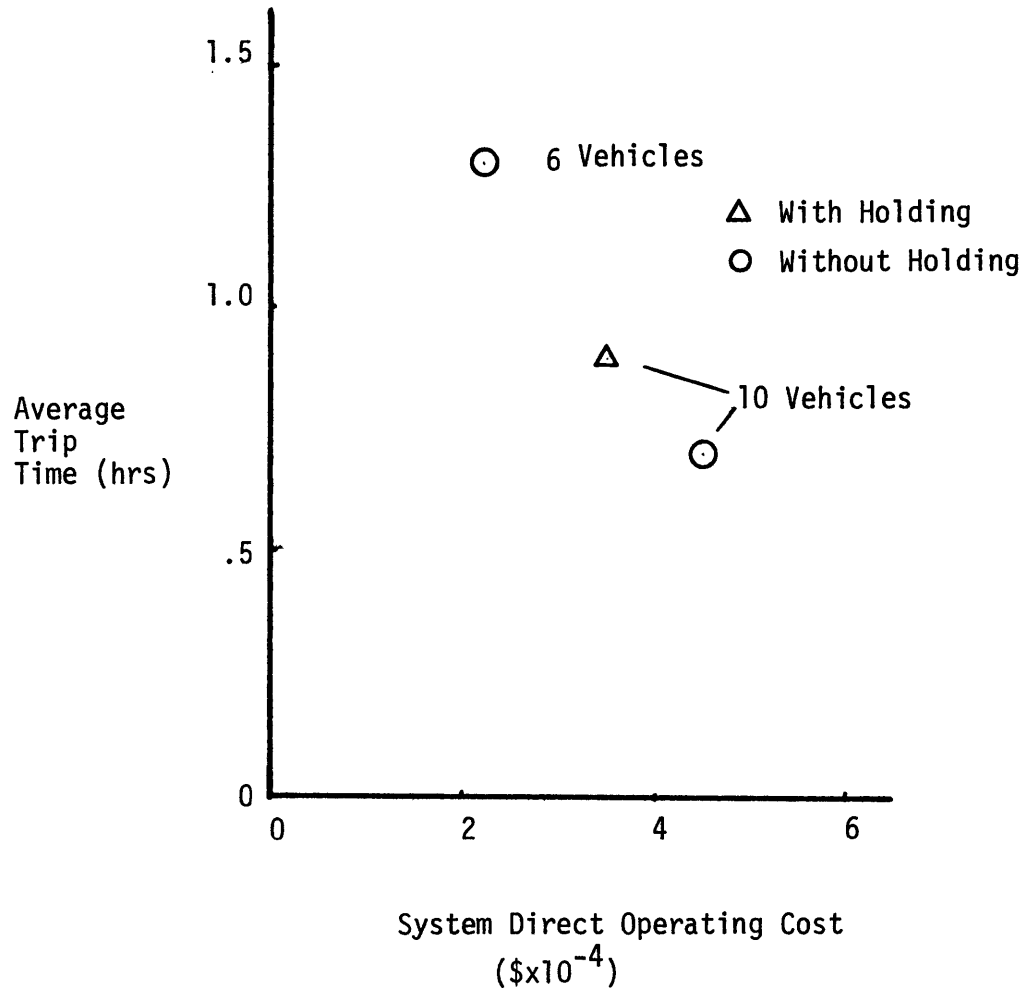


Figure 3.25 - The effect of Holding on System Performance

there may be a tremendous need for service on the second leg. This problem can be eliminated by considering the MC on the second leg of the trip also. The algebra is messy, but the result is a strategy as follows: at the first station of the two-leg trip (i-j-k), hold if

$$N < \lambda S \left[ \frac{a + \frac{b}{2} w}{a + \frac{b}{2} (w - S)} \right] + H \quad (3.16)$$

where H depends on the number waiting at j,  $S_{ij}$ ,  $S_{jk}$ , and  $\lambda_{jk}(t)$ , the arrival rate of passengers at j. Although the correction should help this strategy, it was not implemented.

It was decided that further work should be concentrated on the final holding strategy, which is more justifiable on both mathematical and intuitive levels.

### 3.3.2.3 Holding based on waiting and operating costs

The objective of the strategy developed here is to minimize the sum of the daily total vehicle operating and passenger waiting costs. Since a nearly constant number of passengers are served each day, this objective is equivalent to minimizing the average cost per passenger. This cost is the sum of the waiting and the operating cost associated with service. The expected value of both these costs can be found in terms of the headway,  $h$ , in which a passenger arrives. As was derived earlier, the expected cost per passenger arriving in a headway of length  $h$  is,

$$ah/2 + bh^2/6$$

where a and b are the parameters of the individual cost function (Figure 3.19). The expected direct operating cost per passenger is approximately the total cost of flying from the origin to destination divided by the expected number of passengers on board.

$$\begin{aligned} \text{Expected (Operating Cost)} &= \frac{\text{DOC \$/cycle} + \text{DOC \$/f hr(flight time)}}{\text{Expected number of arrivals in h}} \\ &= c/h\lambda(t) \end{aligned}$$

Let z be the sum of these costs per passenger, z is minimized by setting the derivative w.r.t.h to zero.

$$z = \frac{a}{2} h + \frac{b}{6} h^2 + c/h\lambda(t)$$

$$\frac{\partial z}{\partial h} = \frac{a}{2} + \frac{b}{3} h - \frac{c}{h^2\lambda(t)} = 0$$

or

$$\frac{b}{3} h^3 + \frac{a}{2} h^2 - \frac{c}{\lambda(t)} = 0 \quad (3.17)$$

To take advantage of the knowledge of actual system conditions, the following replacements are made,

$$\begin{aligned} h &\approx w & w &= \text{maximum wait} \\ \lambda(t) &\approx N/w & N &= \text{number waiting} \end{aligned}$$

Substitution into (3.16) yields

$$\frac{b}{3} w^3 + \frac{a}{2} w^2 - \frac{c}{N/w} = 0$$

or

$$\frac{b}{3} w^2 + \frac{a}{2} w - c/N = 0 \quad (3.18)$$

a quadratic in  $w$  instead of a cubic in  $h$ , computationally a very fortunate side effect. The optimal maximum wait is then found using the quadratic formula.

$$w^* = \frac{-a/2 + \sqrt{a^2/4 + 4bc/3N}}{2b/3} \quad N, b > 0 \quad (3.19)$$

$$w^* = 2c/aN \quad \text{for } b = 0, N > 0$$

Once again, this is a simple rule to implement for a particular OD pair. When a departure is scheduled to take place,  $w^*$  is determined, if the maximum wait for this group is less than  $w^*$ , and the number waiting is less than the vehicle capacity, the departure is postponed for some  $\Delta t$ .

Since this algorithm is to be used in conjunction with the dynamic vehicle routing algorithm, a correction for the myopic character of the holding rule is considered. Suppose the vehicle is scheduled to fly  $i$ - $j$ - $k$  and is scheduled to leave  $i$  at time  $t_{\text{now}}$ . The decision to hold at  $i$  needs to consider conditions at  $j$  as well as at  $i$ . At  $t_{\text{now}}$ ,  $w_{ij}^*$  is calculated, and the optimal time to begin the first leg of this journey,  $t_i^*$  is determined. Also, while at  $i$ , the best time to begin

the second leg of the trip,  $t_j^*$  can be found by calculating  $w_{jk}^*$ . Based on these two values, the decision to hold at  $i$  can be considered as one of the following cases.

1. If  $t_{\text{now}} \geq t_i^*$ , the vehicle leave immediately, regardless of the conditions at  $j$  (see Figure 3.26a).
2. If  $t_{\text{now}} < t_i^*$ , AND  $t_i^* + S_{ij} \leq t_j^*$ , hold the vehicle at  $i$ . In this case, both best departure times can be met (see Figure 3.26b).
3. If  $t_{\text{now}} < t_i^*$  AND  $t_i^* + S_{ij} > t_j^*$ , both departures cannot occur at the best times (see Figure 3.26c).

Clearly, in this case, either one or both of the departures will miss the best time. We will define  $t_e$  as the total time by which both departures miss the best times.

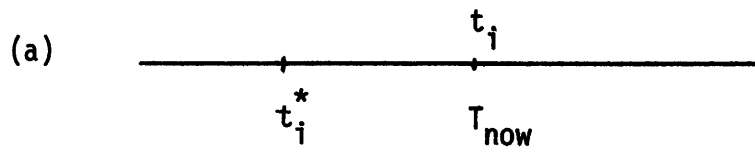
$$t_e = t_i^* + S_{ij} - t_j^*$$

$$t_e = t_{e_i} + t_{e_j}$$

The departure from  $i$  will occur  $t_{e_i}$  early and the departure from  $j$  will be  $t_{e_j}$  late. The total error is divided between the OD pairs based on the expected number of passengers affected at each station, such that the bigger error affects the fewest number of passengers (see Figure 3.26d).

$$t_{e_i} = t_e \left( \frac{N_{jk} + \lambda_{jk} S_{ij}}{N_{ij} + N_{jk} + \lambda_{jk} S_{ij}} \right) \quad (3.20)$$



Time  $\longrightarrow$ 

( $t_i$  = Time of actual departure from node  $i$ )

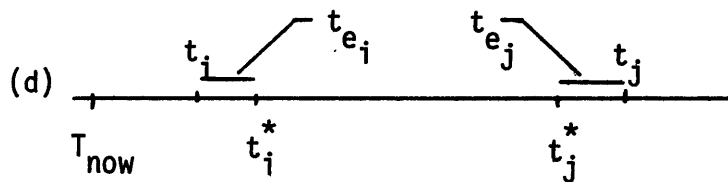
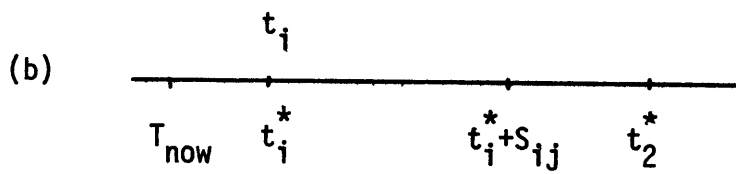


Figure 3.26 - Illustration of Two-Stop Holding

To implement this rule, we need to determine the values of the parameters  $a$ ,  $b$  and  $c$ . For this case,  $a = 10\$/\text{hr}$  and  $b = 10\$/\text{hr}^2$ . Now, what does it cost to fly rather than hold? The direct operating cost is made up of flight time and cycle costs. The cost per hour is the sum of depreciation, insurance, crew, fuel and maintenance costs. If the vehicle is holding ready to fly, the system is still paying for depreciation, insurance and crew. The marginal cost of flying rather than holding is not the total DOC, but rather some fraction of it. This fraction is approximately  $1/2$ . The parameter  $c$  is then given as follows:

$$c = 100 (\$/\text{cycle}) + (300\$/\text{f-hr}) (\text{Flight time})$$

(Note that the flight time is different for each OD pair, and, therefore, so is the cost of service.)

Using this holding rule in conjunction with the dynamic vehicle routing algorithm, the system is simulated with a fleet of 10 vehicles. The results are shown in Table 3.9.

10 Vehicle Nonstop	Average Trip Time	# Cycles	F <sub>hours</sub>	DOC\$
Dynamic Routing with Holding	.87	186	43	44,000
Dynamic Routing No Holding	.75	269	60	63,000
Fixed Routing	.89	275	62	65,000

Table 3.9

It can be seen that the effect of dynamic versus fixed routing is to decrease average travel time. Now, the vehicle holding permits a tradeoff between average time and operating cost. In fact, the total cost is cut 40%, with no loss in LOS, from the fixed to holding case. Next, the system is simulated using the holding strategy in conjunction with the one-stop service option of the routing control. Once again, a significant decrease in operating cost is achieved with a modest increase in travel time. The results are summarized in Table 3.10.

One-Stop	Average Trip Time	# Cycles	F <sub>hours</sub>	DOC
6 Vehicles	1.39	119	29	29,000
10	.87	184	40	43,000
15	.77	206	46	49,000

Table 3.10

Notice that in the 15 vehicle case the total number of takeoff and landing cycles is cut by over 200 in the 8 hour simulation run. At a cost of 100\$ per cycle, this is a savings of 20,000\$. In each case, this strategy provides a large saving in direct operating cost while providing comparable LOS to the passengers.

An investigation of the sensitivity of average trip time and total DOC to the parameter  $c$  yields additional insight into the workings of the system and holding algorithm. In setting the value of  $c$ , the actual cost of service, it is estimated that this cost is 1/2 the

original DOC. If instead of  $1/2$ , this value is  $1/3$ , shorter headways and greater total operating cost should result. Similarly, if the actual cost is  $2/3$  the original DOC, larger headways and reduced total operating cost should result. By varying this fraction, the tradeoff between LOS and cost can be controlled. The one-stop system with 10 vehicles is simulated once again, this time  $c$  is varied. The results are shown in Figure 3.27.

### 3.4 Summary of Results

The purpose of this study is to investigate the effects of general strategies on USH performance and to develop new routing schemes, making the USH system more efficient, through an improvement in LOS, and a reduction in operating cost. The control strategies which evolved in the course of this study do achieve these goals. In fact, the improvement in USH operations is very impressive. This can be illustrated by comparing the LOS provided to passengers by the system operating a 10 vehicle fleet, under various controls. From Figure 3.28, it is obvious that the fixed routing system can not cope with the peak demand. In neither the allstop nor the nonstop cases is a stable condition achieved during the rush. This means that if the rush lasted for another hour, average travel time would continue to increase. The effect of the demand peaking is significantly reduced in the dynamic routing cases. In the case without holding, the average travel time increases during the rush, levels off, and then decreases again after demand subsides. The effect of holding is to decrease this variation of travel time over the day.

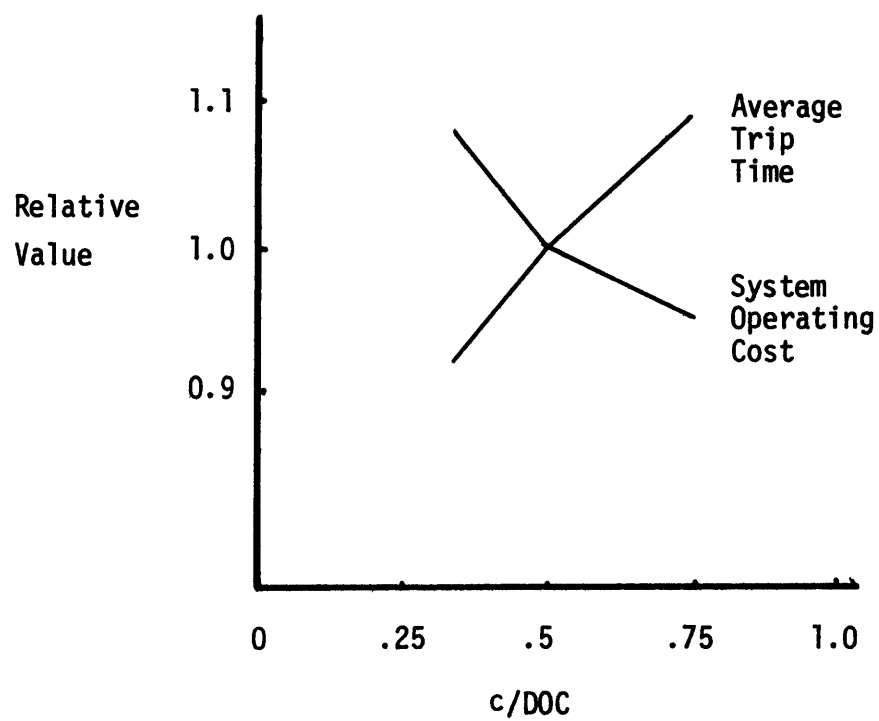


Figure 3.27 - Sensitivity of LOS and DOC to  $c$

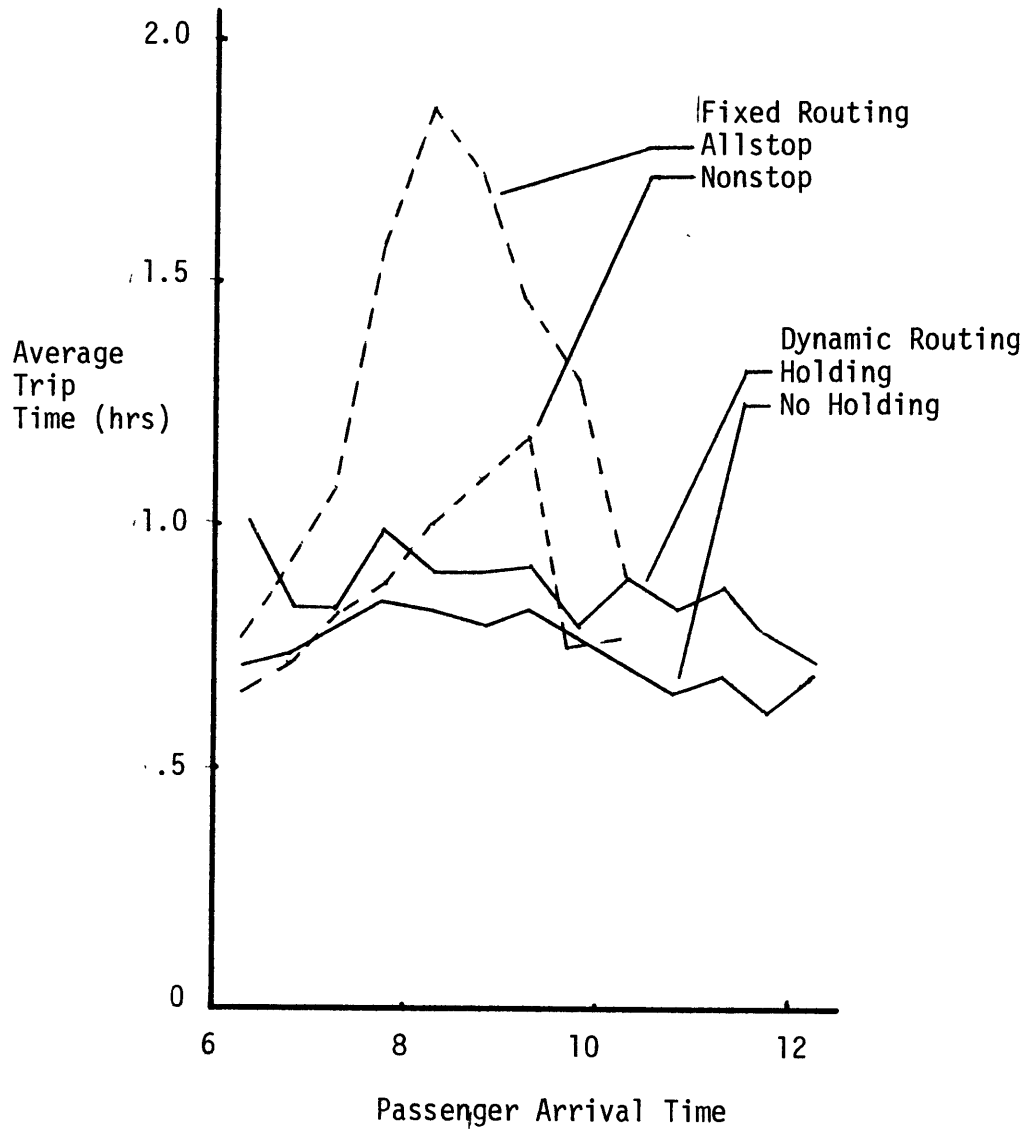


Figure 3.28 - Variation of LOS With Time of Day, 10 Vehicle Cases

In the holding cases, it is difficult to tell where the rush hour begins and ends, from variations in the travel time. In terms of LOS, this is a very stable system.

How much does it cost to run this system? Consider first the variation in average travel time with the number of takeoff and landing cycles (cycle costs are a major portion of total system costs)(Figure 3.29). In all but the holding cases, the number of cycles and the operating cost of the system depend only on the number of vehicles. The effect of the holding strategy is to reduce this cost and slide the curve to the left. As it turns out, the most stable system is also the cheapest to operate! Also, the variation in average travel time with operation cost is shown (Figure 3.30). The Pareto-optimal boundary here is defined by the system operating vehicles using dynamic routing and holding. More points on this boundary could be found by varying the cost parameter in the holding rule.

One item concerning the dynamic routing algorithms developed in Section 3.3 should be stressed. The use of nonlinear waiting costs in this algorithm is not meant to reflect passenger psychology. These costs are strictly a modeling tool, and one which proved to be very useful. By varying the relative values of the cost parameters, the tradeoff between the mean and the variance of trip time can be controlled. Intuitively, it seems reasonable that reducing the uncertainty in trip time increases the LOS to passengers. In the next chapter, a passenger-oriented model is postulated which provides more justification

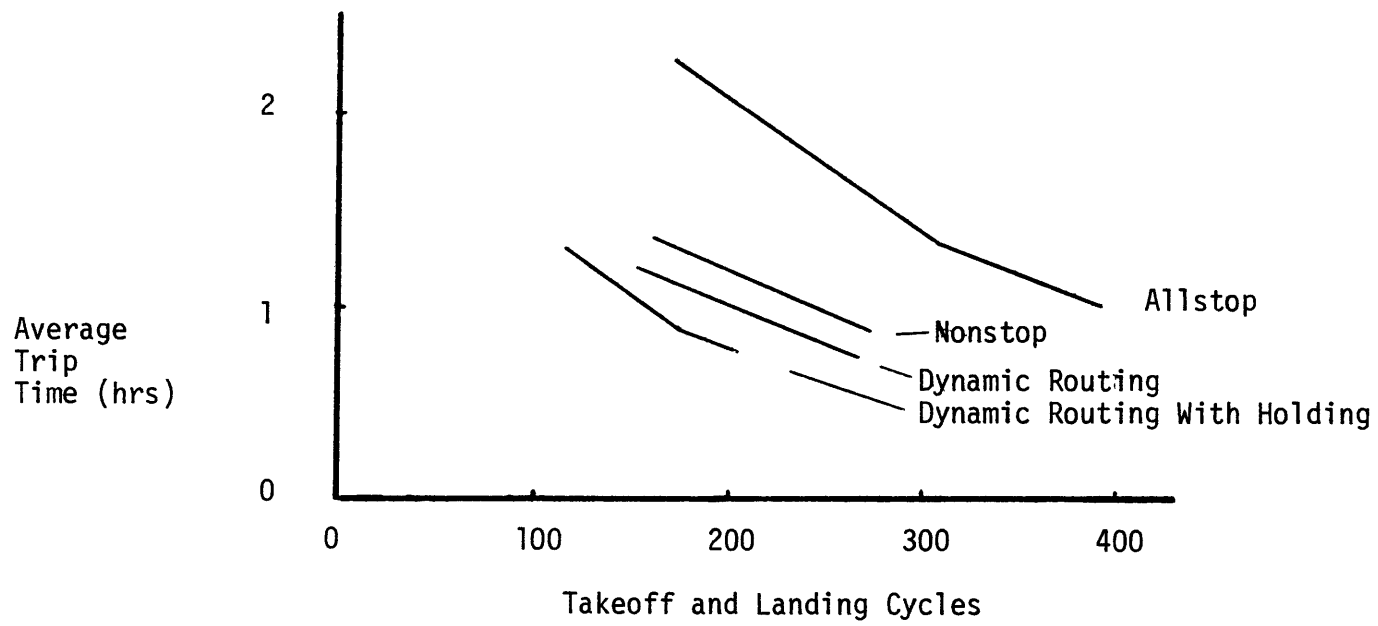


Figure 3.29 - LOS vs. Number of Takeoff and Landing Cycles



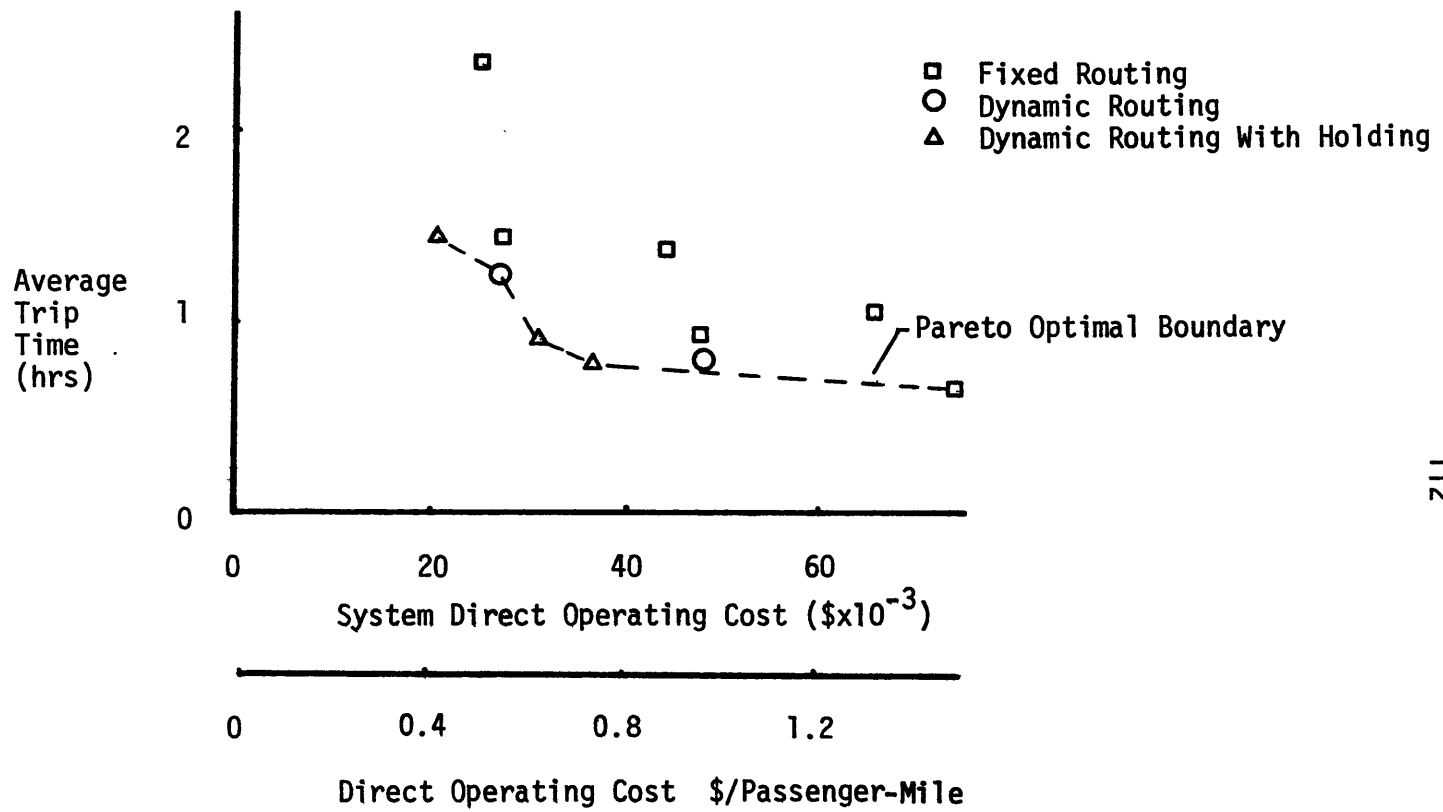


Figure 3.30 - System Performance: LOS vs. DOC

for controlling this uncertainty, and makes use of the algorithm's ability to do so.

The most significant aspects of the dynamic control algorithm is its generality and simplicity. The fleet of vehicles was routed and departures were scheduled using a minimum amount of real time data. The algorithm has no prior information as to actual passenger arrivals or even the average arrival rates of passengers. In fact, the only prior information the control algorithm has is a distance matrix and vehicle parameters. The real time information which is used is simple. For each OD pair, two numbers (number waiting and maximum waiting time) are required. In the simulation model, this is the only information the control algorithm uses. In fact, if an actual USH system existed, this algorithm could be used without modification to control it. The real time information necessary to drive the algorithm can be collected by counting and timing passenger arrivals at each station. This information can be relayed to a central facility via phone lines. The route choice calculation can be made there. These calculations would require a machine no larger than a programmable calculator. Instructions (next destination, departure time) can then be relayed back to the stations.

### SUGGESTED TOPICS FOR FURTHER RESEARCH

This study was undertaken in order to develop a scheduling and routing methodology for the USH system, and to model realistically the operations of the system. The results of this work are to be used in a more general study of USH system feasibility. In this chapter, topics for further research aimed at determining USH feasibility are suggested. This additional research consists of three general areas. The first area is the immediate application of the existing models to the investigation of USH system sensitivity. Next, modifications of the algorithms and of the simulation procedure are suggested in order to make the models more realistic in a USH analysis. Finally, some questions involving fundamental transportation issues are raised.

#### 4.1 Immediate Applications of Models

With the existing algorithms and present version of SMUSH, many aspects of USH system design can be investigated. A few of the more pertinent questions, which may be answered through the use of the existing software, are listed below.

1. What is the effect of city geometry? How are USH operations and system performance affected by the relative locations and size of the suburbs and CBD?
2. How sensitive is the system to demand peaking and directionality? At what level of peaking does congestion significantly degrade

LOS? How is this level affected by vehicle parameters, the number of stations, or individual passenger cost parameters?

3. What is the relationship between the mean and the variance of trip time? Does the effort to reduce uncertainty in trip time increase the likelihood of congestion?

#### 4.2 Modifications to Models

Multiple stop service and load building is an important feature of the USH system, but so far, it has not been fully exploited. While the dynamic control algorithms allow some one-stop service, only non-stop service is considered in determining vehicle routing. Although the modifications necessary to consider explicitly one and two-stop service will add to the complexity of the decision algorithm, the improvement in system performance should make this work worthwhile.

Throughout this study, it is assumed that demand is independent of the schedule. Obviously, this is not a realistic assumption. The probability that a particular passenger chooses the USH system (and therefore the total demand) depends on fare, convenience, and LOS. So, in a given system with some potential demand, a particular schedule will result in some actual daily demand. But the schedule generated by the algorithms developed in this study in turn depends on the actual demand. There is an equilibrium demand and schedule, but due to the complexity of the problem, they cannot be found analytically. The equilibrium conditions can be found using the following iterative method. Some LOS is assumed, the associated demand is estimated, and the system is

simulated. If the estimated and actual LOS differ significantly, LOS is re-estimated and the system is simulated again. Although this procedure should converge to the equilibrium condition, it is not obvious how fast this will occur. So, while this method will yield a good estimate of USH operations, it may be prohibitively expensive.

#### 4.3 Fixed vs. Dynamic Scheduling

With the algorithms developed in this study, the system operates with no timetable (or with a dynamic schedule). This type of policy gives the system the ability to cope with random fluctuations as well as with (unanticipated) overall variations in demand. A system with a fixed departure schedule loses this ability. But availability of a timetable to passengers should reduce the average waiting time throughout the system. The tradeoff between fixed and dynamic scheduling can be evaluated by comparing the effects on LOS of the reduction in waiting time associated with the availability of timetables, and the benefits associated with the ability to cope with the variations in actual demand.

Although the average arrival rate of passengers into a scheduled system depends on the times of departure, the actual times of arrival are random (see Figure 4.1). The reduction in average waiting time to the passengers depends on the amount of influence of the schedule on the arrival rate. This dependence can be estimated by observing the arrival rates of passengers into existing transportation systems operating with a fixed schedule and relatively short headways.

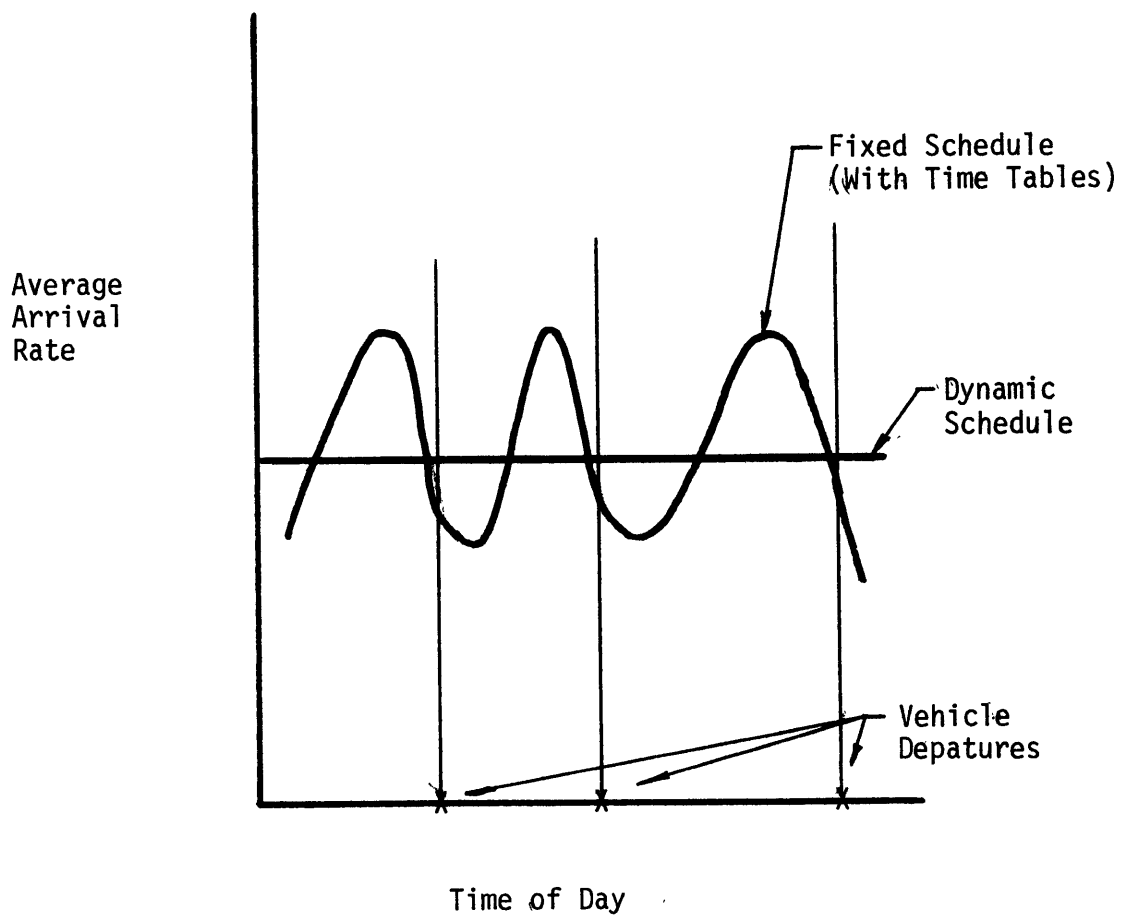


Figure 4.1 - Illustration of the effects of time tables on the average arrival rate of passengers into a transportation system.

Examples of this type of system include the Eastern Shuttle, NYAW, and some bus systems.

The effect on LOS of the system's ability to cope with variations in demand can be estimated with the existing computer models. A fixed schedule can be generated by running the system dynamically and recording the vehicle movements and departure times. This same schedule can be used on other days, where the overall demand patterns are the same, but where individual passenger arrivals are different. The effect of the loss of flexibility can be seen by comparing the LOS on the first and subsequent days.

This analysis should provide a reasonable answer to the fixed versus dynamic scheduling question. But whatever the outcome, the vehicle control algorithms and simulation model can still be used to generate the schedules.

#### 4.4 LOS

Throughout this study, the term LOS has been used as a measure of system performance. It was suggested in the first chapter that this term depends on both the mean and variance of travel time. A justification for this conjecture, and a proposed measure for LOS are now presented.

LOS is a measure of the cost, in time, associated with travel. This generally refers to the time spent in transit. The maximization of this measure of LOS is equivalent to the minimization of average travel time. But there is also a cost to the passenger associated with the

actual arrival time at the destination. Consider a businessman who must meet with a client at a certain time. There is a cost in loss of goodwill or actual dollars associated with being late (see Figure 4.2a). A general traveler also has an arrival cost (Figure 4.2b). Here, the costs are much lower, since the ramifications of being late to a movie or department store are not as great as those associated with the business meeting. Note that there is also a cost associated with being early, for an early arrival may result in waiting outside for a business to open or a long wait to enter a theater, etc. Obviously, these costs vary with the individual and passenger type, but for a class of travelers, these general forms should apply.

Now, consider an individual commuter who has determined the cost spent in travel to be  $C$  \$/hour, and the time by which he arrives early to be worthless. For example, if he allows one hour for the trip to work and arrives early, the total cost is  $\$C$ , regardless of the actual arrival time. Furthermore, suppose that he has determined an arrival cost of the simplified form shown in Figure 4.3.

The total cost of making the trip (TC) is then,

$$TC = C \cdot (T.A.) + L\delta \quad (4.1)$$

where  $TC \equiv$  total cost

$TA \equiv$  time allowed

$$\delta = \begin{cases} 0 & \text{early} \\ 1 & \text{late} \end{cases}$$



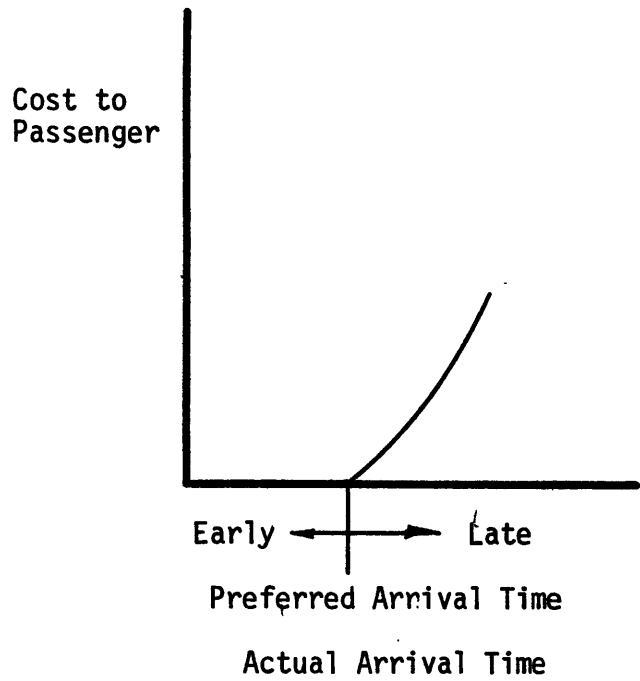


Figure 4.2a - Arrival Cost for Commuters

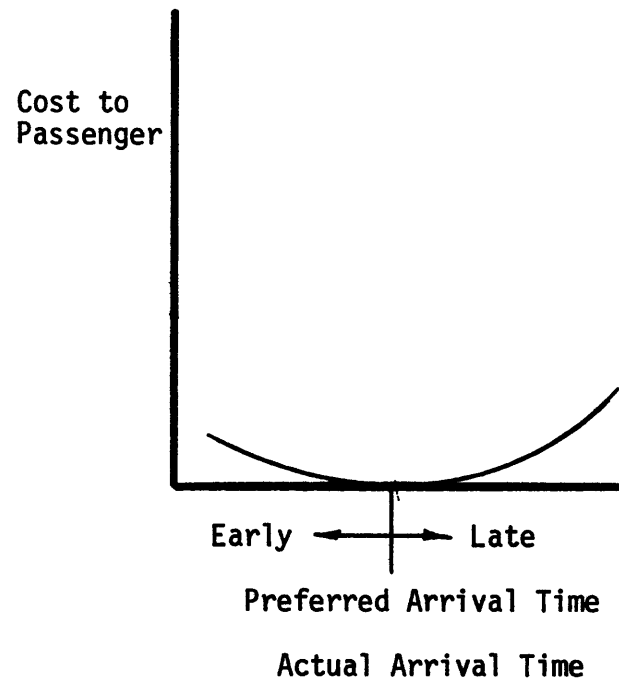


Figure 4.2b - Arrival Costs for General Travelers

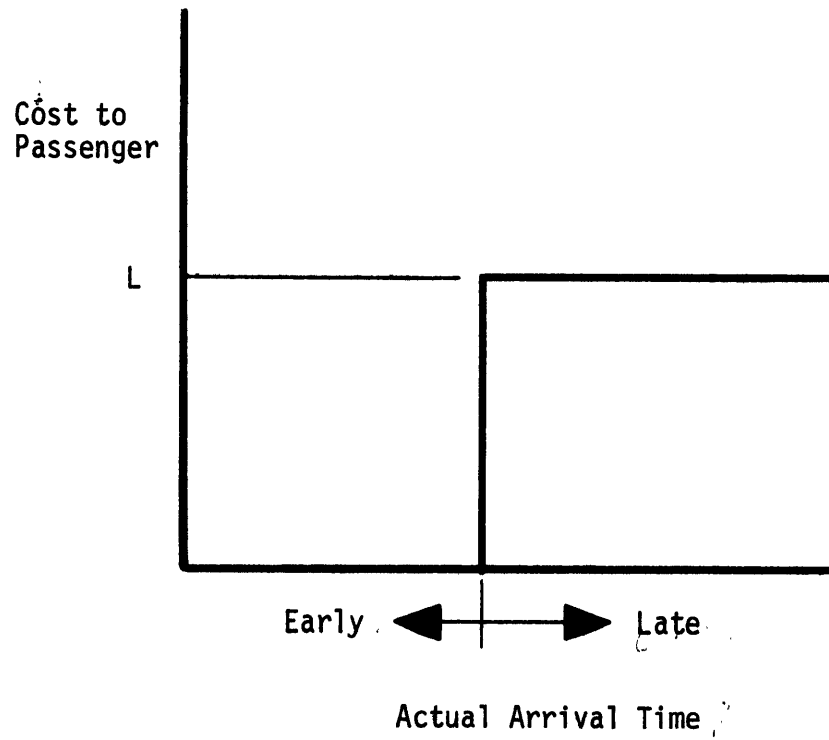


Figure 4.3 - Simplified Commuter Arrival Time Cost

In the long run, it seems reasonable to assume that this commuter will want to minimize average cost, which is found as follows:

$$E(\text{Cost}) = C \cdot (\text{TA}) + L \cdot P(\text{Late}) \quad (4.2)$$

The probability of being late,  $P(\text{Late})$ , is the probability that the actual trip takes longer than the time allowed.

$$\begin{aligned} P(\text{Late}) &= P(\text{Trip Allowed} < \text{Trip Time}) \\ P(\text{Late}) &= 1 - P(\text{Trip Time} < X_0) \quad X > 0 \end{aligned} \quad (4.3)$$

If the commuter knows the cumulative distribution function for his trip time (Figure 4.4a), he can determine  $P(\text{Late})$  as a function of time allowed (Figure 4.4b). With this information, he can calculate the expected total cost in terms of time allowed. Finally, he can determine the amount of time to allow ( $\text{TA}^*$ ) in order to minimize expected cost (Figure 4.5).

Now, while the precise roles of mean and variance in the minimum cost depend on the trip time cdf, it should be clear that minimum cost depends on both terms.

For a system with fixed average travel time as the variance increases, the minimum expected cost associated with travel will also increase. In Chapter 3, it was shown that for a particular system using dynamic vehicle controls the effect of decreasing the variance of travel time is to increase the mean. This leads to a most interesting optimization problem. At what point should the system operate in order to maximize the average LOS offered

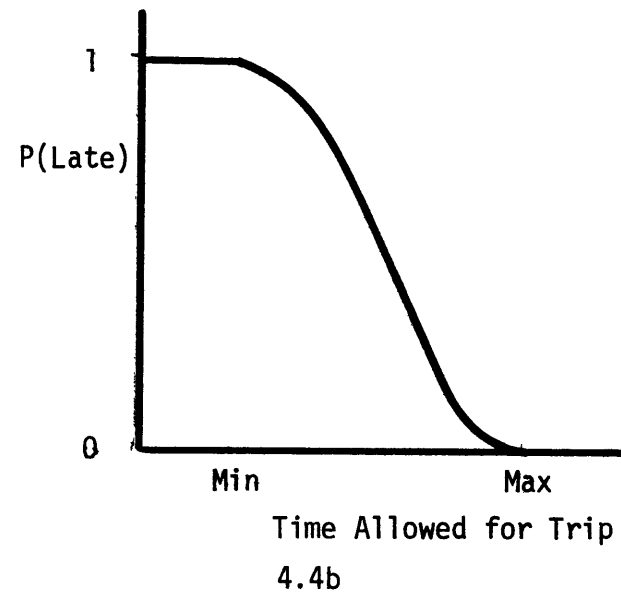
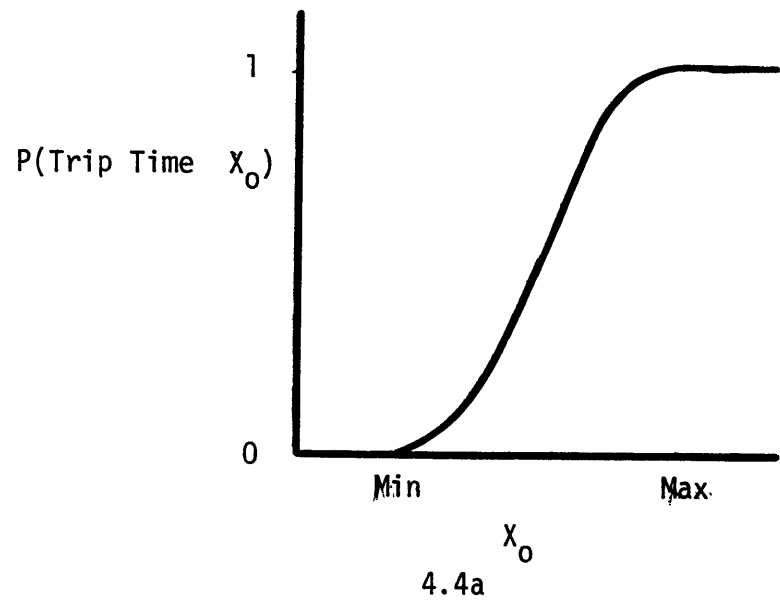


Figure 4.4 - Trip Time CDF and the Probability of Arriving Late

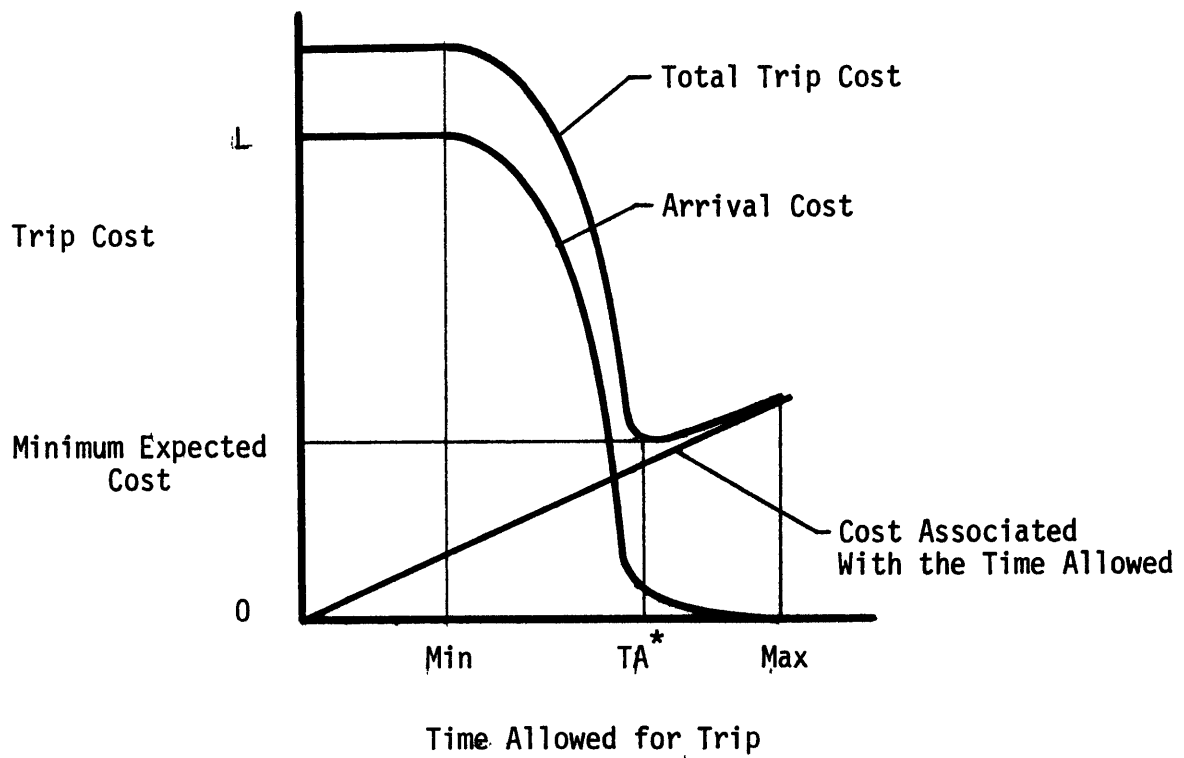


Figure 4.5 - Total Cost Associated with the Time Spent in Transit and the Time of Arrival

to passengers? Recall that this operating point is controlled by the slope of the individual waiting cost curve,  $b$ . So, if the costs associated with travel and arrival times are determined, can the value of  $b$  be determined to maximize average LOS to all passengers? If all passengers had this information and made the correct decisions, the minimum expected cost would be a good measure of LOS. Obviously, passengers do not have this complete information, but this is reasonable passenger behavior. While it is not obvious whether this measure of LOS is mathematically tractable, it may be an interesting concept to pursue, in the context of incomplete or bad information.

In this section, methods of generating an optimal schedule for the USH system are briefly discussed. By optimal, we mean the maximization of LOS to all passengers. A method developed by Devanney for ground transportation, generates an optimal schedule in two steps.<sup>11</sup> First, the network is decomposed into individual OD pairs, and optimal departure schedules for each pair are determined using Dynamic Programming. Next, the network is reconstructed, and the minimum number of vehicles necessary to achieve the departure schedules for the entire network is determined, using Simpson's fleet reduction program (REDUCTA).<sup>12</sup> If it is assumed that the demand for a given departure depends on the LOS as well as time of day, the decision variables in the Dynamic program are the number of departures, the time and the capacity of each departure. It was shown by Vom Saal that if the dependence of demand on the schedule is eliminated, and if the vehicles have sufficient capacity to serve all passengers waiting for each departure, then the number of decision variables necessary in the optimization is reduced to one.<sup>13</sup> This represents a tremendous simplification in the optimization process. This method works well and can be used to generate a schedule in a large network, but it does not accurately model, or fully exploit the capabilities of the USH system.

Next, a shortest path formulation of the USH scheduling problem is described. This formulation will utilize Vom Saal's simplifications.

The vehicle scheduling and routing are considered simultaneously in this optimization process. It is assumed that the demand is known, and is independent of schedule. And for now, it is assumed that vehicles have unlimited capacity. Due to the explosive nature of the state space, it is difficult to obtain the optimum in even a moderately-sized network. The technique is demonstrated on a trivial system, and extensions to a more realistic network are then discussed.

Consider the network shown in Figure A.1a; passengers arrive at nodes 1 and 2 at known rates of arrival. They are transported to their common destination, node 3, by a single uncapacitated vehicle. The state of the system will be defined as the time of the most recent departures from 1 and 2,  $(S_1, S_2)$ , or equivalently, as the time at which the last service to each OD pair began. The initial state of the system is defined to be  $(S_1 = 0, S_2 = 0)$ . For simplicity, only discrete values of  $S_1$  and  $S_2$  are considered. With this restriction, the state space can be represented by a two dimensional array of points (see Figure 2.1b).

As the day progresses, each service to the OD pairs redefines the state of the system. The system state changes via a transition which is defined to occur when the vehicle stops at node 3. There are only four combinations of stops by which the vehicle can leave 3, pick up passengers at one or both origins, and return to 3. Each combination corresponds to a transition (Figure A.2).



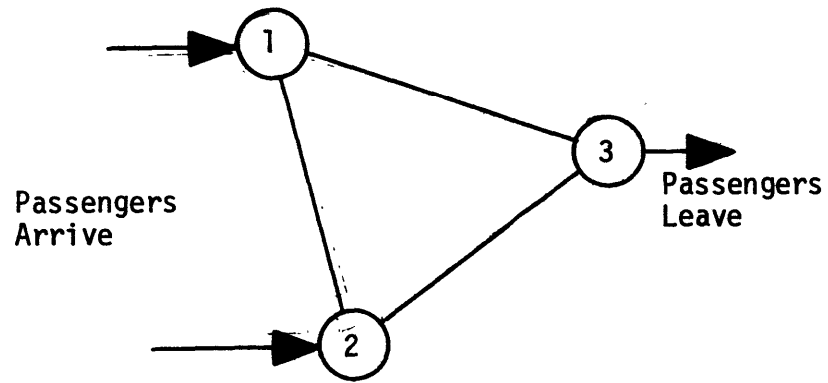


Figure A.1a - Sample Network

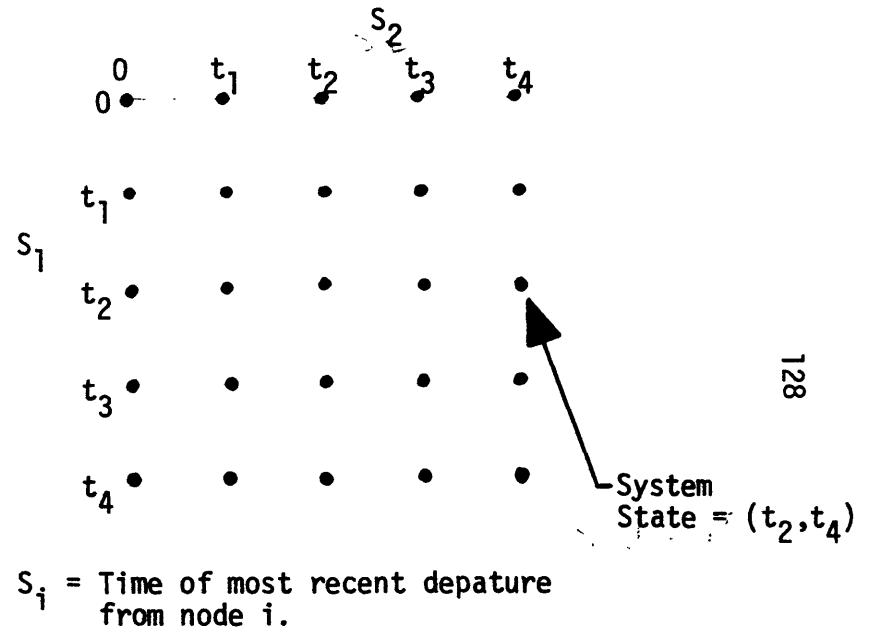


Figure A.1b - State Space

<u>Transition Type</u>	<u>Vehicle Routing</u>
A	3-1-3
B	3-1-2-3
C	3-2-1-3
D	3-2-3

Figure A.2

Note that passengers can receive either nonstop or one-stop service.

Suppose that the system leaves state  $(S_1, S_2)$  via a transition of type B (this corresponds to the vehicle leaving 3, stopping at 1 and 2, and then returning to 3). Let the departure times from 1 and 2 be  $S_1^i$ , and  $S_2^i$ , respectively. Assuming that 1 and 2 are distinct nodes and that the vehicle moves at a finite speed, there is a minimum time  $a_{12}$ , required to fly between the two stations. (In general,  $a_{ij}$  is defined to be the minimum time between departures from  $i$  and  $j$ ,  $a_{ij}$  = block time + minimum ground time). Therefore, the following constraint can be placed on the departure times:

$$S_2^i \geq S_1^i + a_{12}$$

Suppose that the system entered its present state via a transition of type A. The vehicle routing for the present and most recent transitions can be represented schematically, as in Figure A.3. From this figure, it is obvious that  $S_1^i$  depends on  $S_1$ . The constraint is as follows:

$$S_1^i \geq S_1 + a_{13} + a_{31} = S_1 + 2a_{13}$$

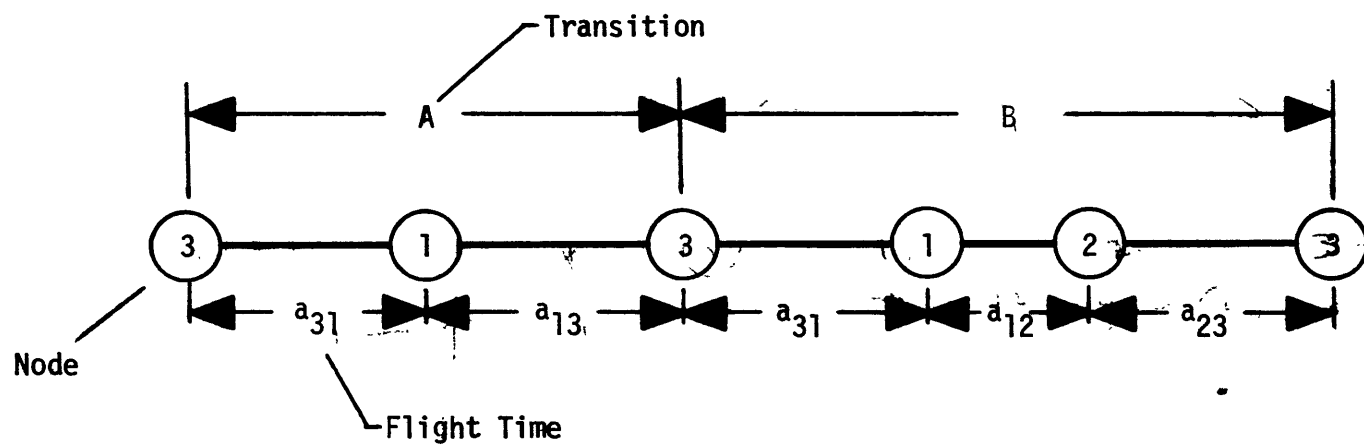


Figure A.3 - Timeline for A, B Transitions.

In general, there are constraints on the times of future departures from each node, which depend on  $S_1$ ,  $S_2$  (present system state) and the vehicle routing (transition type). But it can be shown that these constraints do not depend on how the system reached its present state.<sup>14</sup> In summary, the states which the system may next enter depend on the present state ( $S_1$ ,  $S_2$ ) and the transition type by which this state is left.

There is one additional constraint; all passengers must be served. This requires that exactly one departure occur after the last passenger arrives at each node.

Consider the following example. Passengers arrive at nodes 1 and 2 at rates  $f_1(t)$ , and  $f_2(t)$  (Figure A.4a). The minimum times between departures are  $a_{12} = .2$ ,  $a_{13} = .3$ , and  $a_{23} = .4$  (hours). Discrete values of  $S_1$  and  $S_2$  are considered at .1 hour intervals. The vehicle departure schedule is as shown below (Figure A.4b).

<u>Node</u>	<u>Departure Time</u>	<u>Transition</u>
2	0.20	C
1	0.40	
3	0.80	B
1	1.20	
2	1.40	D
3	1.70	
2	2.00	A
3	2.30	
1	2.70	

Figure A.4b

The path of the system through the state space is shown in

Figure A.4c. It should be noted that the final state of the system satisfies the constraint,  $S_1 \geq 2, S_2 \geq 2$ . Also note that the transitions in which the vehicle stops at one node to pick up passengers are parallel to the axes, while the transitions in which the vehicle stops at both nodes 1 and 2 are diagonal.

There is a cost in passenger-hours to the system in jumping from state  $S_1, S_2$  to  $S_1', S_2'$ . This cost is the sum of the waiting and ride times for those passengers served on this transition. The total waiting time associated with the transition is

$$\sum_{i=1}^2 \int_{S_i}^{S_i'} f_i(x)(S_i' - x) dx$$

The ride time for the passengers also depends on transition type since the transition type determines whether a passenger receives one-stop or nonstop service.

Given that the system is presently in state  $S_1, S_2$ , the constraints on future departure times determine which states the system may next enter, and the cost associated with each transition (pax-hrs) can be found. If the state space is considered a network, and the cost associated with each possible transition a distance, then the optimal schedule is the shortest path from the initial state to some final state  $S_1'', S_2''$ , where  $S''$  indicates a time after the last passenger arrival.

Now, this schedule can be generated using a shortest path algorithm. To determine an eight-hour schedule with time increments

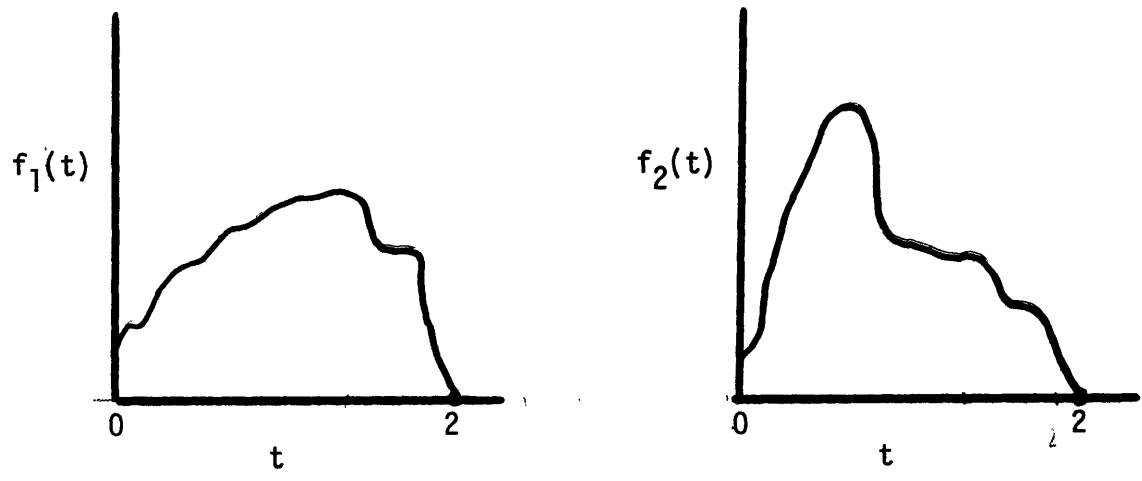


Figure A.4a - Passenger Arrival Rates

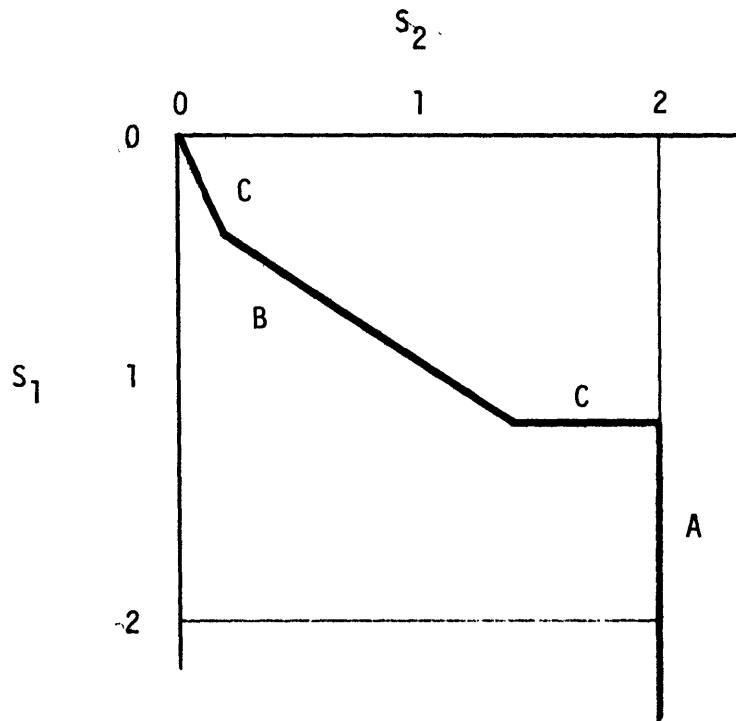


Figure A.4c - System Path Through the State Space

of .1 hours, the network necessary contains about 6400 nodes. This can be solved on the computer with a small amount of storage and time, but this is still a trivial problem. More meaningful results come at the cost of a much larger state space and associated network.

Finite vehicle capacity can be modeled with the addition of a variable corresponding to an overflow number of passengers for each OD pair. For the two OD pair case, the system state can be defined as  $(S_1, S_2, W_1, W_2)$ , where  $W_i$  is the number of passengers at node  $i$  who were not boarded on the last departure, due to capacity constraints. The initial state of the system is  $(0, 0, 0, 0)$  and the final state is  $(S_1'', S_2'', 0, 0)$ . Although this is a simple addition conceptually, the extra variables add two more dimensions to the state space. If values for  $W_i$  are considered in a range of 0 to 100, the network from the previous example would now have  $64 \times 10^6$  nodes. This is no longer a simple problem for computer application of the shortest path algorithm. A more reasonably sized USH system could have 10 OD pairs. The state of the system would now be defined in 20 dimensions. The size of the state space would be  $(100 \times 80)^{10}$  or around  $10^{39}$ . The addition of multiple vehicles adds still more complications to this already astronomical state space.

This formulation is a victim of the "Curse of Dimensionality." The fact that the entire state space must be stored to obtain the optimal solution using the shortest path algorithm makes computer applications impossible. So, while the determination of an optimal schedule using Dynamic Programming (in the form of the shortest path) is conceptually feasible, it is totally impractical from the computational point of view.

## Appendix B

SMUSH User's GuideIntroduction

The Simulation Model for the Ultra Short Haul transportation system (SMUSH) was developed to investigate the effects of vehicle routing and scheduling strategies on the performance of the USH system. SMUSH is written in the GASP IIA simulation language. GASP is a collection of FORTRAN subroutines which provide the general framework for event-paced simulation. SMUSH is a FORTRAN program, consisting of standard GASP and non-GASP subroutines. The GASP portion of SMUSH provides the mechanism to store and retrieve data, advance the simulation clock, and collect statistics. The non-GASP portion is dedicated to the input, output and handling of data unique to the USH system. Due to this structure, the simulation methodology and program code are completely accessible to the user.

The user of SMUSH provides the following system parameters:

Network layout

Fleet and vehicle parameters

Passenger arrival rates

Routing and scheduling structure.

The output of SMUSH consists of the following system performance data:

Vehicle schedule map

Passenger trip time statistics

Vehicle utilization statistics



## System operating costs

Vehicle Control Options

The user of SMUSH has the option of two general vehicle control strategies, these are fixed and dynamic controls. In both options, the basic mechanism by which the vehicles are routed is the same. Each vehicle flies an assigned route. When this route is completed, the vehicle immediately begins the completed route's successor. The minimum time to complete each takeoff and landing cycle is assumed to be a function of the number of passengers boarded, see Figure B.1. (The actual cycle time may be increased by holding.)

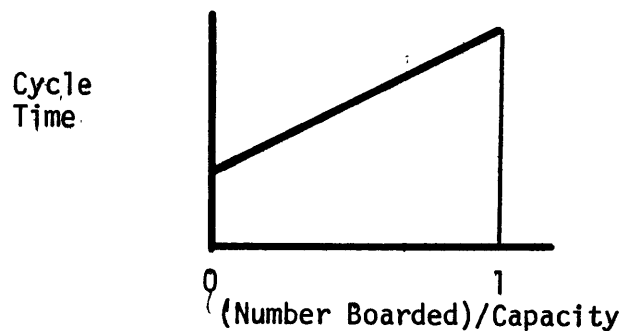


Figure B.1 - Takeoff and Landing Cycle Time

At each stop, the vehicle normally boards all passengers whose destination is on the remaining portion of the route. But if the present route is its own successor, then passengers are boarded whose destination is any node on the route. For example, if the route calls for stops at nodes  $i$ ,  $j$ , and  $k$ , at node  $j$  only,  $jk$  passengers are boarded. But if this route is its own successor, the vehicle will stop at  $i$ ,  $j$ ,  $k$  and then  $i$ ,  $j$ ,  $k$  again. So, at node  $j$ ,  $jk$  and  $ji$  passengers are boarded.

Under the fixed control option, the complete routing for each vehicle is specified by the user. The vehicle flies its initial route, this route's successor, etc. When the dynamic controls are invoked, routes are constructed for each vehicle during system operation. The routes and successors are determined according to the algorithm developed in Section 3.3.1.

If the fixed control option is used, the user must provide the following data:

- Complete list of routes
- Successor for each route
- Initial route and starting time for each vehicle
- Minimum headway
- Maximum number to be loaded on one vehicle for each OD pair.

If the dynamic control option is invoked, the user must provide the following data:

- Passenger cost parameters
- Initial node and starting time for each vehicle.

The user must also specify whether all passengers will receive nonstop service or if some one-stop service is to be provided, and whether the vehicles will hold according to the algorithm described in 3.3.2.3.

#### Inputs to SMUSH

For each run, input is made in two parts; these are model controls and individual case data. The model controls appear once per run, the case data is input for each case to be run.

The first three lines of input are the model control data, these are as follows:

Number of cases to be run ( $\leq 10$ ) Format (I5)

Number of Monte Carlo iterations for each case Format (10I5)

Begin and end time for each case (2F5.2)

The individual case data is made in ten sections (See Figure B.2). The first case must contain an entry for each section (even if default values are used). In the following cases, only those sections in which inputs change between successive cases need to be entered. In this manner, some duplication of input data is avoided. Each case must begin with a title card and end with a "99" card.

#### Title Card

- 01 Network data
- 02 Passenger arrival data
- 03 Minimum headway
- 04 Maximum loading of each OD pair
- 05 Routing option and passenger cost parameters
- 06 Vehicle parameters
- 07 OD pair identification
- 08 Output parameters
- 99 End of case

Figure B.2

The data input in each section is as follows:

- 01 Number of nodes ( $\leq 10$ ) Format (I5)  
Distance Matrix (miles) Format (10F5.0)

- 02 Number of Arrival Rate Tables Format (I5)  
 Arrival Rate Tables (passengers/hr) Format (10F5.0)  
 Number of Passenger Types Format (I5)  
 For each Passenger Type - The Number of Data Points Format (I5)  
 For each Data Point - A Time and Corresponding Table ID Format  
 (F5.2, I5)
- 03 Minimum Headway (hours) Format (F3.1)
- 04 Maximum Boarding for each OD Pair Format (I2I5)  
 Default: -1, No Maximum, Format (I5)
- 05 Passenger Waiting Cost Parameters a, b Format (2F10.7)  
 Routing Options NRTS, MHOLD Format (2I5)  
 If NRTS > 0, NRTS = Number of Routes (Fixed Routing)  
 If NRTS = -1, Dynamic Routing, Nonstop Service  
 If NRTS = -2, Dynamic Routing, Some One-Stop Service  
  
 If MHOLD = 0, No Holding  
 If MHOLD > 0, Dynamic Holding
- 06 Number of Vehicles, Capacity, Speed (m/hr), Cycle Time Min (hrs)  
 Cycle Time Max (hrs), DOC \$/hr, DOC \$/cycle  
 Format (2I5, F5.0, 2F5.2, 2F5.0)  
  
 Initial Route (Fixed Routing) or Initial Node (Dynamic Routing)  
 and Starting Time Format (I5, F5.2)
- 07 Number of OD Pairs Format (I5)  
 (If number of OD Pairs < 0 SMUSH Assigns Identification)  
 OD Pair Identification Format (I2I5)
- 08 Number of Route Histograms Format (I5)  
 Route Numbers for Histograms Format (I5)  
 Number of OD Histograms Format (I5)  
 OD Number of Histograms Format (I5)

On the following pages appears an example input deck. This is a setup to run 2 cases. The first case uses the dynamic routing with one stop and holding options to control a fleet of 10 vehicles. The second case is a fixed routing setup with 15 vehicles. Notice that the network and passenger arrival data is the same for both cases, and does not need to be repeated in the second case. Only those parameters which are changed

between cases need to appear in both cases.

Sample Input to SMUSH

```
//G.SYSIN DD *
2
1 1
n.00 8.00
DYNAMIC ROUTING WITH HOLDING. ONE STOP SERVICE
n1
4
0. 5. 14. 18. 24. 19.
5. 0. 14. 15. 28. 23.
14. 14. 0. 10. 34. 20.
18. 15. 10. 0. 41. 37.
24. 28. 34. 41. 0. 5.
19. 23. 30. 37. 5. 0.
n2
3
1
0. 3. 4. 6. 4. 10.
3. 0. 3. 4. 3. 8.
4. 3. 0. 6. 4. 11.
6. 4. 6. 0. 5. 12.
4. 3. 4. 5. 0. 0.
10. 8. 11. 12. 0. 0.
2
0. 0. 0. 0. 29. 71.
0. 0. 0. 0. 22. 54.
0. 0. 0. 0. 31. 77.
0. 0. 0. 0. 34. 82.
0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.
3
0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.
2
?
0.0 1
8.0 1
4
0.5 3
1.0 2
3.0 2
3.5 3
n3
0.0
n4
-1
n5
10.00 10.00
-2 1
n6
10 50 100. .05 .15 300. 100.
1 0.25
2 0.25
3 0.25
4 0.25
5 0.25
6 0.25
```

Number of Cases (I5)  
 Number of MC iterations for each case (I0I5)  
 Begin and end time (2F5.2)  
 Title (15A4)  
 Number of nodes (I5)  
 Distance matrix (10F5.0)  
 (miles)  
 Number of arrival rate tables (I5)  
 ID number for first table (I5)  
 Arrival rate table (pax/hr) (10F5.0)  
 ID number for second table (I5)  
 Arrival rate table  
 Table 3  
 Number of passenger types (I5)  
 Number of data points for type 1 (I5)  
 Time and table ID (F5.2,I5)  
 Number of points for passenger type 2 (I5)  
 Time and table ID (F5.2,I5)  
 Minimum headway F3.1  
 Maximum OD pair loading (I5)  
 Passenger cost parameters a,b (2F10.7)  
 NRTS, MHOLD (2I5)  
 NVEH, CAP, VELC, TCMIN, TCMAX, TCOST, CCOST  
 (2I5,F5.0,2F5.2,2F5.0)  
 Vehicle initial routes and times (I5,F5.2)

1 0.25  
 2 0.25  
 3 0.25  
 4 0.25  
 07  
 08  
 09  
 10  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20

Initial nodes and starting times cont'd

MOBPR (15)

Number of route time histograms (15)  
 Number of OD pair histograms (15)

OD pairs for which histograms are printed (15)

00 End of first case  
 FIXED ROUTING, ALL-STOP WITH CRITICAL LINK QUOTA, MIN HDWY = 0.05 HRS

03  
 0.05  
 04

1						
2	50	50	50	6	9	
3	0	50	50	3	6	
4	50	50	0	50	4	9
5	50	50	50	0	4	11
6	50	50	50	50	0	50
7	50	50	50	50	50	0

05

7	1	1	2	4	3	5	6	1
7	2	2	4	3	5	6	1	2
7	3	4	3	5	6	1	2	4
7	4	3	5	6	1	2	4	3
7	5	5	6	1	2	4	3	5
7	6	6	1	2	4	3	5	6

06

15	50	100.	.05	.15	600.	100.
1	0.25					
2	0.25					
3	0.25					
4	0.25					
5	0.25					
6	0.25					
1	0.30					
1	0.40					
4	0.30					
3	0.40					
5	0.35					
6	0.50					

5 0.60  
4 0.50  
6 0.30

oo



Output

The output of SMUSH consists of the following:

1. Summary of input data
2. Schedule map
3. GASP summary report, for each run
4. Average economic results
5. Histograms of system parameters

The summary of input data and schedule map are self-explanatory. For each simulation run, a GASP summary report is printed. This report, entitled, "Generated Data," lists statistics collected for sample values of the following parameters.

1. Wait, ride and total trip times
  - for the entire system
  - for each route
  - for each OD pair
2. Vehicle load factor
  - for the entire system
  - for each route
3. Average trip time vs. arrival time for each passenger trip
4. Number of each passenger type waiting at the stations during the run

For each parameter, the following statistics are printed:

- mean
- standard deviation
- minimum observed value
- maximum observed value
- total number of observations

The  $n^{\text{th}}$  report gives statistics of all observations made in the first  $n$  Monte Carlo runs. Associated with each variable is a numeric code; Figure B.3 provides the key to this code.

GASP SUMMARY REPORT

## Generated Data

Code	Parameter	
1	Passenger Wait Time	
2	Passenger Ride Time	System
3	Passenger Trip Time	
4	Passenger Wait Time	
5	Passenger Ride Time	Route 1
6	Passenger Trip Time	
7		
8		Route 2
9		
.		
.		
.	The code number for Route n	Wait Time = $3n + 1$
.		Ride Time = $3n + 2$
.		Trip Time = $3n + 3$
91		
92		Route 30
93		
94	Load Factor	System
95	Load Factor	Route 1
.		
124	Load Factor	Route 30
127	Passenger Wait Time	
128	Ride Time	OD Pair 1
129	Trip Time	
	Wait Time	
	Ride Time	OD Pair 2
	Trip Time	
.		
.		
394	Wait Time	
395	Ride Time	OD Pair 90
396	Trip Time	

The code number for OD Pair n Wait Time =  $(n + 41) * 3 + 1$   
 Ride Time =  $(n + 41) * 3 + 2$   
 Trip Time =  $(n + 41) * 3 + 3$

401	Trip Time (hrs) for Type 1 Passengers Who Arrive in the Interval	(0, .5)
402		(.5, 1.0)
.		
.		
420		(9.5, 10.0)
421	Trip Time (hrs) for Type 2 Passengers Who Arrive in the Interval	(0, .5)
.		
.		
440		(9.5, 10.0)
441	System Load Factor during the Interval	(0, .5)
.		
.		
460		(9.5, 10.0)
461	Number of Type 1 Passengers Waiting at Node 1 during the Interval	(0.0, 0.5)
462		(0.5, 1.0)
.		
.		
480		(9.5, 10.0)
481	Number of Type 1 Passengers Waiting at Node 2 during the Interval	
.		
.		
500		
.		
.		
641	Number of Type 1 Passengers Waiting at Node 10 during the Interval	
.		
.		
660		

661      Number of Type 2 Passengers Waiting at Node 1  
  .  
  .  
  .  
680  
  .  
  .  
  .  
841      Number of Type 2 Passengers Waiting at Node 10  
  .  
  .  
  .  
860

LIST OF SUBROUTINESGASP Subroutines

The following is a list of the GASP subroutines used in SMUSH. Complete documentation of them may be found in Reference 7.

- GASP - Controls the simulation start and finish  
- Advances simulation clock
- LOCAT - Locates address of particular file entry
- FILEM - Loads entry into NSET
- REMOVE - Removes entry from NSET
- SET - Updates file pointers in NSET
- COLCT - Collects sample statistics
- TMST - Collects time weighted statistics
- SUMRY - Prints GASP summary report

Non-GASP Subroutines

- DATAx - Modification of DATAN (see Ref. 7)  
- Initializes GASP variables
- EVENTS - Calls subroutine corresponding to the  
pending event
- INTL - Event #1; Initialization  
- Initializes non-GASP variables  
- Prints input data summary  
- Stores initial events (vehicle departures,  
passenger generation)
- PXARV - Event #2; Passenger arrival  
- Transfers passenger data from file 1  
to file 2
- VEHDP - Event #3; Vehicle departure  
- Loads passengers  
- Stores arrival event

- VEHAR - Event #4; Vehicle arrival
  - Unloads passengers
  - Stores departure event
  
- GENPX - Event #5; Passenger Generation
  - Venerates passengers arriving in the interval,  
 $T_{\text{now}} \leq T \leq T_{\text{now}} + .5$  (hours)
  - Stores generation event for  $T_{\text{now}} + .5$  (hours)
  
- GRATE - Interpolates linearly to determine the average passenger arrival rate at any time
  
- HISTO  
 OPUT  
 HPLOT - Prepares and prints output
  
- PSET - Prints contents of NSET (diagnostic)
  
- REX - Generates exponentially distributed random deviate
  
- XRUTE - Determines next route (dynamic routing)
  
- MCOST - Calculates marginal cost for an OD pair
  
- HOLD - Determines whether to hold vehicle or go now.

The following is a description of SMUSH common variables. (The GASP common variables are described in Ref. 7.)

- ARATE(N,I,J,K) - Average arrival rate of passenger type N, going from i to j at time = TRATE (N,K)
  
- MRT(I,J) - Node Number of the  $j^{\text{th}}$  stop on Route i
  
- NSTOP(I) - Number of stops on Route I
  
- NPX(L) - Number of passengers on board vehicle L
  
- NXRT(M) - Successor Route to rate m
  
- D(I,J) - Distance (miles) from I to J
  
- VELC - Velocity of vehicle (mph)
  
- NWAIT(N,J) - Number of passengers waiting of type n at node J

TRATE(N,K)	- Time of the $K^{\text{th}}$ OD arrival rate matrix of passenger type n
TCOST	- DOC \$/HR
CCOST	- DOC \$/CYCLE
NCYCL	- Number of T/O-landing cycles
FTIME	- Total flight time (hrs)
PMILE	- Total passenger miles
NNODE	- Number of nodes in network
NRTS	{ > 0 # routes - 1 flag for dynamic nonstop - 2 flag for dynamic one-stop
TDEL	- Time between successive calls to passenger generator
NOD(I,J)	- Identification number of OD pair i j
NODPR	- Number of OD pairs; if $\leq 0$ , SMUSH determines NODPR and identification numbers itself
CBX	- Slope of passenger waiting cost curve
CAX	- Intercept of passenger waiting cost curve
JHOLD(L)	- Number of times vehicle L holds at present node
MNOLD	- 0 No holding 1 Holding

The following is a source list of SMUSH.



## SMUSH Source List

```

DIMENSION NSET(12000),QSET(4000)
COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INQ,NORPT,NOT,NPPTS,NPUN,NRUNS,NSTAT,OUT,ISEED,TNOW,
2TREG,TFIN,MAX,NPNT,NCRDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
COMMON ATRB(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12)
COMMON/TURC/MPT(20,10),NSTOP(30),NPX(20),ARATF(2,10,10,10),NXRT
1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
2NCYCL,FTIME,PMTLF,NNODE,NRTS,TDFL,NQ(10,10),NODPR,TCMIN,TCMAX
3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPD(10,10)
4,THMIN,NVEH,TLAST(10),TFL(10,10),NWQ(90),EARLY(90),ISFRV(90),CRX
5,CAX,JHOLD(20),MHOLD
DIMENSION VRN(10)
C
C MAIN PROGRAM
C READ NUMBER OF CASES..IPUNS
C READ NUMBER OF M C ITERATIONS..NIRUN(I)
C READ TIME LIMITS.. TREG, TFIN
C
      NCRDR=5
      NPRNT=6
      IRUN=0
      READ(5,1000)IRUNS
1000  FORMAT(15)
      READ(5,1001)(NPN(I),I=1,IRUNS)
1001  FORMAT(10I5)
      READ(5,1010)TREG,TFIN
1010  FORMAT(2F5.2)
      1 IRUN=IRUN+1
      NRUNS=NRN(IRUN)
      NPUN=1
      ISEED=5
      TDFL=.25
      CALL GASP(NSET,QSET)
      CALL OPUT
      IF(IRUN.LT.IRUNS)GO TO 1
      STOP
      END
      SUBROUTINE EVNTS(I,NSET,QSET)
      DIMENSION NSET(1),QSET(1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INQ,NORPT,NOT,NPPTS,NPUN,NRUNS,NSTAT,OUT,ISEED,TNOW,
2TREG,TFIN,MAX,NPNT,NCRDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
COMMON ATRB(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12)
COMMON/TURC/MPT(20,10),NSTOP(30),NPX(20),ARATE(2,10,10,10),NXRT
1(30),D(10,10),VFLC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
2NCYCL,FTIME,PMTLF,NNODE,NRTS,TDFL,NQ(10,10),NODPR,TCMIN,TCMAX
3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPD(10,10)
4,THMIN,NVEH,TLAST(10),TFL(10,10),NWQ(90),EARLY(90),ISFRV(90),CRX
5,CAX,JHOLD(20),MHOLD
C
C CALL SUBROUTINE CORRESPONDING TO ONE OF FIVE EVENTS

```

```

      IF(I.EQ.0) GO TO 9999
      GO TO(1,2,3,4,5),I
      1 CALL INTL(NSFT,QSET)
      GO TO 9999
      2 CALL PXARV(NSFT,QSET)
      GO TO 9999
      3 CALL VEHDP(NSFT,QSET)
      GO TO 9999
      4 CALL VEHR(NSFT,QSET)
      GO TO 9999
      5 CALL GENPX(NSFT,QSET)
9999 RETURN
      END
      SUBROUTINE INTL(NSFT,QSET)
C      ***** INITIALIZE ... EVENT NUMBER 1 *****
      DIMENSION NSET(1),QSET(1)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
      INNO,NORPT,NOT,NPRMS,NPUN,NRUNS,NSTAT,OUT,ISFED,TNOW,
      TREG,TFIN,VXX,NPRNT,NCRDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
      COMMON ATRIR(10),FNO(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
      IMF(4),MLC(4),MLF(4),NCELS(20),NO(4),PARAM(20,4),QTIME(4),SSJMA
      2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTPIR(12)
      COMMON/TURC/MRT(30,10),NSTOP(30),NPX(20),ARATE(2,10,10,10),NXRT
      1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
      2NCYCL,FTIME,PMILE,MNODE,NHTS,TDFL,NOC(10,10),NODPR,TCMTN,TCMAX
      3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPD(10,10)
      4,THMIN,NVE+T,TLAST(10),TFL(10,10),NWOD(90),EARLY(90),ISERV(90),CRX
      5,CAX,JHOLD(20),MHOLD
      DIMENSION XTM(20),JRT(20),XRATE(10,10,10),TTL(15)
C
C      INITIALIZE SOME VARIABLES FOR EACH ITERATION
      DO 90 J=1,10
      TLAST(J)=0.
      DO 90 I=1,2
90  NWAIT(I,J)=0
      DO 91 J=1,20
      JHOLD(J)=0
91  NPX(J)=0
      DO 95 I=1,90
      ISERV(I)=1
      NWOD(I)=0
95  EARLY(I)=TREG
C
C      SKIP TO 16 IF THIS IS NOT A NEW CASE
      IF(NRUN.GT.1)GO TO 16
C
C      INITIALIZE VARIABLES FOR NEW CASE
      NCYCL=0
      FTIME=0.
      PMILE=0.
      DO 7 I=1,20
      MAPX(I)=0
      DO 7 J=1,200
      7 MAP(I,J,1)=-1
C
C      READ DATA
      RFAD(5,36)(TTL(I),I=1,15)
      36 FORMAT(15A4)
      50 RFAD(5,60)TREAD

```

```

60 FORMAT(I2)
   IF(IREAD.EQ.99)GO TO 16
   IF(IREAD.LE.0.OR.IREAD.GE.9)CALL FPROR(IREAD)
   GO TO(51,52,53,54,55,56,57,58),IREAD
51 CONTINUE

```

```

C
C READ NETWORK DATA
  READ(5,1)NNODE
  1 FORMAT(I5)
  DO 2 I=1,NNODE
  2 READ(5,3)(? (I,J),J=1,10)
  3 FORMAT(10F5.0)
  GO TO 50

```

```

C
C READ DEMAND DATA
52 CONTINUE
  READ(5,1)NX
  DO 4 K=1,NX
  READ(5,1)KY
  DO 4 I=1,NNODE
  4 READ(5,5)(XRATE(I,J,KY),J=1,NNODE)
  5 FORMAT(10F5.0)
  READ(5,1)NPTYP
  DO 8 KP=1,NPTYP
  READ(5,1)NTIME(KP)
  NT=NTIME(KP)
  DO 8 K=1,NT
  READ(5,6)TRATE(KP,K),XX
  6 FORMAT(F5.2,I5)
  DO 8 I=1,NNODE
  DO 8 J=1,NNODE
  8 ARATE(KP,I,J,K)=XRATE(I,J,KX)
  GO TO 50

```

```

C
C READ MIN HEADWAY
53 CONTINUE
501 READ(5,40)TMIN
  40 FORMAT(F5.3)
  GO TO 50

```

```

C
C READ MAX LOADING FOR EACH OD PAIR
54 CONTINUE
  READ(5,1)MAXOD
  IF(MAXOD.LE.0)GO TO 29
  DO 9 I=1,NNODE
  9 READ(5,11)(MPOD(I,J),J=1,NNODE)
  GO TO 32
29 DO 31 I=1,NNODE
  DO 31 J=1,NNODE
  31 MPOD(I,J)=100
  32 CONTINUE
  GO TO 50

```

```

C
C READ PAX COST PARAMETERS AND CONTROL OPTIONS
55 CONTINUE
  READ(5,61)CAX,CBX
  61 FORMAT(F5.2,5X,F5.2)
  READ(5,62)VRTS,MHOLD
  62 FORMAT(2I5)
  IF(NRTS.LT.1)GO TO 50

```

```

      DO 10 I=1,NRTS
10  READ(5,11)NSTOP(I),NXRT(I),(MRT(I,J),J=1,10)
11  FORMAT(12I5)
      GO TO 50
C
C   READ VEHICLE PARAMETERS
56  CONTINUE
      READ(5,12)NVEH,ICAP,VFLC,TCMIN,TCMAX,TCOST,CCOST
12  FORMAT(2I5,F5.0,2F5.2,2F5.0)
C
C   READ INITIAL POSITIONING FOR EACH VEHICLE
      DO 13 JD=1,NVEH
      READ(5,14)JPT(JD),XTM(JD)
14  FORMAT(I5,F5.2)
13  CONTINUE
      GO TO 50
C
C   READ OD IDENTIFICATION DATA
57  CONTINUE
      READ(5,1)NODPR
      IF(NODPR.LE.0)GO TO 17
      DO 30 K=1,NODPR
      READ(5,35)N,I,J
35  FORMAT(3I5)
30  NOD(I,J)=N
      GO TO 24
C
C   DEFAULT ID OPTION
17  K=0
      DO 18 I=1,NODPR
      DO 19 J=1,NODPR
      IF(I.EQ.J)GO TO 25
      K=K+1
      NOD(I,J)=K
25  CONTINUE
19  CONTINUE
18  CONTINUE
      NODPR=K
24  CONTINUE
      GO TO 50
C
C   READ HISTOGRAM DATA
58  CONTINUE
      DO 20 I=1,120
20  NDRW(I)=0
      READ(5,1)NHRT
      IF(NHRT.LE.0)GO TO 27
      DO 21 I=1,NHRT
      READ(5,1)J
21  NDRW(J)=I+3
27  READ(5,1)NHOD
      IF(NHOD.LE.0)GO TO 23
      DO 22 I=1,NHOD
      K=NHRT+I+3
      IF(K.GT.NHIST)GO TO 22
      READ(5,1)J
      NDRW(J+30)=K
22  CONTINUE
23  CONTINUE
      GO TO 50

```

```

16 CONTINUE
C
C STORE INITIAL PAX GENERATOR EVENTS
DO 26 I=1,NPTYP
  ATRIB(1)=TRATE(I,1)
  JTRIB(1)=5
  JTRIB(4)=I
26 CALL FILEM(1,NSET,QSET)
C
C STOPE INITIAL DEPARTURE EVENTS
DO 15 JD=1,NVEH
  ATRIB(1)=XTM(JD)
  JTRIB(1)=3
  JTRIB(2)=JD
  JTRIB(3)=JRT(JD)
  JTRIB(4)=1
  IF(NRTS.GT.0)GO TO 28
C
C CALCULATE INITIAL ROUTES FOR DYNAMIC ROUTING OPTION
M=(JD*2)-1
MRT(M,1)=JRT(JD)
MRT(M,2)=JRT(JD)
NXRT(M)=0
NSTOP(M)=2
JTRIB(3)=M
28 CONTINUE
  CALL FILEM(1,NSET,QSET)
15 CONTINUE
C CALCULATE FLIGHT TIME TABLE
DO 92 I=1,NNODE
  DO 92 J=1,NNODE
    TFL(I,J)=D(I,J)/VELC
92 IF(I.EQ.J)TFL(I,J)=0.05
  IF(NRUN.GT.1)RETURN
C
C PRINT INPUT DATA IF THIS IS A NEW CASE
WRITE(6,100)
100 FORMAT(1H1,40X,2PH**INPUT DATA SUMMARY**//)
WRITE(6,104)(TTL(I),I=1,15)
104 FORMAT(//,10X,15A4//)
WRITE(6,105)NNODE
105 FORMAT(10X,'NUMBER OF NODES =',I5/)
WRITE(6,110)
110 FORMAT(10X,'DISTANCE MATRIX')
WRITE(6,115)(J,J=1,NNODE)
115 FORMAT(20X,10I5)
DO 120 I=1,NNODE
120 WRITE(6,125)I,(D(I,J),J=1,NNODE)
125 FORMAT(15X,I5,10F5.0)
  IF(NRTS.LE.0)GO TO 139
  WRITE(6,130)NRTS
130 FORMAT(////,10X,'NUMBER OF ROUTES=',I3/)
  WRITE(6,135)
135 FORMAT(10X,'ROUTE NEXT ROUTE STOPS:')
  DO 138 I=1,NRTS
    JX=NSTOP(I)
138 WRITE(6,140)I,NXRT(I),(MRT(I,J),J=1,JX)
140 FORMAT(10X,I3,6X,I3,10X,10I5)
139 IF(NRTS.LE.0)WRITE(6,141)CBX,CAX
141 FORMAT(////,10X,'ROUTES ARE DETERMINED DURING SIMULATION. R= ')

```

```

1F5.2,' A =',F5.2,')
WRITE(6,145)NVFH
145 FORMAT(///,10X,'NUMBER OF VEHICLES =',I3/)
IF(NRTS.GT.0)WRITE(6,150)
150 FORMAT(12X,'A/C CAPACITY VELOCITY INITIAL ROUTE START TIME
1 CYCLE TIME DOC $/HR DOC $/CYCLE')
IF(NRTS.LE.0)WRITE(6,151)
151 FORMAT(12X,'A/C CAPACITY VELOCITY INITIAL NODE START TIME
1 CYCLE TIME DOC $/HR DOC $/CYCLE')
DO 155 J=1,NVFH
155 WRITE(6,160)J,TCAP,VELC,JRT(J),XTM(J),TCMIN,TCMAX,TCOST,CCOST
160 FORMAT(10X,I3,6X,I4,6X,F4.0,8X,I2,11X,F6.2,6X,F4.2,2X,F4.2,3X,
1F6.2,5X,F6.2)
DO 170 KP=1,NPTYP
WRITE(6,162)KP
162 FORMAT(///,10X,'PASSENGER TYPE',I5)
NT=NTIME(KP)
DO 170 K=1,NT
WRITE(6,165)TRATE(KP,K)
165 FORMAT(//,10X,'AVERAGE ARRIVAL RATE AT TIME =',F6.2/)
WRITE(6,115)(J,J=1,NNODE)
DO 175 I=1,NNODE
175 WRITE(6,125)I,(ARATE(KP,I,J,K),J=1,NNODE)
170 CONTINUE
WRITE(6,180)
180 FORMAT(//,10X,'OD PAIR ID FROM / TO')
WRITE(6,115)(J,J=1,NNODE)
DO 210 I=1,NNODE
210 WRITE(6,215)I,(NOD(I,J),J=1,NNODE)
215 FORMAT(15X,11I5)
RETURN
END
SUBROUTINE PXARV(NSFT,QSET)
C ***** PAX ARRIVAL... EVENT NUMBER 2 *****
DIMENSION NSET(1),QSET(1)
COMMON ID,IM,INIT,JFVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
1NQG,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,ISFED,TNOW,
2TREG,TFIN,XXX,NPRNT,NCRDR,NEP,VNQ(4),IMM,MAXDS,MAXNS
COMMON ATRIB(10),ENQ(4),INN(4),JCELS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),OTIME(4),SSUMA,
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYP,JCLR,JTRIB(12)
COMMON/TURC/MPT(30,10),NSTOP(30),NPX(20),ARATE(2,10,10,10),NXPT
1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
2NCYCL,FTIME,PMILE,NNODE,NRTS,TDFL,NOD(10,10),NODPR,TCMIN,TCMAX
3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPND(10,10)
4,THMIN,NVEI,TLAST(10),TFL(10,10),NWOD(90),EARLY(90),ISERV(90),CRX
5,CAX,JHOLD(20),MHOLD
C
C STORE PAX DATA IN FILE 2
C COLLECT STATISTICS AND UPDATE COUNTERS
C
I=JTRIB(2)
J=JTRIB(3)
KP=JTRIB(4)
JTRIB(1)=KP
CALL FILEM(2,NSET,QSET)
K=NOD(I,J)
NWOD(K)=NWOD(K)+1
IF(NWOD(K).EQ.1)EARLY(K)=TNOW
NWAIT(KP,I)=NWAIT(KP,I)+1

```

```

CALL HISTO(TNOW,0.50,0.50,1)
AT=TNOW*2.
KT1=INT(AT)+20*(I-1)+461
KT2=KT1+200
W1=FLOAT(NWAIT(1,I))
W2=FLOAT(NWAIT(2,I))
CALL COLCT(W1,KT1)
CALL COLCT(W2,KT2)
RETURN
END
SUBROUTINE VEHDP(NSET,QSET)
C ***** VEHICLE DEPARTS EVENT NUMBER 3 *****
DIMENSION NSET(1),QSET(1)
DIMENSION NLOAD(10),NHOLD(20)
COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INQO,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,ISFEN,KNOW,
2TPEG,TFIN,MAX,NPNT,NCRDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
COMMON ATTRIB(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLE(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAMF(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIR(12)
COMMON/TURC/MRT(30,10),NSTOP(30),NPX(20),ARATF(2,10,10,10),NXRT
1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
2NCYCL,FTIME,PMILE,MMODE,NRTS,TOEL,NOC(10,10),NODPR,TCMIN,TCMAX
3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPND(10,10)
4,THMIN,NVE4,TLAST(10),TFL(10,10),NWOJ(90),EARLY(90),ISFRV(90),CRX
5,CAX,JHOLD(20),MHOLD
C
C DETERMINE ORIGIN AND DESTINATION NODES FOR THIS FLIGHT
LOAD=0
DO 1 I=1,10
1 NLOAD(I)=0
LD=JTRIR(2)
IRT=JTRIR(3)
ISTOP=JTRIR(4)
IORG=MRT(IRT,ISTOP)
ISNXT=ISTOP+1
I2=NHOLD(IRT)
C
C DETERMINE IF THIS IS A FAKE FLIGHT (ORIGIN = DESTINATION)
IF(IORG.EQ.MRT(IRT,ISNXT))GO TO 9
C
C IF NOBODY IS GOING TO SHOW UP...GO
RATE=0.
DO 100 KP=1,2
CALL GRATE(KP,IORG,MRT(IRT,ISNXT),TNOW,RPAX)
100 RATE=RATE+RPAX
IF(RATE.LE.0.)GO TO 11
C
C DEFCTDE TO HOLD OR GO
C IF HOLD STORE NEW DEPARTURE FOR TNOW+0.05 AND RETURN
C IF GO CONTINUE
IF(MHOLD.LE.0)GO TO 11
KNEST=MRT(IRT,ISNXT)
CALL HOLD(IORG,KNEST,LD,KHOLD)
IF(KHOLD.EQ.0)GO TO 11
NHOLD(LD)=NHOLD(LD)+1
ATTRIB(1)=ATTRIB(1)+.05
CALL FILEM(1,NSET,QSET)
GO TO 10
11 CONTINUE
C

```

```

C   DETERMINE WHICH OD PAIRS WILL BE LOADED
      JHOLD(LD)=0
      I1=ISNXT
      IF(IRT.EQ.NXRT(IRT)) I1=1
      DO 2 J=I1,I2
      IDEST=MRT(IRT,J)
      INX=NOD(IORG,IDEST)
      2  ISERV(INX)=1
C
C   SEARCH FILE 2 FOR PAX TO LOAD
      KCOL=MFE(2)
      IF(NPX(LD).GE.ICAP) GO TO 3
      7  IF(KCOL.EQ.0) GO TO 3
      IND=LOCAT(2,KCOL,2,NSFT,0SET)
      JORG=NSET(IND)
      IF(JORG.NE.IORG) GO TO 4
      JDEST=NSET(IND+1)
      DO 5 J=I1,I2
      IF(JDEST.EQ.MRT(IRT,J)) GO TO 6
      5  CONTINUE
      GO TO 4
      6  CONTINUE
      IF(NLOAD(JDEST).GE.MPOD(JORG,JDEST)) GO TO 4
      KCOL1=NSET(IND+3)
C
C   REMOVE PAX FROM FILE 2
      CALL PMOVE(KCOL,2,NSFT,0SET)
C
C   UPDATE COUNTERS AND LOAD PAX INTO FILE 3
      ATTRIB(2)=TNOW
      JTRIB(4)=L3
      KP=JTRIB(1)
      CALL FILEM(3,NSFT,0SET)
C
C   UPDATE COUNTERS
      NLOAD(JDEST)=NLOAD(JDEST)+1
      NPX(LD)=NPX(LD)+1
      LOAD=LOAD+1
      INX=NOD(IORG,JDEST)
      NWOD(INX)=NWOD(INX)-1
      EARLY(INX)=ATTRIB(1)
      NWAIT(KP,IORG)=NWAIT(KP,IORG)-1
      IF(NPX(LD).GE.ICAP) GO TO 3
      KCOL=KCOL1
      GO TO 8
      4  KCOL=NSET(IND+3)
      8  IF(KCOL.LT.7777) GO TO 7
      3  CONTINUE
C
C   COLLECT LOAD FACTOR STATISTICS
      A=FLOAT(NPX(LD))
      B=FLOAT(ICAP)
      FLOAD=A/B
      CALL COLCT(FLOAD,94)
      CALL COLCT(FLOAD,94+IRT)
      AT=TNOW/.5
      KT=INT(AT)
      CALL COLCT(FLOAD,KT+441)
      CALL HISTO(FLOAD,0.05,0.05,3)
      9  CONTINUE

```



```

C
C SET UP ARRIVAL EVENT
  JTRIB(1)=4
  JTRIB(2)=LD
  JTRIB(3)=IRT
  JTRIB(4)=ISNXT
C
  IF(IORG.EQ.MRT(IRT,ISNXT))GO TO 15
  FTIME=FTIME+TFL(IORG,MRT(IRT,ISNXT))
  NCYCL=NCYCL+1
15 CONTINUE
  TLOAD=FLOA)*(TCMAX-TCMIN)
  THOLD=TLAST(IORG)+THMIN-(TNOW+TLOAD)
  IF(THOLD.LT.0.)THOLD=0.
  ATRIB(1)=TNOW+TLOAD+THOLD)+TFL(IORG,MRT(IRT,ISNXT))
C
C STOP ARRIVAL EVENT
  CALL FILEM(1,NSET,QSET)
C
C UPDATE SCHEDULE MAP IF THIS IS FIRST ITERATION OF CASE
  IF(IORG.EQ.MRT(IRT,ISNXT))GO TO 10
  TLAST(IORG)=TNOW+TLOAD+THOLD
  IF(NRUN.GT.1)GO TO 10
  TNEX=TNOW+TLOAD+THOLD
  MAPX(LD)=MAPX(LD)+1
  N=MAPX(LD)
  TMAP(LD,N)=TNEX
  MAP(LD,N,1)=1
  MAP(LD,N,2)=IORG
  MAP(LD,N,3)=LOAD
  MAP(LD,N,4)=NPX(LD)
  MAP(LD,N,5)=NHOLD(LD)
  NHOLD(LD)=0
10 CONTINUE
  RETURN
  END
SUBROUTINE VEHAR(NSET,QSET)
C ***** VEHICLE ARRIVES EVENT NUMBER 4 *****
  DIMENSION NSET(1),QSET(1)
  COMMON ID,IM,INIT,JEVNT,JMNT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
  INAQ,NORPT,VOT,NPRMS,NPUN,NRUNS,NSTAT,OUT,ISFFD,TNOW,
  ?TREG,TFIN,MAX,NPRNT,NCRDR,NEP,VNQ(4),IMM,MAXOS,MAXNS
  COMMON ATRIB(10),ENO(4),INN(4),JCELS(20,22),KRANK(4),MAXNO(4),
  IMFE(4),MLC(4),MLE(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
  2(10,5),SUMA(1000,5),NAME(6),NPROJ,MOY,NDAY,NYR,JCLR,JTRIB(12)
  COMMON/TURC/MRT(30,10),NSTOP(30),NPX(20),ARATE(2,10,10,10),NXRT
  1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
  2NCYCL,FTIME,PMTLE,NNONE,VRTS,TDEL,NOD(10,10),NONPR,TCMIN,TCMAX
  3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPOD(10,10)
  4,THMIN,NVEH,TLAST(10),TFL(10,10),NWOD(90),EARLY(90),ISFRV(90),CRX
  5,CAX,JHOLD(20),MHOLD
C
C DECODE BUFFER ARRAY
C LD = AIRCRAFT NUMBER
C IRT = ROUTE NUMBER
C IDEST = NODE NUMBER
  LD=JTRIB(2)
  IRT=JTRIB(3)
  ISTOP=JTRIB(4)
  IDEST=MRT(IRT,ISTOP)

```

```

LOAD=0
IF (NRUN.GT.1) GO TO 10
C
C  UPDATE SCHEDULE MAP
MAPX(LD)=MAPX(LD)+1
N=MAPX(LD)
TMAP(LD,N)=TNOW
MAP(LD,N+1)=0
MAP(LD,N+2)=INDEST
MAP(LD,N+4)=NPX(LD)
10 CONTINUE
KX=IRT*3
C
C  SEARCH FILE 3 AND REMOVE PAX WHO UNLOAD HERE
KCOL=MFF(3)
3 IF (KCOL.EQ.0) GO TO 1
IF (NPX(LD).EQ.0) GO TO 1
IND=LOCAT(?,KCOL,4,NSFT,3SET)
JD=NSET(IND)
IF (JD.NE.L7) GO TO 2
JDEST=NSET(IND-1)
IF (JDEST.NE.INDEST) GO TO 2
KCOL1=NSET(IND+1)
CALL REMOVE(KCOL,3,NSET,0SET)
KD=JTRIP(1)
KY=(KD+41)*3
TWAIT=ATRIP(2)-ATRIP(1)
TRIDE=TNOW-ATRIP(2)
TTRIP=TWAIT+TRIDE
NPX(LD)=NPX(LD)-1
PMILE=PMILE+D(JTRIP(2),JTRIP(3))
LOAD=LOAD+1
AT=2.*ATRIP(1)
JT=INT(AT)
KT=JT+20*(? - 1) + 401
C
C  COLLECT STATISTICS
CALL COLCT(TWAIT,1)
CALL COLCT(TRIDE,2)
CALL COLCT(TTRIP,3)
CALL COLCT(TWAIT,KX+1)
CALL COLCT(TRIDE,KX+2)
CALL COLCT(TTRIP,KX+3)
CALL COLCT(TWAIT,KY+1)
CALL COLCT(TRIDE,KY+2)
CALL COLCT(TTRIP,KY+3)
CALL COLCT(TTRIP,KT)
CALL HISTO(TTRIP,0.20,0.20,2)
IF (NDRW(IRT).GT.0) CALL HISTO(TTRIP,0.20,0.20,NDRW(IRT))
IF (NDRW(KD+30).GT.0) CALL HISTO(TTRIP,0.20,0.20,NDRW(KD+30))
KCOL=KCOL1
GO TO 5
2 KCOL=NSET(IND+1)
5 IF (KCOL.LT.7777) GO TO 3
1 CONTINUE
MAP(LD,N+3)=LOAD
C
C  SETUP DEPARTURE EVENT
ATRIP(1)=TNOW+TCMIN

```

```

JTRIB(1)=3
JTRIB(2)=LJ
JTRIB(3)=IRT
JTRIB(4)=ISTOP
C
C CHECK IF THERE ARE MORE STOPS ON THIS ROUTE
C IF THIS IS LAST STOP GET NEXT ROUTE
  IF(ISTOP.LT.NSTOP(IRT))GO TO 4
  JTRIB(3)=NXRT(IRT)
  IF(NXRT(IRT).LF.0)CALL XRUTE(LD,IDEST)
  JTRIB(4)=1
C IF THE FIRST STOP OF NEXT ROUTE IS NOT THIS NODE CALL ERROR
  IF(MRT(JTRIB(3),1).NF.IDEST)CALL ERROR(17)
C
C STORE DEPARTURE EVENT
4 CALL FILEM(1,NSET,OSET)
  RETURN
  END
  SHAROTIME GENPX(NSET,OSET)
C ***** GENERATE PASSENGERS...EVENT NUMBER 5 *****
  DIMENSION NSET(1),OSET(1)
  DIMENSION J(10),PX(10)
  COMMON ID,IV,INIT,JEVNT,JMNT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
  INNO,NORPT,VOT,NPPMS,NPJN,NRUNS,NSTAT,OUT,ISFEQ,TNOW,
  2TREG,TFIN,4XX,NPPT,NCRD,NEP,VNO(4),IUM,MAXOS,MAXNS
  COMMON ATRIB(10),FNO(4),INN(4),JCFLS(20,22),KRAK(4),MAXNO(4),
  1MFE(4),MLC(4),MLE(4),NCELS(20),NO(4),PARAM(20,4),OTIME(4),SSJMA
  2(10,5),SUMA(1000,5),NAME(6),NPROJ,MOY,NDAY,NYR,JCLR,JTRIB(12)
  COMMON/TURC/MPT(30,10),NSTOP(30),NPX(20),ARATE(2,10,10,10),NXRT
  1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
  2NCYCL,FTIME,PMILF,NNODE,VPTS,TDFL,NOJ(10,10),NONPR,TCMN,TCMAX
  3,THAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPND(10,10)
  4,THMIN,NVEJ,TLAST(10),TFL(10,10),NWD(90),EARLY(90),ISFRV(90),CRX
  5,CAX,JHOLD(20),MHOLD
  KP=JTRIB(4)
  NT=NTIME(KP)
C
C IF TNOW IS PAST LAST DATA POINT RETURN
  IF(TNOW.GE.TRATE(KP,NT))RETURN
C STORE NEXT CALL TO GENPX
  ATRIB(1)=TNOW+TDFL
  CALL FILEM(1,NSET,OSET)
C
C STEP THROUGH OD PAIRS TJ
  DO 100 I=1,NNODE
  DO 101 J=1,NNODE
  IF(I.EQ.J)GO TO 101
C
C SET UP DATA FOR PAX ARRIVAL EVENT
  JTRIB(1)=2
  JTRIB(2)=I
  JTRIB(3)=J
  JTRIB(4)=KP
20 CONTINUE
C
C COUNT DATA POINTS IN THIS INTERVAL
C FOR EACH SEGMENT DETERMINE THE ARRIVAL RATE FOR THE ENDPOINTS A AND B
  T=TNOW
  CALL GRATE(KP,I,J,T,RATE)
  A=RATE

```

```

T=TNOW+TDEL
CALL GRATE(KP,T,J,T,RATE)
R=RATE
DO 1 K=1,10
IF (TRATE(KP,K).GE.TNOW)GO TO 2
1 CONTINUE
2 K1=K
IF (TRATE(KP,K).NF.TNOW)K1=K-1
DO 3 K=1,10
IF (TRATE(KP,K).GE.T)GO TO 4
3 CONTINUE
4 KP=K
L=K2-K1
Q(1)=TNOW
Q(L+1)=TNOW+TDEL
PX(1)=A
RX(L+1)=B
IF (L.EQ.1)GO TO 8
DO 7 M=2,L
RX(M)=ARATE(KP,T,J,K1+M-1)
7 Q(M)=TRATE(KP,K1+M-1)
8 CONTINUE
C DETERMINE THE NUMBER AND PLACEMENT OF ARRIVALS IN EACH SEGMENT
DO 5 K=1,L
DEL=Q(K+1)-Q(K)
A=RX(K)
B=RX(K+1)
APAR=(A+B)/2.
T=Q(K)
6 TADD=PEX(APAR)
IF (T+TADD.GT.Q(K+1))GO TO 10
T=T+TADD
CALL DRAND(ISFFD,R)
IF (A.EQ.6)GO TO 12
S=SQRT((1-R)*(A**2)+(R*(B**2)))
TW=(DEL)/(R-A)*(S-A)
GO TO 13
12 TW=DEL*R
13 ATRIB(1)=Q(K)+TW
C
C STOP ARRIVAL EVENT
IF (TW.LT.99999.)CALL FILEM(1,NSFT,QSET)
GO TO 6
5 CONTINUE
10 CONTINUE
101 CONTINUE
100 CONTINUE
RETURN
END
SUBROUTINE GRATE(KP,T,J,T,RATE)
DIMENSION VSET(1), SFT(1)
COMMON ID,IM,INIT,JFVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INQ,NORPT,NOT,NPOMS,NPUN,NRUNS,NSTAT,OUT,ISFFD,TNOW,
2TEFG,TFIN,MAX,NPANT,NORDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
COMMON ATRIB(10),FNO(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
IMFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,MYR,JCLR,JTRIP(12)
COMMON/TURC/MPT(30,10),NSTOP(30),NPX(20),ARATE(2,10,10,10),NXPT
1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST.

```

```

2NCYCL,FTIME,PMILF,NNODE,VRTS,TOFL,NOD(10,10),NODPR,TCMIN,TCMAX
3,MAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPOD(10,10)
4,THMIN,NVE+,TLAST(10),TFL(10,10),NWD(90),EARLY(90),ISFRV(90),CRX
5,CAX,JHOLD(20),MHOLD

```

```

C
C GET RATE...LINEARLY INTERPOLATE
C KP = PAX TYPE
C I = ORIG
C J = DEST
C T = TIME
C R = RATE
C

```

```

RATE=0.
NT=NTIME(KP)
DO 1 K=1,NT
IF (TRATE(KP,K).EQ.T)GO TO 2
IF (TRATE(KP,K).GT.T)GO TO 3
1 CONTINUE
RETURN
2 RATE=ARATE(KP,I,J,K)
RETURN
3 IF (K.GT.1)GO TO 4
RETURN
4 RATE=ARATE(KP,I,J,K-1)+(T-TRATE(KP,K-1))*(ARATE(KP,I,J,K)
1-ARATE(KP,I,J,K-1))/(TRATE(KP,K)-TRATE(KP,K-1))
RETURN
END
SUBROUTINE XRITE(LX,IX)
COMMON ID,IV,INIT,JEVNT,JMNIT,MFA,MSTUP,MX,MXC,NCLCT,NHIST,
1NNO,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,ISFEN,TNOW,
2TRFG,TFIN,XXX,NPRNT,NCRD2,NEP,VNO(4),IMM,MAXOS,MAXNS
COMMON ATTR(10),END(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLF(4),NCELS(20),NO(4),PARAM(20,4),OTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,NOV,NDAY,NYR,JCLR,JTRIB(12)
COMMON/TURC/MRT(30,10),NSTOP(30),NPX(20),ARATE(2,10,10,10),NXPT
1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
2NCYCL,FTIME,PMILF,NNODE,VRTS,TOFL,NOD(10,10),NODPR,TCMIN,TCMAX
3,MAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPOD(10,10)
4,THMIN,NVE+,TLAST(10),TFL(10,10),NWD(90),EARLY(90),ISFRV(90),CRX
5,CAX,JHOLD(20),MHOLD

```

```

C
C DETERMINE THE NEXT ROUTE FOR DYNAMIC ROUTING OPTION
C LX = AIRCRAFT NUMBER
C IX = PRESENT POSITION
C I=IX
C L=LX
C CM=0.

```

```

C
C DETERMINE J AND K SUCH THAT THE MARGINAL COST/SERVICE TIME
C FOR IJ AND JK IS MAXIMIZED
DO 1 JX=1,NNODE
IF (I.EQ.JX)GO TO 1
S1=TNOW+TCMIN
CALL MCOST(S1,I,JX,C1)
S2=S1+TFL(I,JX)+TCMIN
DO 2 KX=1,NNODE
IF (JX.EQ.KX)GO TO 2
CALL MCOST(S2,JX,KX,C2)
CMX=(C1+C2)/(S2+TFL(JX,KX)-S1)
IF (CMX.LE.CM)GO TO 2

```

```

      CM=CMX
      J=JX
      K=KX
2 CONTINUE
1 CONTINUE
  M=(2*L)-1
  IF (CM.GT.0.) GO TO 6
  NSTOP(M)=2
  MDT(M,1)=I
  MDT(M,2)=I
  NXRT(M)=0
  JTRIB(3)=M
  GO TO 4
6 CONTINUE
C
C  UPDATE SERVICE PENDING FLAGS
  ID1=NOD(I,J)
  ID2=NOD(J,<)
  ID3=NOD(I,<)
  ISERV(ID1)=0
  ISFPV(ID2)=0
  IF (NRTS+2) 3,3,5
5 CONTINUE
C
C  NONSTOP SERVICE ONLY..SET JP 2 NONSTOP ROUTES
  N=M+1
  MRT(M,1)=I
  MDT(M,2)=J
  MDT(N,1)=J
  MDT(N,2)=K
  NSTOP(M)=2
  NSTOP(N)=2
  NXRT(M)=N
  NXRT(N)=0
  JTRIB(3)=M
  GO TO 4
3 CONTINUE
C
C  ONE STOP SERVICE..SET UP ONE 2 STOP ROUTE
  NSTOP(M)=3
  MDT(M,1)=I
  MDT(M,2)=J
  MDT(M,3)=K
  NXRT(M)=0
  JTRIB(3)=M
  MPOD(I,J)=TCAP
  MPOD(J,K)=TCAP
  IF (I.FQ.K) GO TO 4
  ISERV(ID3)=0
C
C  PUT LIMITS ON NUMBER OF IK PAX TO BOARD
  MPOD(I,K)=TCAP-NWOD(ID1)
  IF (NWOD(ID1).GT.NWOD(ID2)) GO TO 4
  MPOD(I,K)=TCAP-NWOD(ID2)
4 CONTINUE
  JHOLD(L)=K
  RETURN
  END
  SUBROUTINE MCOST(S,I,J,C)
  COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,

```

```

INNO,NORPT,VOT,NPRMS,NPUN,NRUNS,NSTAT,OUT,ISFEQ,TNOW,
2TRFG,TFIN,MAX,NPRNT,NCRDR,NEP,VNO(4),IMM,MAXQS,MAXNS
COMMON ATTRB(10),ENO(4),INN(4),JCELS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLE(4),NCELS(20),NO(4),PARAM(20,4),QTIME(4),SSJMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MOV,NDAY,NYR,JCLR,JTRIP(12)
COMMON/TURC/MRT(30,10),NSTOP(30),NPX(20),ARATF(2,10,10,10),NXRT
1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
2NCYCL,FTIME,PMILE,NNODE,NRTS,TOEL,NOD(10,10),NODPR,TCMTN,TCMAX
3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPOD(10,10)
4,THMIN,NVEH,TLAST(10),TFL(10,10),NWD(90),EARLY(90),ISFRV(90),CRX
5,CAX,JHOLD(20),MHOLD

```

```

C
C DETERMINE MARGINAL COST AT TIME S FOR OD PAIR IJ

```

```

C=0.
K=NOD(I,J)
IF(ISFRV(K).EQ.0)RETURN
R=0.
DO 1 KP=1,2
CALL GRATE(KP,I,J,S,RATE)
1 R=R+RATE
W1=FLOAT(NWOD(K))
W=W1+R*(S-TNOW)
C=W*(CAX+(CRX/2.)*(S-EARLY(K)))
RETURN
END

```

```

SUBROUTINE HOLD(I,J,L,KHOLD)
COMMON ID,IM,INIT,JFVMT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INNO,NORPT,VOT,NPRMS,NPUN,NRUNS,NSTAT,OUT,ISFEQ,TNOW,
2TRFG,TFIN,MAX,NPRNT,NCRDR,NEP,VNO(4),IMM,MAXQS,MAXNS
COMMON ATTRB(10),ENO(4),INN(4),JCELS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLE(4),NCELS(20),NO(4),PARAM(20,4),QTIME(4),SSJMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MOV,NDAY,NYR,JCLR,JTRIP(12)
COMMON/TURC/MRT(30,10),NSTOP(30),NPX(20),ARATF(2,10,10,10),NXRT
1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
2NCYCL,FTIME,PMILE,NNODE,NRTS,TOEL,NOD(10,10),NODPR,TCMTN,TCMAX
3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPOD(10,10)
4,THMIN,NVEH,TLAST(10),TFL(10,10),NWD(90),EARLY(90),ISFRV(90),CRX
5,CAX,JHOLD(20),MHOLD

```

```

C
C DETERMINE WHETHER TO HOLD OR GO ON FLIGHT FROM I TO J

```

```

C KHOLD = 1... HOLD
C KHOLD = 0... GO
C K=JHOLD(L)
IFLAG=K
KHOLD=0
T1=0.
T2=0.
TF=0.
TF1=0.
M=I
N=J
S=TFL(M,N)+TCMTN
1 MN=NOD(M,N)
NW=NWOD(MN)
IF(NRTS+2)30,30,35
30 IF(K.EQ.0)NW=NW+NPX(L)
IF(K.EQ.0,TR,IFLAG,NE,K,OR,I,FO,K)GO TO 35
IK=NOD(I,K)
NW=NW+NWOD(IK)
35 CONTINUE

```

```

      IF (NW.EQ.0) GO TO 3
      WN=FLOAT(NW)
      IF (NW.GT.0) GO TO 35
      W=TNOW-EARLY(MN)
      GO TO 37
36  IF (K.EQ.0) GO TO 20
      IF (K.GT.0) W=TNOW-EARLY(IK)
37  CONTINUE
      NJK=0
      TJK=0.
      IF (IFLAG.EQ.K) GO TO 2
      PIK=S/W*WN
      NJK=INT(PJK)
      TJK=S
      2  IF (NW+NJK-ICAP) 6,20,20
      5  C=CCOST+TCOST*TFL(M,N)/2.
      IF (CBX.LE.0.) GO TO 40
      F=(CAX**2)/4.+(4.*C*CBX)/(3.*WN)
      F=SQRT(F)
      WSTAR=(F-(CAX/2.))/(2.*CBX/3.)
      40  IF (CBX.EQ.0.) WSTAR=(2.*C)/(CAX*WN)
C
C   THIS IS A USEFUL DIAGNOSTIC
C   IF (TNOW.GT.3. AND TNOW.LT.5.) WRITE(6,2000) L,WN,W,WSTAR
C   IF (TNOW.LE.2.) WRITE(6,2000) L,WN,W,WSTAR
2000  FORMAT(5X,I5,3F8.4)
C
      IF (W+TJK-WSTAR) 3,7,7
      7  IF (IFLAG-K) 9,20,9
      3  IF (NW.EQ.0) WSTAR=10.
      IF (IFLAG.EQ.0) GO TO 4
      T1=EARLY(MN)+WSTAR
      IFLAG=0
      N1=NW
      WN1=WN
      M=J
      N=K
      GO TO 1
      4  IF (K.EQ.0) GO TO 10
      IF (N1.EQ.0. AND NW.EQ.0) GO TO 10
      9  T2=EARLY(MN)+WSTAR
      IF (T1+S-T2) 10,5,5
      5  TE=T1+S-T2
      TE1=TE*(WN+(WN*S/W))/(WN1+WN+(WN*S/W))
      IF (T1-TE1-TNOW) 20,20,10
      10  K HOLD=1
      20  CONTINUE
C
C   THIS IS A USEFUL DIAGNOSTIC
C   IF (TNOW.LT.3. OR TNOW.GT.5.) RETURN
C   IF (TNOW.GT.2.) RETURN
C   WRITE(6,1000) N1,NW,I,J,K,TNOW,W,T1,T2,TE,TE1,TJK,NJK,KHOLD
1000  FORMAT(1X,I5,7F8.2,2I5/)
      RETURN
      END
      SUBROUTINE OPHIT
      DIMENSION NSET(1),OSFT(1)
      DIMENSION TTL1(15),TTL2(5),TX(20),PX(20),KCFLS(22,20)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
      INQG,NORPT,NOT,NPRMS,NPUN,NRUNS,NSTAT,OUT,ISFEQ,TNOW,

```



```

2TREG,TFIN,MAX,NPRNT,NCRDR,NEP,VNQ(4),IMM,MAXOS,MAXNS
COMMON ATRJ(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLE(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSJMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRJ(12)
COMMON/TURC/MPT(30,10),NSTOP(30),MPX(20),APATF(2,10,10,10),NXPT
1(30),D(10,10),VELC,NWAIT(2,10),TRATE(2,10),NTIME(2),TCOST,CCOST,
2NCYCL,FTIME,PMILE,NNONE,VRTS,TOFL,NOC(10,10),NONPR,TCMIN,TCMAX
3,TMAP(20,200),MAP(20,200,5),MAPX(20),NDRW(120),TCAP,MPOD(10,10)
4,THMIN,NVE,TLAST(10),TFL(10,10),NWO(90),EARLY(90),ISFRV(90),CRX
5,CAX,JHOLD(20),MHOLD
DATA TX/' ' SY',STEM',TRIP',TI',MES',ACTU',AL',
3'APPI',
1'VALS',LTA',DFA',CTOR',R',ROUTE',TIME',HRS',
2'OD',PAIR',TO'
DATA PX/' ' 1', 2', 3', 4', 5', 6', 7'
1', 8', 9', 10', 11', 12', 13', 14', 15'
2', 16', 17', 18', 19', 20'
XP=FLOAT(NRJNS)
NCYCL=NCYCL/NRJNS
PMILE=PMILE/XP
FTIME=FTIME/XP
TC=(FLOAT(NCYCL))*CCOST+(FTIME*TCOST)
PM=TC/PMILE
WRITE(6,1)
1 FORMAT(1H1,40X,22H**STIMULATION RESULTS**//)
DO 2 I=1,22
C
C TRANSPOSE HISTOGRAM DATA
DO 2 J=1,20
2 KCFLS(I,J)=JCFLS(J,I)/NRJNS
K=NCELS(1)
K2=K+2
DO 3 J=1,K2
3 WRITE(6,4)J,(JCFLS(I,J),I=1,20)
4 FORMAT(1X,21I5)
WRITE(6,10)NCYCL
10 FORMAT(////,30X,'AVERAGE NUMBER OF CYCLES = ',TR)
WRITE(6,11)FTIME
11 FORMAT(30X,'AVERAGE FLIGHT TIME = ',FR,2)
WRITE(6,12)PMILE
12 FORMAT(30X,'AVERAGE PASSENGER MILES = ',FR,0)
WRITE(6,13)TC
13 FORMAT(30X,'TOTAL DIRECT OPERATING COST = ',FR,0)
WRITE(6,14)PM
14 FORMAT(30X,'DIRECT COST PER PAX MILE = ',FR,4)
DO 15 KP=1,2
WRITE(6,900)
900 FORMAT(////)
DO 16 NN=1,6
NS=20*(NN-1)+200*(KP-1)+461
NF=NS+15
16 WRITE(6,910)(SIJMA(N,5),N=NS,NF)
910 FORMAT(1X,16F6,0)
15 CONTINUE
WRITE(6,1000)
1000 FORMAT(1H1,///,40X,16H**SCHEDULE MAP**)
DO 100 LD=1,NVFN
IF(MAP(LD,1,1),LT,0)GO TO 112
WRITE(6,1010)LD
1010 FORMAT(////,40X,'VEHICLE',I4,/)

```

```

      DO 110 I=1,200
      IF (MAP(LD,I,1)) 112,102,104
102 WRITE(6,1020)MAP(LD,I,2),TMAP(LD,I),MAP(LD,I,4),MAP(LD,I,3)
1020 FORMAT(5X,'ARRIVE NONE',I4,' AT TIME',F5.2,' WITH',I4,
1' ON BOARD UNLOAD',I4)
      GO TO 110
104 WRITE(6,1025)MAP(LD,I,2),TMAP(LD,I),MAP(LD,I,4),MAP(LD,I,3)
1,MAP(LD,I,5)
1025 FORMAT(5X,'LEAVE NONE',I4,' AT TIME',F5.2,' WITH',I4,
1' ON BOARD LOAD',I4,' AFTER HOLDING ',I4,' TIMES')
110 CONTINUE
112 CONTINUE
100 CONTINUE
C
C   SET UP ACTUAL PAX ARRIVAL HISTOGRAM
      DO 19 JW=1,15
19 TTL1(JW)=TX(1)
      DO 20 JW=2,5
20 TTL1(JW)=TX(JW+5)
      DO 22 JW=1,5
22 TTL2(JW)=TX(1)
      TTL2(2)=TX(16)
      TTL2(3)=TX(17)
      DO 23 K4=1,22
23 KCELS(K4,1)=KCELS(K4,1)*2
      CALL HPL0T(NCFLS(1)+2,KCELS(1,1),6.0,0.50,TTL1,TTL2)
C
C   SYSTEM TRIP TIME HISTOGRAM
      DO 25 JW=1,15
25 TTL1(JW)=TX(1)
      TTL1(3)=TX(2)
      TTL1(4)=TX(3)
      DO 27 JW=1,3
27 TTL1(JW+5)=TX(JW+3)
      CALL HPL0T(NCFLS(2)+2,KCELS(1,2),0.0,0.20,TTL1,TTL2)
C
C   ROUTE TIME HISTOGRAM
      TTL1(2)=TX(14)
      TTL1(3)=TX(15)
      IF (NRTS.LE.0) GO TO 32
      DO 30 JW=1,NRTS
      IF (NDRW(JW).EQ.0) GO TO 31
      TTL1(4)=PX(JW)
      K=NDRW(JW)
      CALL HPL0T(NCFLS(K)+2,KCELS(1,K),0.0,0.20,TTL1,TTL2)
31 CONTINUE
30 CONTINUE
32 CONTINUE
C
C   OD PAIR HISTOGRAM
      TTL1(2)=TX(18)
      TTL1(3)=TX(19)
      DO 50 K=1,NODPP
      M=NDRW(K+30)
      IF (M.EQ.0) GO TO 51
      DO 70 I=1,NVONF
      DO 70 J=1,NVONF
      IF (MOD(I+J).EQ.K) GO TO 80
70 CONTINUE
80 CONTINUE

```

```

      TTL1(4)=TX(1)
      IF(K.LF.20)TTL1(4)=PX(K)
      TTL1(9)=PX(I)
      TTL1(10)=TX(20)
      TTL1(11)=PX(J)
      CALL HPL0T(NCELS(M)+2,KCELS(1,4),0.0,0.20,TTL1,TTL2)
51 CONTINUE
50 CONTINUE
C
C   LOAD FACTOR HISTOGRAM
DO 28 JW=1,15
28  TTL1(JW)=TX(1)
      TTL1(3)=TX(2)
      TTL1(4)=TX(3)
      TTL1(5)=TX(11)
      TTL1(6)=TX(12)
      TTL1(7)=TX(13)
DO 29 JW=1,5
29  TTL2(JW)=TX(1)
      CALL HPL0T(NCELS(3)+2,KCELS(1,3),0.0,0.05,TTL1,TTL2)
      RETURN
      END
      SUBROUTINE HISTO(X1,A,W,N)
      DIMENSION NSET(1),QSFT(1)
      COMMON ID,IM,INIT,JFVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
      INNO,NORPT,NOT,NPMS,NPUN,NRIUS,NSTAT,OUT,ISFFD,TNOW,
      PREG,TFIN,MAX,NPRT,NCRDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
      COMMON ATTR(10),ENO(4),INN(4),JCELS(20,22),KRANK(4),MAXNO(4),
      MF(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
      2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,VDAY,NYR,JCLR,JTRIB(12)
      IF(N-NHIST)11,11,2
      2 WRITE(NPRT,250)4
250 FORMAT(1X,'ERROR IN HISTOGRAM',I4//)
      CALL ERROR(62)
11 IF(N)2,2,3
      3 X=X1-A
      IF(X)6,7,7
      6 IC=1
      GO TO 8
      7 IC=X/W+2.
      IF(IC-NCELS(N)-1)8,8,9
      9 IC=NCELS(N)+2
      8 JCELS(N,IC)=JCELS(N,IC)+1
      RETURN
      END
      SUBROUTINE HPL0T(NPTS,ITAB,XA,XW,TITLE1,TITLE2)
C   GENERALIZED HISTOGRAM PLOT
      DIMENSION LINE(50),XLINE(50),ITAB(50),TITLE1(15),TITLE2(5)
      DIMENSION FTAB(30)
      DATA KBLAN(1),KAST(1),KX(1),LEND(50),M(6)/
      V1 = 0.
      DO 20 J=1,NPTS
C   TEMPORARY
      FTAB(J) = ITAB(J)
20  V1 = AMAX1(V1,FTAB(J))
      WRITE (M,10) TITLE1
10  FORMAT(1H1,30X,15A4,/)
      JSCAL = MAX1(1.,V1/50.)
      IF (V1.GT.=FLOAT(JSCAL*50)) JSCAL = JSCAL+1
      ICSAL=2

```

```

I1 = V1
KMAX = MIN0(I1, JSCAL*50)
LINF(1) = <AST
I0 = 51*JSCAL
INC = 10*JSCAL
DO 200 L=1,50
I0 = I0-JSCAL
LAREL = KRLANK
DO 100 J=2,LEND
100 LINF(J) = <BLANK
IF (I0.GT.<MAX) GO TO 140
DO 130 J=1,NPTS
K = J*JSCAL+2
IF (I0.LE.FTAR(1)) LINF(K) = KY
130 CONTINUE
140 CONTINUE
IF (MOD(I0,INC).NE.0) GO TO 120
LINF(2) = <AST
WRITE(6,210)I0,LINF
210 FORMAT(24X,I4,2X,50A1)
GO TO 215
120 WRITE(M,212)LINF
212 FORMAT(30X,50A1)
215 CONTINUE
200 CONTINUE
I0=0
DO 220 J=1,LEND
220 LINF(J) = <AST
WRITE(M,210)I0,LINF
XLINF(1) = XA
A1=4.
N1=6
DO 230 J=2,N1
230 XLINF(J) = XLINF(J-1)+A1*XW
WRITE (M,240) (XLINF(J),J=1,N1)
240 FORMAT(26X,I2(4X,F4.1))
WRITE (M,250) TITLE?
250 FORMAT (/40X,5A4)
RETURN
END
SUBROUTINE GASP(NSFT,0SET)
COMMON ID,IM,INIT,JFVNT,JMNIT,MFA,MSTUP,MX,MXC,NCLCT,NHIST,
IND0,NORPT,NOT,NPPTS,NPUN,NRUNS,NSTAT,OUT,ISFED,TNOW,
2TBEG,TFIN,MXX,NPNT,NCPDR,NEP,VNQ(4),IMM,MAXOS,MAXNS
COMMON ATRIR(10),ENO(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLE(4),NCELS(20),NQ(4),PARAM(20,4),OTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MOV,NDAY,NYR,JCLR,JTRIR(12)
DIMENSION NSFT(1),0SET(1)
1 CALL DATAX(NSFT,0SET)
JFVNT = 10)
C CALL MONTR(NSFT,0SET)
WRITE (NPRNT,403)
403 FORMAT (1H),3RX,24H**INTERMEDIATE RESULTS**//)
10 KCOL = MFE(1)
CALL RM0VE(KCOL,1,NSFT,0SET)
TNOW =AMAX1(TNOW,ATRIR(1))
JFVNT = JTRIR(1)
IF (JFVNT-100) 13,12,6
13 I = JFVNT
CALL EVNTS(I,NSFT,0SET)

```

```

      IF (MSTOP) 40,8,20
40  MSTOP = 0
      IF (NORPT) 14,22,42
20  IF (TNOW-TFIN) 8,22,22
22  CALL SUMRY
42  IF (NRUNS-NRUN) 14,9,23
23  NRUN=NRUN+1
      GO TO 1
14  CALL ERROR(93)
      CALL MONTR
      GO TO 10
12  IF (JMNIT) 14,30,31
30  JMNIT = 1
      GO TO 10
31  JMNIT = 0
      GO TO 10
      IF (JMNIT) 14,10,32
32  JTRIB(1) = JEVNT
      JEVNT = 100
      CALL MONTR
      GO TO 10
0  RETURN
END
SUBROUTINE DATX(NSFT,OSFT)
  DIMENSION NSET(1),OSET(1)
  COMMON ID,IV,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
  INQ,NORPT,NOT,NPOMS,NPUN,NRJNS,NSTAT,UIT,ISFD,TNOW,
  TREG,TFIN,XX,NPNT,NCRDR,NEP,VNO(4),IMM,MAXOS,MAXNS
  COMMON ATRB(10),FNO(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
  IMFF(4),MLC(4),MLE(4),NCFLS(20),NQ(4),PARAM(20,4),OTIME(4),SSUMA
  2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12)
  NHIST=20
  NCLCT=1000
  NSTAT=10
  IN=2000
  IV=4
  IMM=2
  NQ=3
  MXC=0
  MSTOP=1
  JCLR=0
  DO 1 I=1,20
1  NCFLS(I)=20
  DO 2 I=1,4
  KRANK(I)=1
2  INN(I)=1
  DO 3 I=1,10
  ATRB(I)=0.
3  JTRIB(I)=0
  TNOW = TREG
  DO 142 J=1,NQ
142 OTIME(J) = TNOW
26  JMNIT = 0
44  INIT = 1
      CALL SET(1,NSFT,OSFT)
      JTRIB(1)=1
      ATRB(1)=TNOW
      CALL FILEM(1,NSFT,OSFT)
      IF (NRUN.GT.1.AND.JCLR.EQ.0) GO TO 60
10  IF (NSTAT) 117,110,117

```

```

117 DO 360 I=1,NSTAT
    SCUMA(I,1) = TNOW
    DO 370 J=2,3
370 SCUMA(I,J) = 0.
    SCUMA(I,4) = 1.E20
360 SCUMA(I,5) = -1.F20
110 IF (NCLCT) 116,60,116
116 DO 18 I=1,NCLCT
    DO 17 J=1,3
    17 SIJMA(I,J) = 0.
    SIJMA(I,4) = 1.F20
    18 SIJMA(I,5) = -1.E20
    DO 20 I=1,20
    DO 20 J=1,22
    20 JCFLS(I,J)=0
60 RETURN
    END
    SUBROUTINE FILFM(JQ,NSET,QSET)
    DIMENSION NSET(1),QSET(1)
    COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
    INQ,NORPT,VOT,NPRMS,NPUN,NRUNS,NSTAT,OUT,ISFED,TNOW,
    2TREG,TFIN,MXX,NPRNT,NCRD2,NEP,VNQ(4),IMM,MAXOS,MAXNS
    COMMON ATTRB(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
    1MFE(4),MLC(4),MLE(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSJMA
    2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12)
    IF (MFA-ID) 2,2,3
    3 WRITE (NPRNT,4)
    4 FORMAT (//24H OVERLAP SET GIVEN BELOW/)
    CALL ERROR(87)
    2 INDX = (MFA-1)*IMM
    DO 1 I=1,IMM
    INDX = INDX+1
    1 QSET(INDX) = ATTRB(I)
    INDX = (MFA-1)*MXC
    DO 10 I=1,IM
    INDX = INDX+1
    10 NSET(INDX) = JTRIB(I)
    CALL SET(JQ,NSET,QSET)
    RETURN
    END
    SUBROUTINE RMV(KCOL,JQ,NSET,QSET)
    DIMENSION NSET(1),QSET(1)
    COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
    INQ,NORPT,VOT,NPRMS,NPUN,NRUNS,NSTAT,OUT,ISFED,TNOW,
    2TREG,TFIN,MXX,NPRNT,NCRD2,NEP,VNQ(4),IMM,MAXOS,MAXNS
    COMMON ATTRB(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
    1MFE(4),MLC(4),MLE(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSJMA
    2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12)
    IF (KCOL) 16,16,2
    16 CALL ERROR(97)
    2 MLC(JQ) = <COL
    INDX = (KCOL-1)*IMM
    DO 3 I=1,IMM
    INDX = INDX+1
    3 ATTRIB(I) = QSET(INDX)
    INDX = (KCOL-1)*MXC
    DO 10 I=1,IM
    INDX = INDX+1
    10 JTRIB(I) = NSET(INDX)
    OUT = 1.

```

```

CALL SET(JQ,NSFT,QSET)
RETURN
END
SUBROUTINE SET(JQ,NSFT,QSET)
DIMENSION NSET(1),QSET(1)
COMMON ID,IM,INIT,JFVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INNO,NORPT,NOT,NPRMS,NPIUN,NRUNS,NSTAT,OUT,ISFED,TNOW,
?TREG,TFIN,MAX,NPRT,NCRD3,NEP,VNQ(4),IMM,MAXQS,MAXNS
COMMON ATTRB(10),ENQ(4),INN(4),JCELS(20,22),KRANK(4),MAXNO(4),
MFE(4),MLC(4),MLE(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSJMA
2(10,5),SUMA(1000,5),NAYF(6),NPROJ,MON,NDAY,NYP,ICLR,JTRIR(12)
IF (INIT-1) 27,28,27
28 KOL = 7777
KOF = 8888
KLF = 9999
Mx = IM+1
MXX = IM+2
MAXQS = ID*IMM
MAXNS = ID*MXX
DO 2 J=1,MAXQS
2 QSET(J) = 0.
DO 4 J=1,MAXNS
4 NSFT(J) = 0
DO 1 I=1,17
INDX = I*MXX
NSET(INDX-1) = I+1
1 NSET(INDX) = I-1
NSET(MAXNS-1) = KOF
DO 3 K=1,NQ3
NQ(K) = 0
MLC(K) = 0
MFE(K) = 0
MAXNQ(K) = 0
MLE(K) = 0
ENQ(K) = 0.
VNQ(K) = 0.
3 QTIME(K) = TNOW
MFA = 1
INIT = 0
OUT = 0.
GO TO 99
27 MFEX = MFE(JQ)
KNT = 2
KS = KRANK(JQ)
KSJ = 1
IF (KS-100) 1020,100,1000
1000 KSJ = 2
KS = KS-100
1020 IF (OUT) 100,8,5
8 INDX = MFA*MXX-1
NYFA = NSET(INDX)
IF (INN(JQ)-1) 100,7,6
7 MLEX = MLE(JQ)
IF (MLEX) 100,10,11
10 INDX = MFA*MXX
NSET(INDX) = KLF
MFF(JQ) = MFA
17 INDX = MFA*MXX-1
NSET(INDX) = KOL
MLE(JQ) = MFA

```

```

14 MFA = NXFA
   IF (MFA-KOF) 237,238,238
237 INDX = NXFA*MXX
   NSET(INDX) = KLE
238 XNQ = NQ(JQ)
   ENQ(JQ) = ENQ(JQ)+XNQ*(TNOW-QTIME(JQ))
   VNQ(JQ) = VNQ(JQ)+XNQ*XNQ*(TNOW-QTIME(JQ))
   QTIME(JQ) = TNOW
   NQ(JQ) = NQ(JQ)+1
   MAXNQ(JQ) = MAX0(MAXNQ(JQ),NQ(JQ))
   MLC(JQ) = MFE(JQ)
   GO TO 99
11 GO TO (1100,1120), KSJ
1100 INDX1 = (MFA-1)*IMM+KS
     INDX2 = (MLEX-1)*IMM+KS
     IF (QSET(INDX1)-QSET(INDX2)) 12,13,13
1120 INDX1 = (MFA-1)*MXX+KS
     INDX2 = (MLEX-1)*MXX+KS
     IF (NSET(INDX1)-NSET(INDX2)) 12,13,13
13 INDX = MLEX*MXX-1
   MSU = NSET(INDX)
   NSET(INDX) = MFA
   INDX = MFA*MXX
   NSET(INDX) = MLEX
   GO TO (18,17), KNT
18 INDX = MFA*MXX-1
   NSET(INDX) = MSU
   INDX = MSU*MXX
   NSET(INDX) = MFA
   GO TO 14
12 KNT = 1
   INDX = MLEX*MXX
   MLEX = NSET(INDX)
   IF (MLEX-KLE) 11,16,11
16 INDX = MFA*MXX
   NSET(INDX) = KLE
   MFE(JQ) = MFA
26 INDX = MFA*MXX-1
   NSET(INDX) = MFEX
   INDX = MFEX*MXX
   NSET(INDX) = MFA
   GO TO 14
6 IF (MFEX) 100,10,19
19 GO TO (1200,1220), KSJ
1200 INDX1 = (MFA-1)*IMM+KS
     INDX2 = (MFEX-1)*IMM+KS
     IF (QSET(INDX1)-QSET(INDX2)) 20,21,21
1220 INDX1 = (MFA-1)*MXX+KS
     INDX2 = (MFEX-1)*MXX+KS
     IF (NSET(INDX1)-NSET(INDX2)) 20,21,21
20 KNT = 1
   MPRF = MFEX
   INDX = MFEX*MXX-1
   MFEX = NSET(INDX)
   IF (MFEX-K7L) 19,24,19
21 GO TO (22,16), KNT
22 KNT = 2
24 INDX = MFA*MXX
   NSET(INDX) = MPRF
   INDX = MPRF*MXX-1

```



```

      NSET(INDX) = MFA
      GO TO (17,26), KNT
5     OUT = 0.
      INDX = (MLC(JQ)-1)*IMM
      DO 32 I=1,IMM
      INDX = INDX+1
32    QSET(INDX) = 0.
      INDX = (MLC(JQ)-1)*MXX
      DO 1300 I=1,IM
      INDX = INDX+1
1300  NSET(INDX) = 0
      INDX = MLC(JQ)*MXX
      JL = NSET(INDX-1)
      JK = NSET(INDX)
      IF (JL-KOL) 33,34,33
33    IF (JK-KLE) 35,36,35
35    INDX = JK*MXX-1
      NSET(INDX) = JL
      INDX = JL*MXX
      NSET(INDX) = JK
37    INDX = MLC(JQ)*MXX-1
      NSET(INDX) = MFA
      NSET(INDX+1) = KLE
      IF (MFA-KOF) 234,235,235
234   INDX = MFA*MXX
      NSET(INDX) = MLC(JQ)
235   MFA = MLC(JQ)
      MLC(JQ) = MFE(JQ)
      XNQ = NQ(JQ)
      ENQ(JQ) = ENQ(JQ)+XNQ*(TNOW-OTIME(JQ))
      VNQ(JQ) = VNQ(JQ)+XNQ*XNQ*(TNOW-OTIME(JQ))
      OTIME(JQ) = TNOW
      NQ(JQ) = NQ(JQ)-1
      GO TO 99
36    INDX = JL*MXX
      NSET(INDX) = KLE
      MFE(JQ) = JL
      GO TO 37
34    IF (JK-KLE) 38,39,38
38    INDX = JK*MXX-1
      NSET(INDX) = KOL
      MLE(JQ) = JK
      GO TO 37
39    MFE(JQ) = 0
      MLE(JQ) = 0
      GO TO 37
100  CALL ERROR(88)
99   RETURN
      END
      SUBROUTINE SUMMARY
      COMMON ID,IM,INIT,JFVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
      INQ,NORPT,NOT,NPOMS,NPUN,NRUNS,NSTAT,OUT,ISFED,TNOW,
      TPEG,TFIN,MXX,NPRT,NCRDR,NEP,VNQ(4),IMM,MAXOS,MAXNS
      COMMON ATR19(10),ENQ(4),INN(4),JCFLS(20,22),KPRNK(4),MAXNO(4),
      IMFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),OTIME(4),SSUMA
      2(10,5),SUMA(1000,5),NAMF(6),NPROJ,MOY,NDAY,NYR,JCLR,JTRIB(12)
      WRITE (NPRNT,21)
21   FORMAT (1H1,39X,234**GAS? SUMMARY REPORT**/)
147  IF (NCLCT) 5,60,66
5    WRITE (NPRNT,199)

```

```

199 FORMAT (///36X,26HERROR EXIT, TYPE 99 ERROR.)
      STOP
66 WRITE (NPRNT,23)
23 FORMAT (//44X.18H**GENERATED DATA**/27X.4HCODE.4X.4HMFA.N.6X.8HSTD.
1DEV.5X.4HMIN.7X.4HMAX.5X.4HORS./)
      DO 2 I=1,NCLCT
      IF (SUMA(I,3)) 5,2,61
61 XS = SUMA(I,1)
   XSS = SUMA(I,2)
   XN = SUMA(I,3)
   AVG = XS/XN
   IF (XN-1.0001) 80,81,81
80 STD = 0.
   GO TO 82
81 STD =SQRT(AMAX1(0.,(XN*XSS-XS*XS)/(XN*(XN-1.))))
82 N = XN
   WRITE (NPRNT,24) I,AVG,STD,SUMA(I,4),SUMA(I,5),N
24 FORMAT (27X,I3.4F11.4,I7)
      ? CONTINUE
60 IF (NSTAT) 5,67,4
4 WRITE (NPRNT,29)
29 FORMAT (//44X.23H**TIME GENERATED DATA**/27X.4HCODE.4X.4HMFA.N.6X.
1RHSTD.DEV.5X.4HMIN.7X.4HMAX.7X.10HTOTAL TIME/)
      DO 6 I=1,NSTAT
      IF (SSUMA(I,1)) 5,6,72
72 XT = SSUMA(I,1)
   XS = SSUMA(I,2)
   XSS = SSUMA(I,3)
   AVG = XS/XT
   STD =SQRT(AMAX1(0.,XSS/XT-AVG*AVG))
   WRITE (NPRNT,30) I,AVG,STD,SSUMA(I,4),SSUMA(I,5),XT
30 FORMAT (27X,I3.5F11.4)
      6 CONTINUE
67 CONTINUE
75 DO 15 I=1,N00
15 CALL PRNTQ(I)
      RETURN
      END
      SUBROUTINE COLCT(X,N)
      COMMON ID,IM,INIT,JEVNT,JMNT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
IN00,NORPT,VOT,NPMS,NPUN,NRUNS,NSTAT,OUT,ISFEN,TNOW,
PTREG,TFIN,MAX,NPNT,NCRD2,NEP,VNO(4),IMM,MAXOS,MAXNS
      COMMON ATRIB(10),ENO(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
IMFE(4),MLC(4),MLE(4),NCELS(20),NO(4),PARAM(20,4),QTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12)
      IF (N) 2,2,1
      ? CALL ERROR(90)
1 IF (N-NCLCT) 3,3,2
3 SUMA(N,1) = SUMA(N,1)+X
  SUMA(N,2) = SUMA(N,2)+X*X
  SUMA(N,3) = SUMA(N,3)+1.
  SUMA(N,4) =AMIN1(SUMA(N,4),X)
  SUMA(N,5) =AMAX1(SUMA(N,5),X)
      RETURN
      END
      SUBROUTINE TMST(X,T,N)
      COMMON ID,IM,INIT,JEVNT,JMNT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
IN00,NORPT,VOT,NPMS,NPUN,NRUNS,NSTAT,OUT,ISFEN,TNOW,
PTREG,TFIN,MAX,NPNT,NCRD2,NEP,VNO(4),IMM,MAXOS,MAXNS
      COMMON ATRIB(10),ENO(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4)

```

```

1MFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIR(12)
IF (N) 2,2,1
2 CALL ERROR(91)
1 IF (N-NSTAT) 3,3,2
3 TT = T-SSUMA(N,1)
SCUMA(N,1) = SSUMA(N,1)+TT
SCUMA(N,2) = SSUMA(N,2)+X*TT
SCUMA(N,3) = SSUMA(N,3)+X*X*TT
SCUMA(N,4) = AMIN(SSUMA(N,4),X)
SCUMA(N,5) = AMAX(SSUMA(N,5),X)
RETURN
END
SUBROUTINE ERROR(J)
COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INNO,NORPT,NOT,NPRMS,NPUN,NRJNS,NSTAT,OUT,ISFFD,TNOW,
2TRFG,TFIN,MAX,NPRT,NCRDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
COMMON ATTR(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIR(12)
WRITE (NPRNT,100) J,TNOW
.100 FORMAT (//35X,16HERROR EXIT, TYPE,13,74 ERROR, //214 FILE STATUS AT
1 TIME,F10.4/)
CALL SUMRY
CALL OPUIT
CALL OUIIS(1)
3 RETURN
END
SUBROUTINE PSET(NSET,OSSET)
DIMENSION NSET(1),OSSET(1)
COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INNO,NORPT,NOT,NPRMS,NPUN,NRJNS,NSTAT,OUT,ISFFD,TNOW,
2TRFG,TFIN,MAX,NPRT,NCRDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
COMMON ATTR(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA
2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIR(12)
WRITE(6,3)(MFE(I),I=1,3)
3 FORMAT(1X,3I10)
DO 1 I=1,I1
J1=(I-1)*6+1
J2=J1+5
K1=(I-1)*4
K2=K1+3
1 WRITE(6,2)I,(NSET(J),J=J1,J2),(OSSET(K),K=K1,K2)
2 FORMAT(1X,7I5.4F10.5)
RETURN
END
SUBROUTINE PRNTQ(JQ)
RETURN
END
SUBROUTINE MONTP
RETURN
END
FUNCTION LOCAT(JSW,K,JATT,NSET,OSSET)
DIMENSION NSET(1),OSSET(1)
COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
INNO,NORPT,NOT,NPRMS,NPUN,NRJNS,NSTAT,OUT,ISFFD,TNOW,
2TRFG,TFIN,MAX,NPRT,NCRDR,NEP,VNQ(4),IMM,MAXQS,MAXNS
COMMON ATTR(10),ENQ(4),INN(4),JCFLS(20,22),KRANK(4),MAXNO(4),
1MFE(4),MLC(4),MLF(4),NCELS(20),NQ(4),PARAM(20,4),QTIME(4),SSUMA

```

```

2(10,5),SUMA(1000,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12)
GO TO (200,300,200,300),JSW
200 L=IMM
GO TO 400
300 L=MXX
400 IF(JATT-L)500,500,2000
500 IF(JSW-3) 500,800,900
600 IF(K-ID) 700,700,2000
700 LOCAT=(K-1)*L+JATT
RETURN
800 IF(K-MAXQS) 1000,1000,2000
900 IF(K-MAXNS) 1000,1000,2000
2000 CALL EPROR(R5)
1000 LOCAT=1+(K-JATT)/L
RETURN
END
SUBROUTINE QUIT(S(J)
WRITE(6,1)
1 FORMAT(' QUIT(S)')
STOP
END
SUBROUTINE DRAND(ISFFD,RNUM)
DATA IA/24298/,IC/99991/,M/199017/
IX=ISFFD*IA+IC
ISFFD=MOD(IX,M)
RNUM=FLOAT(ISFFD)/FLOAT(M)
RNUM=ABS(RNUM)
RETURN
END
FUNCTION RFX(A)
RFX=1.E20
IF(A.LE.0.)RETURN
CALL DRAND(ISFFD,R)
RFX=- (ALOG(R))/A
RETURN
END

```

SUMMARY OF FUTURE V/STOL OPERATING COSTS

In this appendix, the effects of stage length, external noise constraints, and fuel costs on the direct operating costs of future VTOL concepts are summarized.

In the period 1973 to 1976, NASA sponsored Conceptual Engineering Design studies of 1985-era commercial VTOL transports. The purpose of these studies was to assess the effects of new technology, rising fuel costs, and external noise constraints on future VTOL aircraft.

The design guidelines are as shown below in Table C.1.

Passengers	100 Maximum
Stage Length	200 NM
V Cruise	Minimum DOC
Hover	Out of Ground Effect One Engine Inoperative @ Sea Level 32.2°C
Initial Fabrication	1980 (Introduction to Service in 1985)
External Noise	95 PNdB 500 ft. Sideline
Aircraft Optimization	Min DOC
Cruise Altitude	Min DOC

Table C.1

The Direct Operating Costs were calculated using a 1973 NASA modification of the AIA method for calculating operating expense (Reference 15). These modifications appear below in Table C.2.

Year Dollars	1974
Avionics Cost	250,000 \$/Acft.
Airframe Cost	90 \$/16
Dynamic System Cost	80 \$/16
Engine Cost	280(HP <sup>.785</sup> ) \$/Rate SHP
Crew Costs	$\frac{.067}{1000}$ GW + 134 \$/hr.
Fuel	.02 \$/lb.
Oil	1.24 \$/lb.
Nonrevenue Factor	2%
Labor Rate	6 \$/hr.
Airframe Labor	AIA
Airframe Material	AIA
Engine Labor	.65 AIA
Engine Material	.65 AIA
Engine TBO	4500 Hr.
Dynamic System Labor	AIA
Dynamic System Materials	AIA
Dynamic System TBO	3000 Hr.
Maintenance Burden	150% Direct Labor
Depreciation	12 Years
Spares:	
Airframe	8%
Engines	40%
Dynamic System	25%
Utilization	2500 Hr.

Table C.2

For performance and cost calculations, a standard flight profile was issued. This is shown in Figure C.1.

Three helicopter configurations were studied. The designs include tandem rotor<sup>16</sup>, conventional and compound helicopters.<sup>17</sup> See Table C.3.

Two tilt rotor vehicles and short take-off derivatives were evaluated,<sup>16,18,19</sup> see Table C.4.

Although the baseline designs were not well defined, three lift fan configurations were evaluated.<sup>20,21,22</sup> A representative configuration is presented in Table C.5.

The direct operating cost for all configurations are plotted vs. stage length. These operating costs are shown in terms of dollars per hour and cents per available seat-statute mile. (Figures C.2-C.4).

The fuel cost used in this study (2¢/lb.) was typical for 1973. By 1975, fuel cost had risen to between 4 and 5¢/lb. It is projected that by the early 1980's fuel cost will be nearly 10¢/lb. The effects of this increase is shown in Figures C.5-C.7.

$$\text{DOC Ratio} = \frac{\text{DOC}}{\text{DOC with Fuel at 2¢/lb.}}$$

(200 nm Stage Length)

For each VTOL configuration (excluding lift fan), two derivative aircraft were designed to investigate the effects of sideline noise constraints. The sensitivity of DOC to the noise constraints is illustrated in Figures C.8 and C.9. One aircraft 5 PndB quieter and one 5 PndB noisier. The ratio is for DOC at a 200 nm stage length.

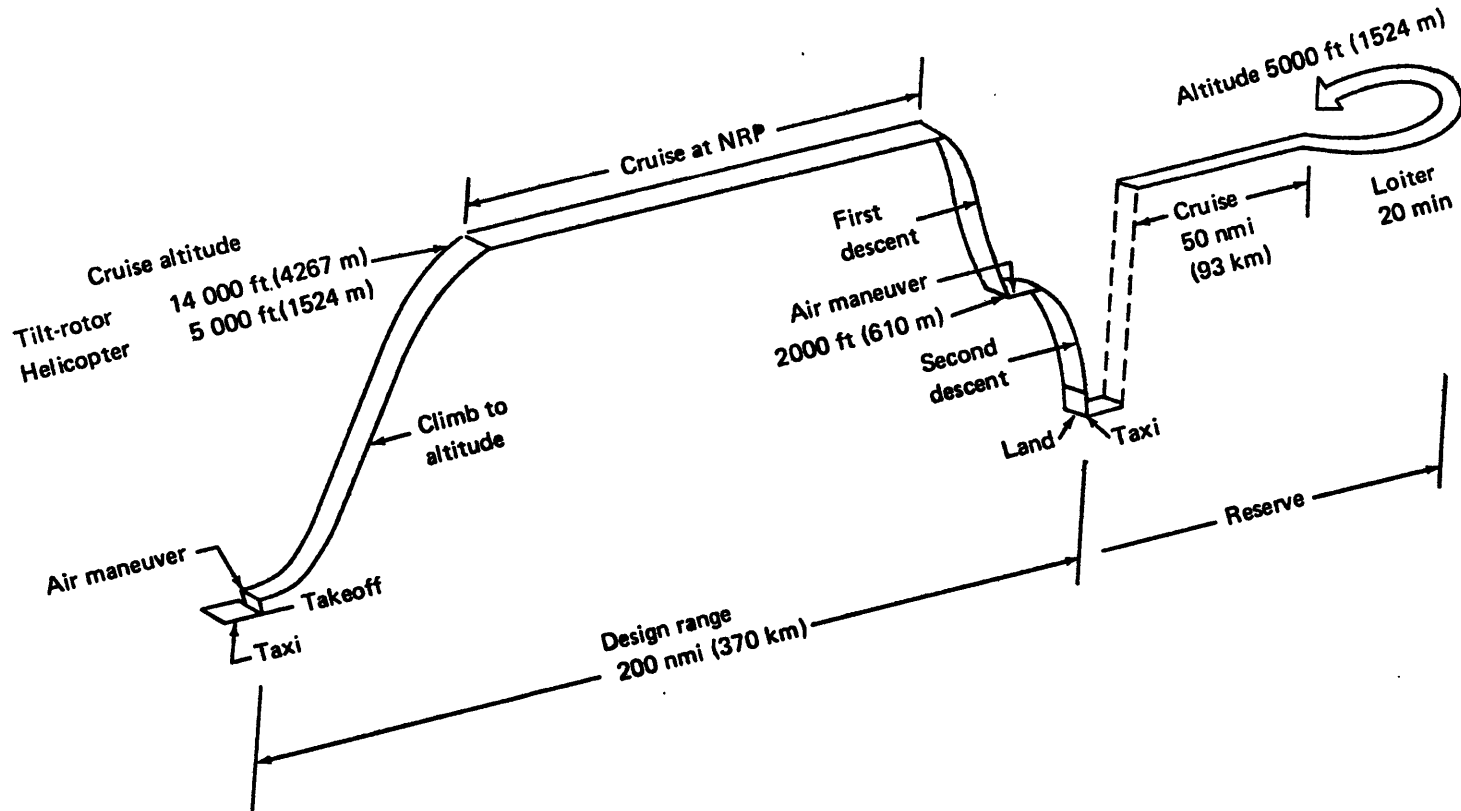


Figure C.1 - Design Short-Haul Mission



ADVANCED HELICOPTERS

<u>Vehicle Type Study</u>	<u>Tandem Rotor Boeing</u>	<u>Conventional Sikorsky</u>	<u>Compound Sikorsky</u>
Weight Empty (lbs)	40181	34374	49564
W Max Gross	67175	58137	75926
Max Pass	100	100	100
Vcruise KTS	165	173	250
ALTcruise FR	5000	5000	14000
PNdB @500 ft.	92.3	93.5	95
Area T/O @ 95 PNdB	.07	.075	.156
Area Landing	.535	.063	.088
Engines	3	3	3
SHP	14472	10605	21979
Block Time Hrs.	1.337	1.331	.958
Block Fuel lbs.	5093	3404	5379
DOC ¢/as-sm	3.53	3.17	3.30
Initial Cost x 10 <sup>6</sup>	4.17	3.95	5.67

Table C.3

TILT ROTORS

<u>VEHICLE TYPE</u>	<u>VTOL BOEING</u>	<u>STOL BOEING</u>	<u>VTOL BELL</u>	<u>VTOL BELL</u>
Weight Empty	50068	45023	33216	42720
W Max T/O	74749	68493	44848	64300
Max Pass	100	100	45	100
V Cruise	349	310	296	248
ALT Cruise	14000	14000	11000	20000
PNdB @ 500 ft.	98.2	101.3	97.2	96
Area T/O @95dB	.09	.115	.083	.191
Area Landing	.15	.14	.078	.098
Engines	4	4	4	4
SHP	16480	11144	9072	6765
Block Time	.742	.82	.858	1.015
Block Fuel	3157	2392	2015	1888
DOC $\phi$ /as-sm	2.41	2.31	4.87	2.77
Initial Cost x 10 <sup>6</sup>	5.15	4.62	3.98	4.90

Table C.4

186  
LIFT FAN

Weight Empty	80,000 Lbs.
Max T/O Weight	105,000 Lbs.
Max Pax	100
V Cruise	M = .75
ALT Cruise	24,000 Ft.
PNdB @ 5W Ft.	97 - 100
Area T/O 95 PndB	~ .11 Mile <sup>2</sup>
Landing	~ .12
Engines	4 - 6
Thrust	~ 120,000 Lbs.
Block Time	~ .6 Hr.
Fuel	~ 7,200 Lbs.
DOC	~ 3.34 ¢/as-sm
Initial Cost	9 - 12 X 10 <sup>6</sup> \$

Table C.5

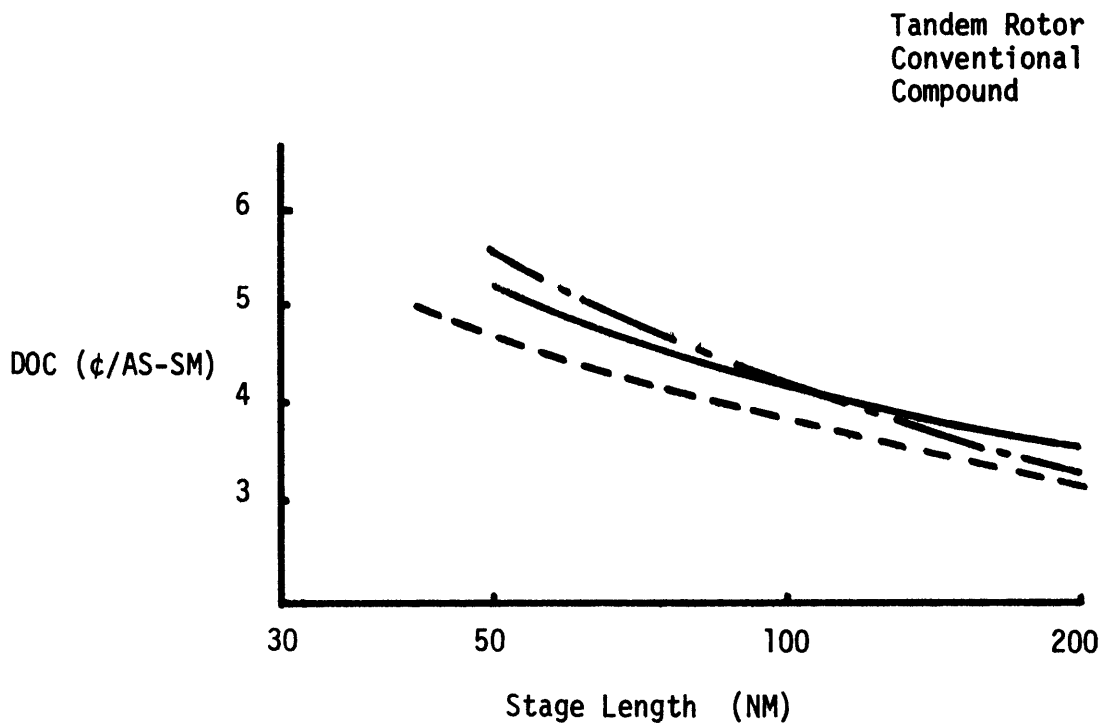
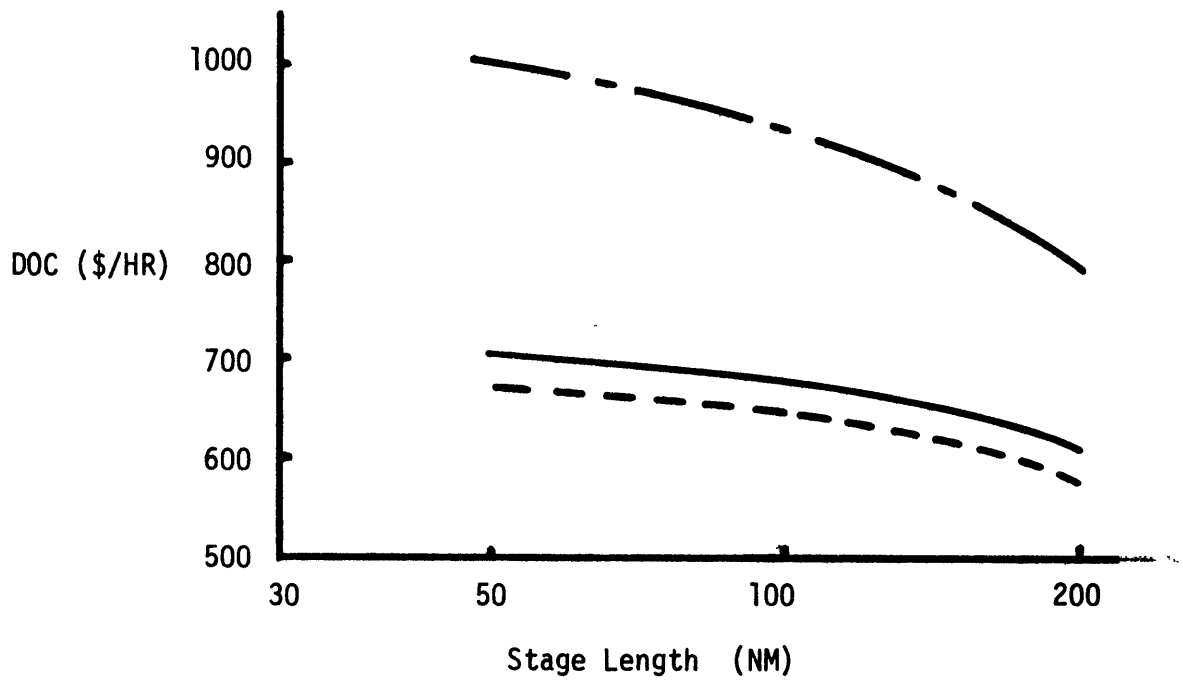


Figure C.2 - Advanced Helicopter Direct Operating Costs

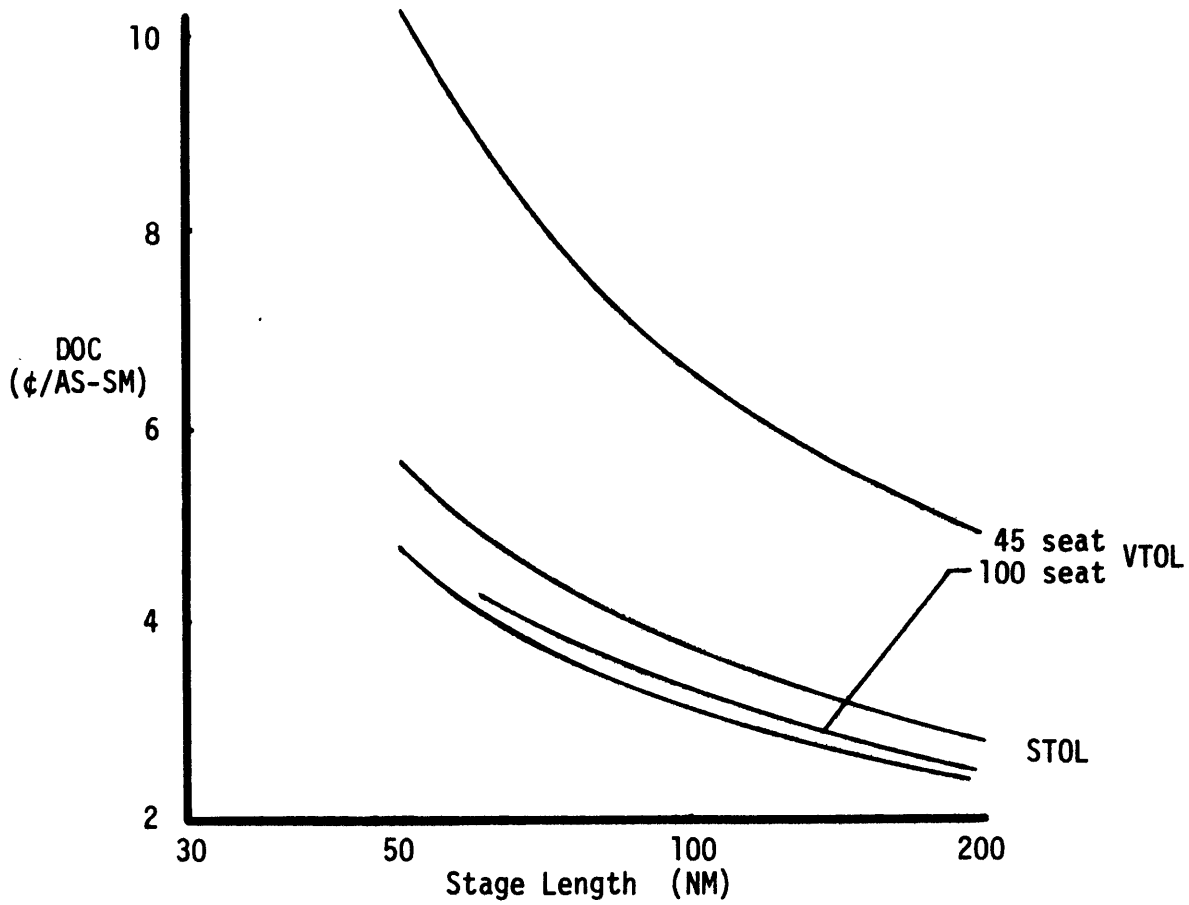
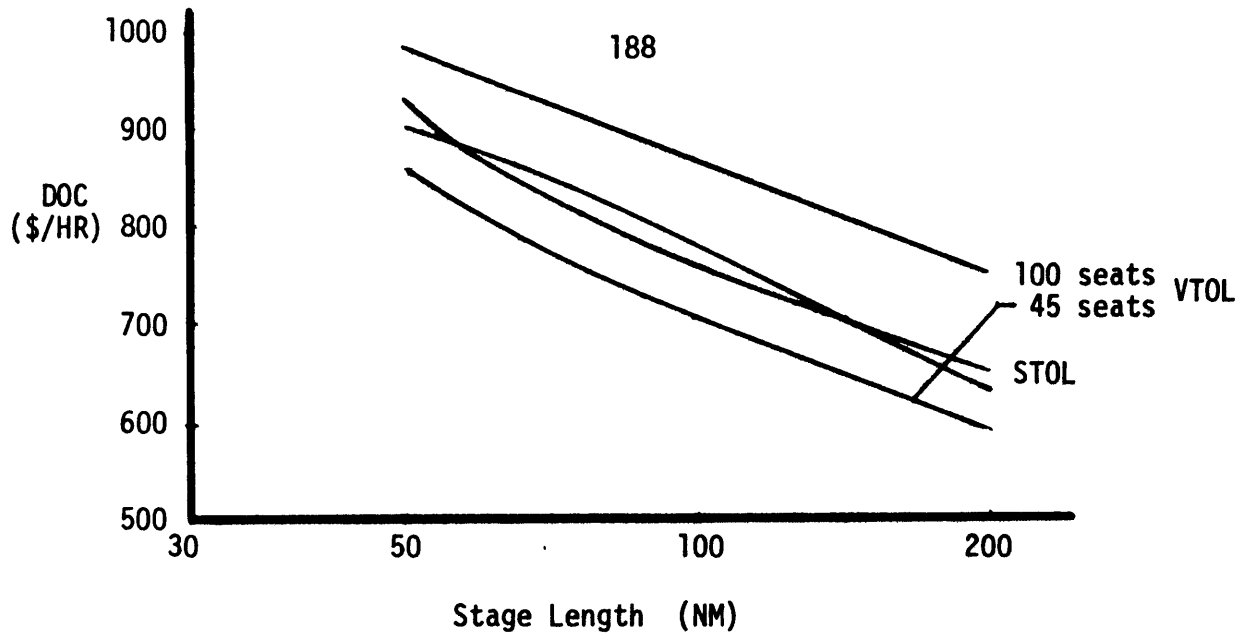


Figure C.3 - Tilt Rotor Direct Operating Costs

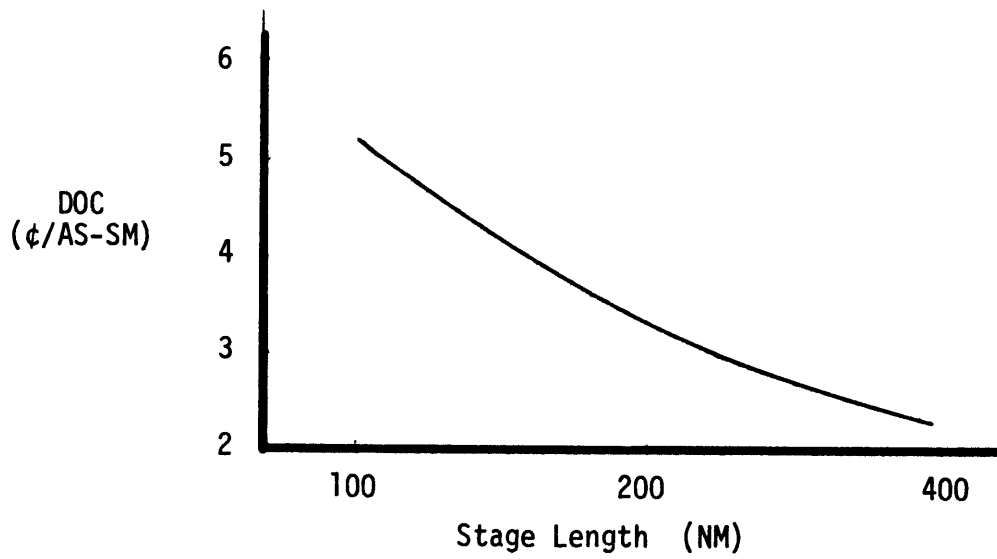
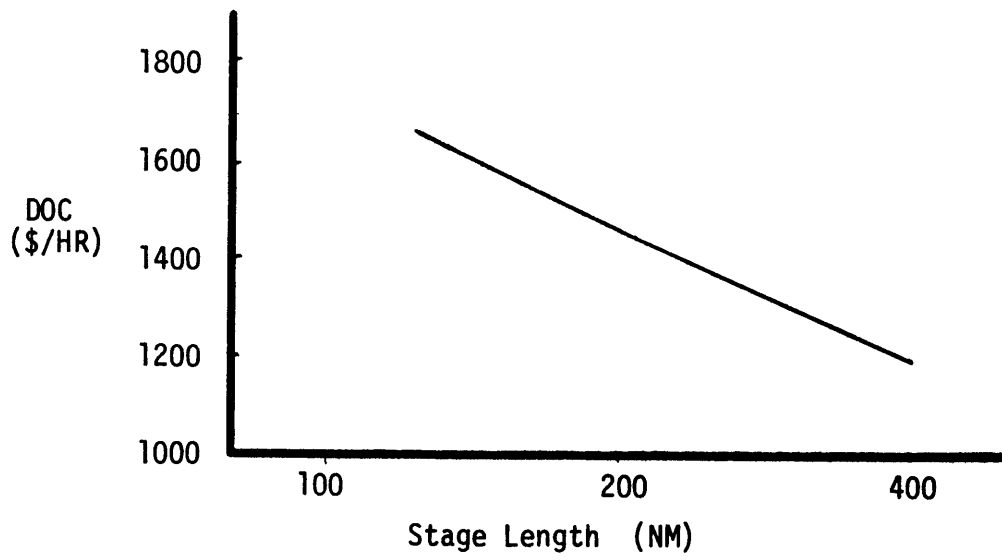


Figure C.4 - Lift Fan Direct Operating Costs

!

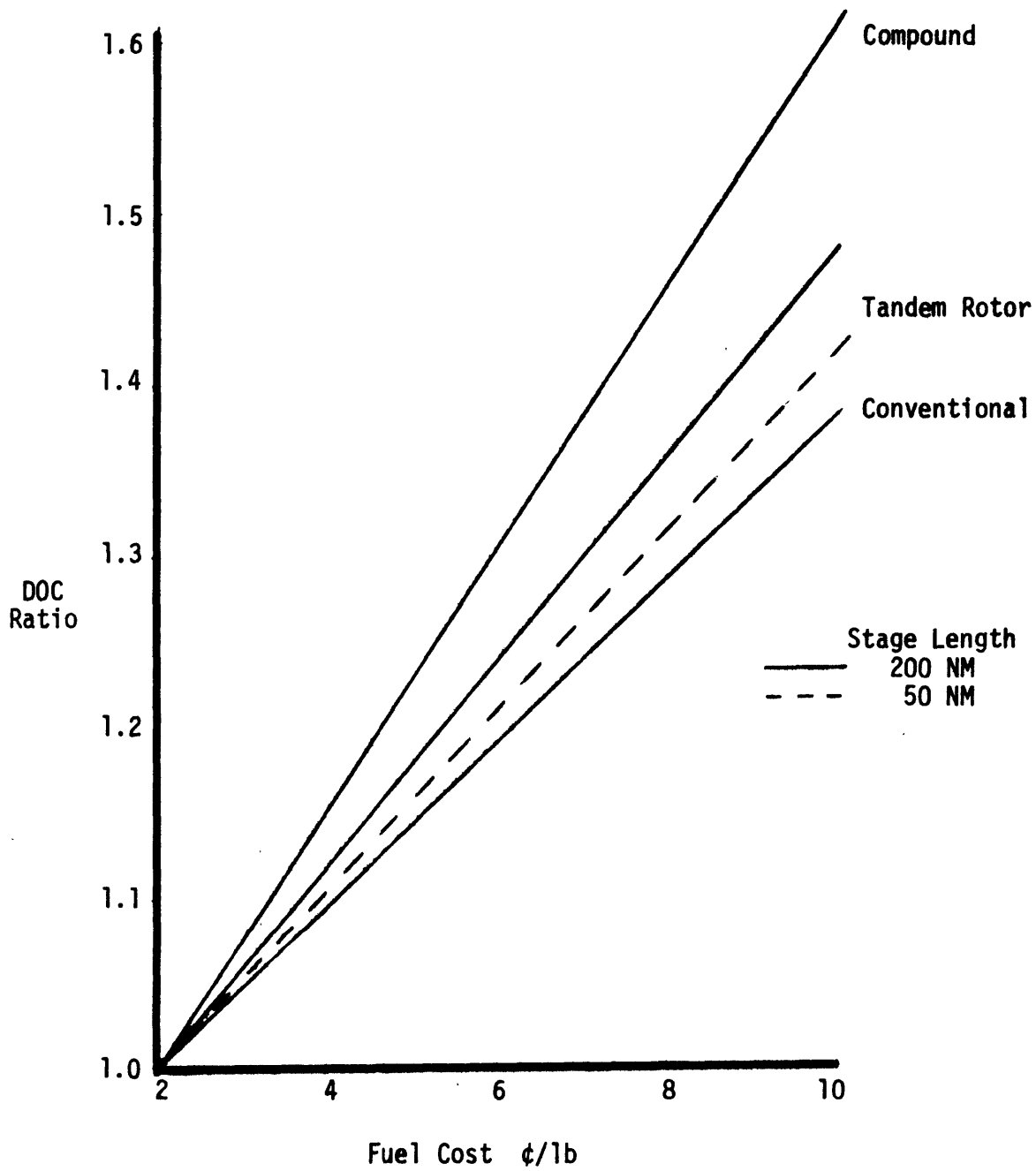


Figure C.5 - The Effect of Fuel Cost on Advanced Helicopter DOC

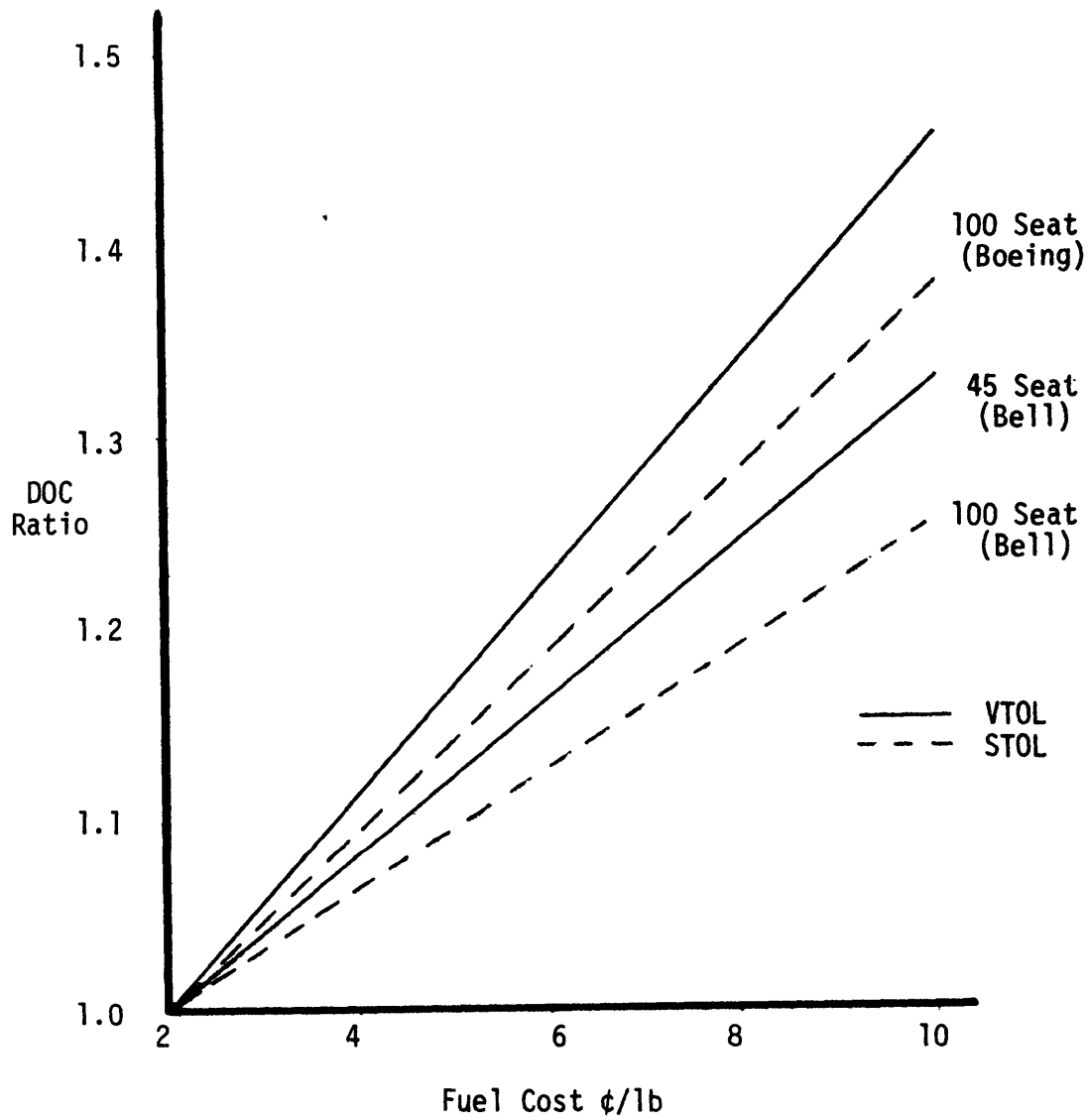


Figure C.6 - The Effect of Fuel Cost on Tilt Rotor DOC



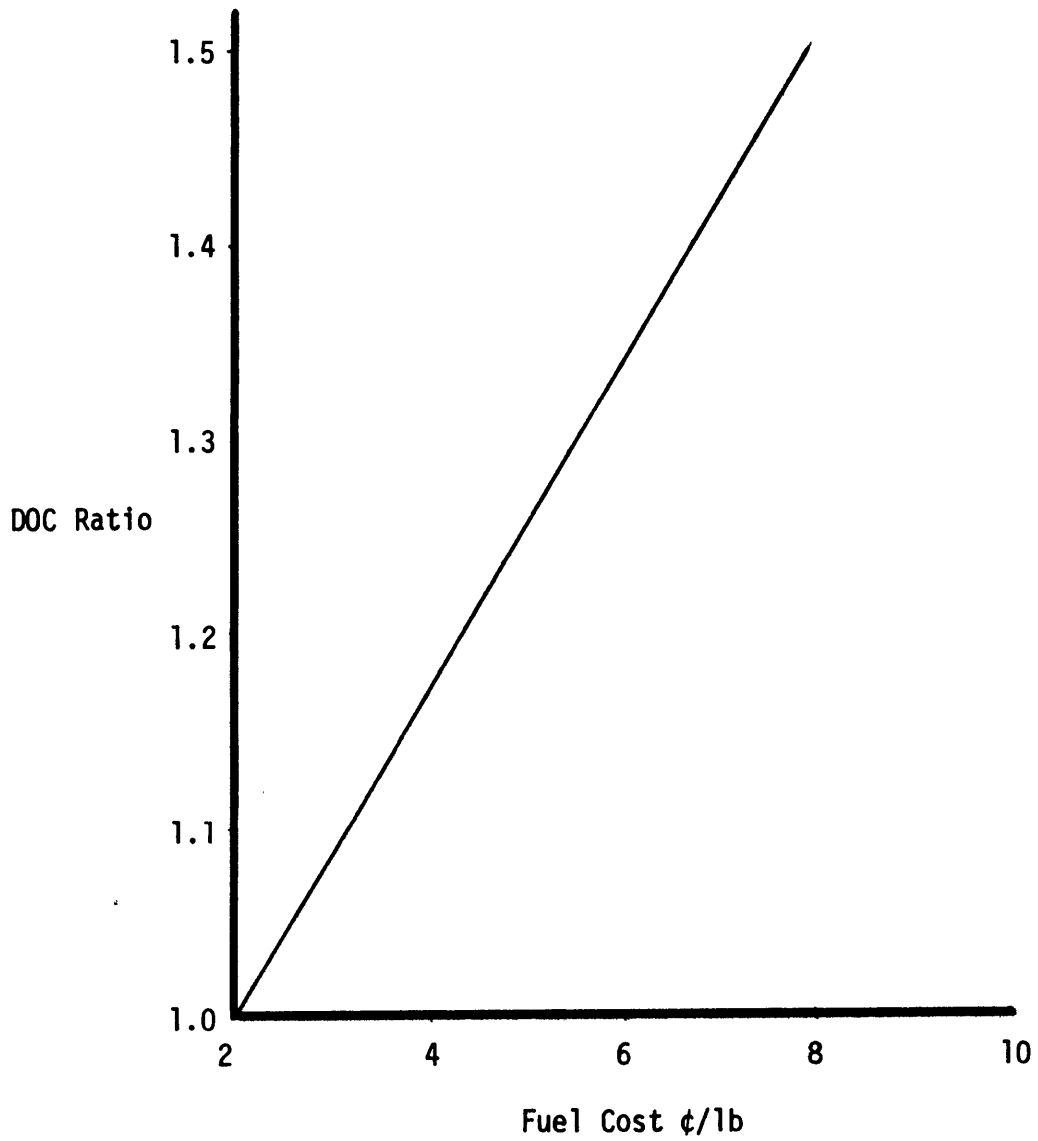


Figure C.7 - The Effect of Fuel Cost on Lift Fan DOC

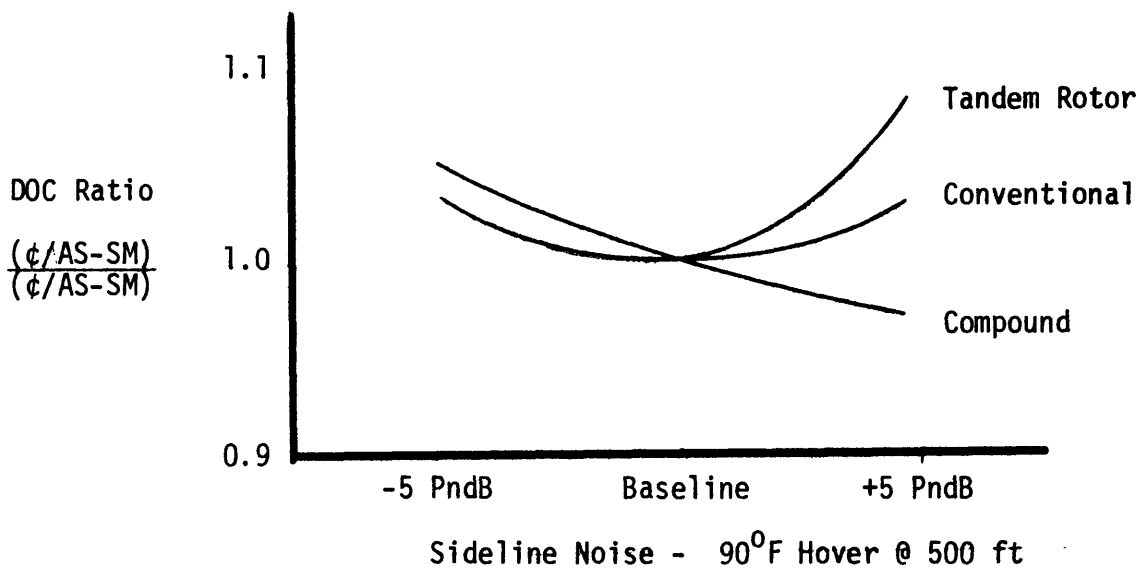
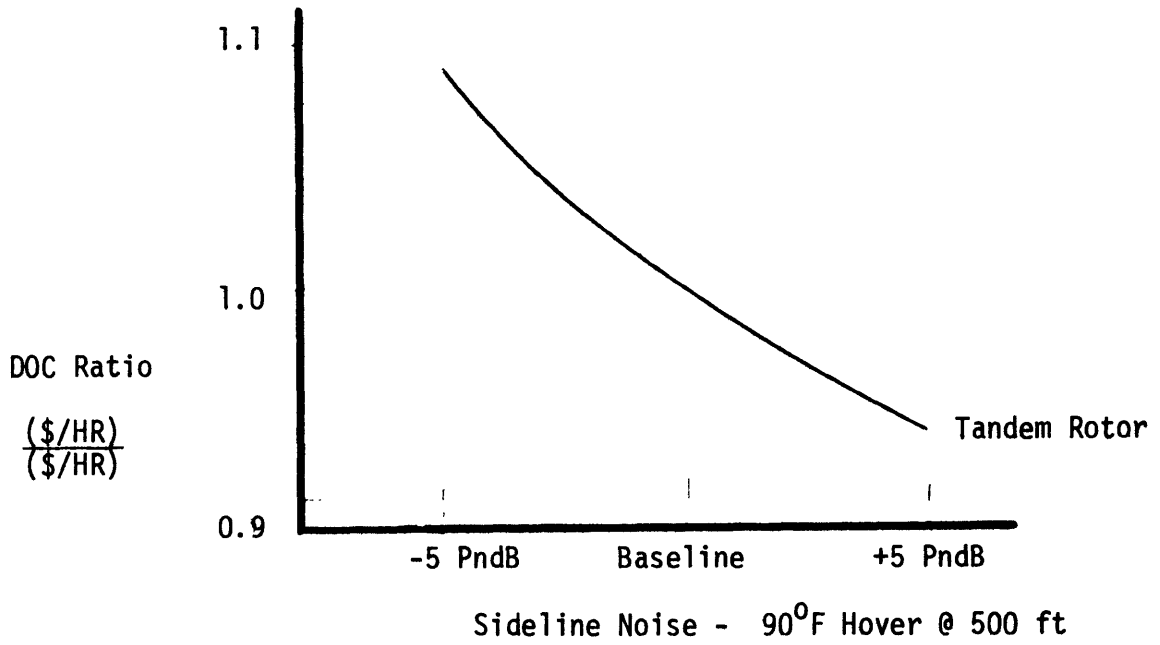


Figure C.8 - The Effect of Sideline Noise Constraints on Advanced Helicopter DOC

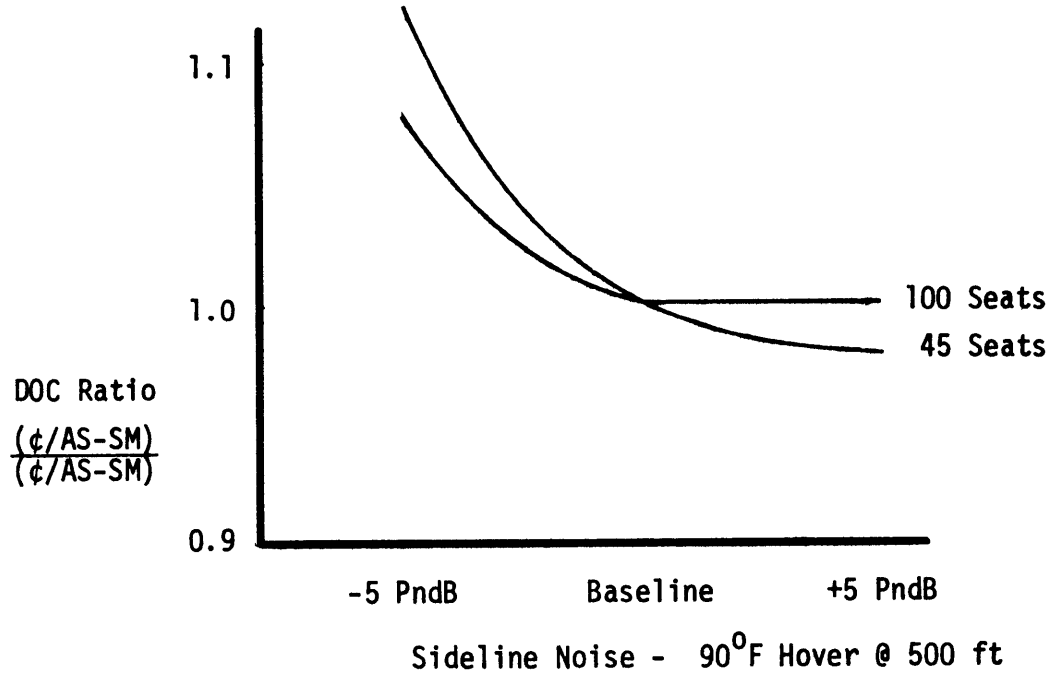
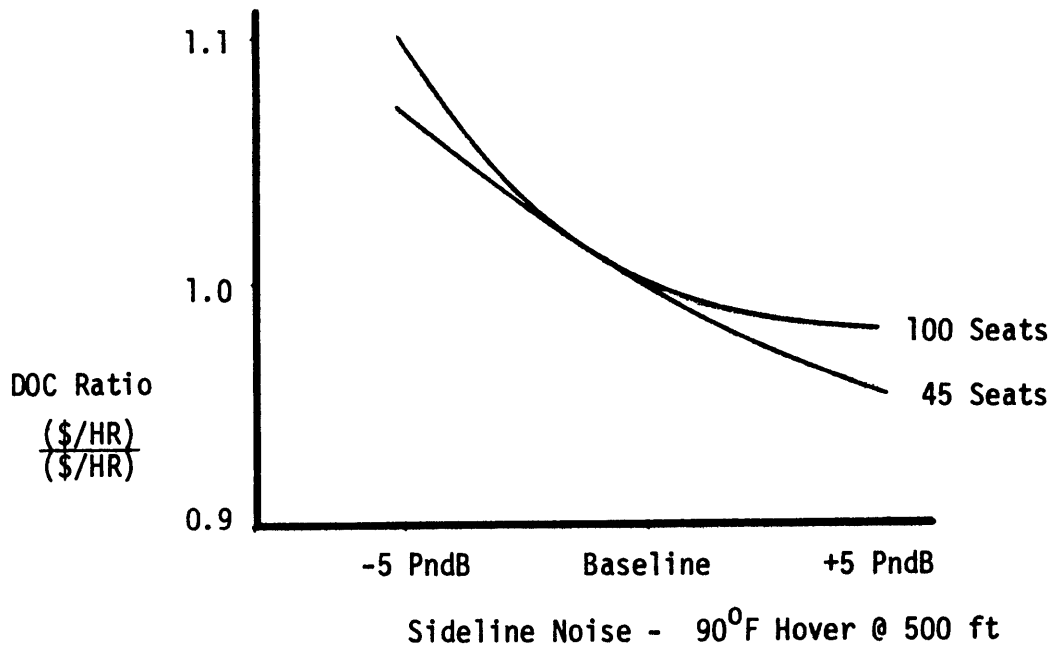


Figure C.9 - The Effect of Sideline Noise Constraint on Tilt Rotor DOC

REFERENCES

1. Concept Studies for Future Intercity Air Transportation Systems, MIT FTL Report R-70-2, December, 1970.
2. Mann, Robert W., "Short Haul Helicopter Demonstration Project," Department of Aeronautics and Astronautics, FTL Report R76-2-1, Cambridge, Mass., April 1976.
3. Mann, Robert W., "Systems Optimization in Ultra Short Haul Air Transportation," Department of Aeronautics and Astronautics, S.M. Thesis, M.I.T., Cambridge, Mass., May, 1977.
4. Swan, William, "The Compleat FA-4 Memo," Department of Aeronautics and Astronautics, FTL Memorandum M72-10, Cambridge, Mass., August, 1972.
5. Barnett, A. I., "Control Strategies for Transportation Systems with Nonlinear Waiting Costs," *Transportation Science*, Volume 12, No. 12, May, 1978, pp. 119-136.
6. Miller, R. H., "Some Air Transportation Concepts for the Future," *The Aeronautical Journal of the Royal Aeronautical Society*, Vol. 75, No. 727, July, 1971.
7. Pritsker, A. A. B. and Kiviat, P. J., Simulation with GASP II, Prentice-Hall, Inc. (1969).
8. Faulkner, H., "The Cost of Noise Reduction in Helicopters," MIT FTL Report R71-5, November, 1971.
9. Larson, R. C., Odoni, A. R., Analysis of Urban Service Systems, in print.
10. Osuna, E. E. and Newell, F. G., "Control Strategies for an Idealized Public Transportation System," *Trans. Sci.* 6, 52-72 (1972).
11. Devanney, John W., "Transportation Scheduling under Multi-Dimensional Demands," Department of Civil Engineering, Ph.D. Dissertation, M.I.T., Cambridge, Mass., 1967.
12. Simpson, R. W., "Computerized Schedule Construction for an Airline Transportation System," MIT Flight Transportation Lab. Report 8-T-66-3, December, 1969.

13. Vom Saal, Robert J., "Fast Analytic Solutions for Optimal Schedules under Time-Varying Demands," Department of Naval Architecture, S.M. Thesis, M.I.T., Cambridge, Mass., October, 1968.
14. Smith, B. C., "Optimal Scheduling of Ultra Short Haul Air Transportation under Time-Varying Demand," Term Paper, 16.77, April, 1978.
15. Anon., "Standard Method of Estimating Comparative Direct Operating Costs of Turbine Powered VTOL Transport Aircraft, Aerospace Industries Association of America, Inc., Vertical Lift Aircraft Council, December, 1968.
16. Magee, J. P., et al, "Conceptual Engineering Design Studies of 1985-era Commercial VTOL and STOL Transports that Utilize Rotors," NASA CR-2545, Boeing Vertol Company, May, 1975.
17. Kefford, N. F. and Munch, C. L., "Conceptual Design Study of 1985 Commercial VTOL Transports that Utilize Rotors, Summary," NASA CR-2532, Sikorsky Aircraft, for Ames Research Center, May, 1975.
18. Detore, J. A. and Sambell, K. W., "Conceptual Design of 1985 Commercial Tilt Rotor Transports, Volume I - VTOL Design Summary," NASA CR-2544, Bell Helicopter Company, for Ames Research Center, May, 1975.
19. Sambell, K. W., "Conceptual Design Study of 1985 Commercial Tilt Rotor Transports, Volume II - STOL Design Summary," NASA CR-2690, Bell Helicopter Company, for Ames Research Center, April, 1976.
20. Anon., "Conceptual Design Studies of a V/STOL Civil Lift Fan Transport including the Effect of Size and Fan Pressure Ratio," NASA CR-2426a, McDonnell Aircraft Company, for Ames Research Center, July, 1974.
21. Cavage, R. L., et al, "Conceptual Design Study of Improved 1985 Remote Lift-Fan V/STOL Commercial Transports," NASA CR-2481, Rockwell International Corp., for Ames Research Center, January, 1975.
22. Zabinsky, J. M., et al, "V/STOL Lift Fan Commercial Short Haul Transports, Continuing Conceptual Design Study," NASA CR-2437, Boeing Commercial Airplane Company, for Ames Research Center, July, 1974.