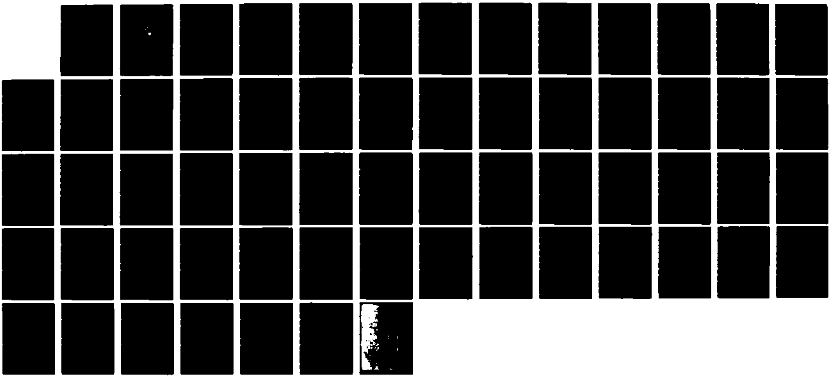
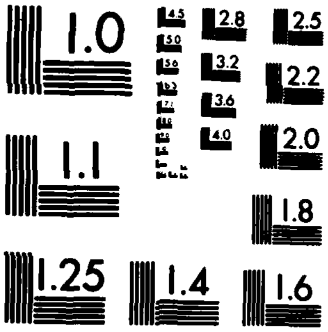
AD-A127 869   COMPARISON OF WAYS TO USE WEIGHTED FACTORS FOR   1/1
DEVELOPING VEHICLE SCHEDULES IN A MASS TRANSIT SYSTEM
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA R A DUGUID
UNCLASSIFIED MAR 83   F/G 13/6   NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPARISON OF WAYS TO USE WEIGHTED FACTORS FOR DEVELOPING VEHICLE SCHEDULES IN A MASS TRANSIT SYSTEM

by

Roger Alan Duguid

March 1983

Thesis Advisor: Lawerence Bodin

Approved for public release; distribution unlimited

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A127 869

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Comparison of Ways to Use Weighted Factors for Developing Vehicle Schedules in a Mass Transit System

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis
March, 1983

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Roger Alan Duguid

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE
March 1983

13. NUMBER OF PAGES
58

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Vehicle Scheduling
Matching
Mass Transit System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Traditionally, fleet vehicle schedules for mass transit systems are determined by using a minimum cost flow model. However, with constraints such as an upper bound on the number of lines that a vehicle can service in a vehicle block, the minimum cost flow structure is lost. Two heuristic procedures, a matching-based procedure and a time increment procedure, are developed for scheduling a fleet of vehicles under these additional constraints. These procedures attempt to minimize the average number of lines a vehicle block will traverse while maintaining a high average number of trips per vehicle schedule, low deadhead and waiting times and a minimum number of vehicles to service a timetable. Both procedures minimize a weighted sum cost function and have been tested on two databases including the Monterey-Salinas Transit system in California. Solutions comparable to the present vehicle schedules for the Monterey-Salinas Transit system were obtained using these procedures.

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification

By
Distribution/
Availability Codes

Dist	Avail and/or Special
A	

Approved for public release; distribution unlimited.

Comparison of Ways to Use Weighted Factors for Developing Vehicle Schedules in a Mass Transit System

by

Roger Alan Duguid
Lieutenant Commander, United States Navy
B.A., Washburn University of Topeka, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March, 1983

Author: Roger Alan Duguid

Approved by: Lawrence Bodin
Thesis Advisor

R K Wood
Second Reader

Chairman, Department of Operations Research

Kneale T. Marshall
Dean of Information and Policy Sciences

ABSTRACT

Traditionally, fleet vehicle schedules for mass transit systems are determined by using a minimum cost flow model. However, with constraints such as an upper bound on the number of lines that a vehicle can service in a vehicle block, the minimum cost flow structure is lost. Two heuristic procedures, a matching-based procedure and a time increment procedure, are developed for scheduling a fleet of vehicles under these additional constraints. These procedures attempt to minimize the average number of lines a vehicle block will traverse while maintaining a high average number of trips per vehicle schedule, low deadhead and waiting times and a minimum number of vehicles to service a timetable. Both procedures minimize a weighted sum cost function and have been tested on two databases including the Monterey-Salinas Transit system in California. Solutions comparable to the present vehicle schedules for the Monterey-Salinas Transit system were obtained using these procedures.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

For a mass transit system, the scheduling of drivers and vehicles is a problem that can be solved with numerous procedures. This thesis only addresses the vehicle scheduling problem but it does so in a way that should assist in the solution to the driver scheduling problem.

A. BACKGROUND

The state-of-the-art in the scheduling of drivers and vehicles for a mass transit system has advanced from primitive, but yet effective, manual methods to computerized procedures. The manual methods have produced reasonable solutions but have several disadvantages.

1. The time required to produce a solution may be lengthy (several weeks).
2. Extra constraints are difficult to handle.
3. Alternate solutions cannot be tested quickly and effectively.

Furthermore, the quality of a manual solution for a complex problem such as the vehicle scheduling problem with side constraints can be dependent upon the experience level of the planner or scheduler and it is becoming extremely difficult to train new schedulers.

With the advent of computerized procedures, good feasible solutions can be quickly found and multiple feasible solutions can be efficiently derived by changing the parameters of the model. Moreover, computerized procedures can reduce the effort and time necessary to train the novice scheduler.

Many of the earlier computerized procedures were developed along the lines of run cutting which takes a vehicle schedule and separates the schedule into segments or pieces for solving the driver scheduling problem. The first large scale computerized implementation of run cutting was called RUCUS. The RUCUS system for scheduling urban mass transit drivers and vehicles, [Ref. 1], was developed in the late 1960's and field tested in the early 1970's. The first version of RUCUS was made available to industry in 1973. Since RUCUS was developed under the sponsorship of the Urban Mass Transportation Administration (UMTA) of the U.S. Department of Transportation, it was intended to be a general package available to and usable by a wide variety of transit agencies. The initial experiences with RUCUS were disappointing in the sense that few agencies were able to successfully use it without significant modification. The later version of RUCUS (called RUCUS2) is much easier to use and operates in a more "user friendly" environment. However, RUCUS2 was just recently released by UMTA so that its success in the field is still too early to determine. Other approaches to solving the driver and vehicle scheduling problems use set partitioning and set covering methods, [Ref. 2]. Except in isolated cases, these procedures have not yet been used in an operational environment. Heuristic approaches for solving these problems, [Ref. 3], are the concurrent scheduler, the service profile decomposition, and the matching-based algorithm. Again, these procedures have only been used in the field in isolated cases. This thesis will adapt some of the matching-based procedures in [Ref. 3] and [Ref. 4] to solve the vehicle scheduling problem with interlining. This problem is described later.

B. THE ALGORITHMIC PHILOSOPHY

The algorithmic philosophy that is traditionally used in scheduling vehicles and drivers for mass transit systems is to solve the vehicle scheduling problem first, perform a run cut and then solve the driver scheduling problem. Recently, this philosophy has come into question by some since driver costs dominate vehicle operating cost (generally driver costs can be well over half the operating cost of a system). By solving the vehicle scheduling problem first, the driver scheduling solution is "locked into" the vehicle schedule solution. A discussion of the recent work in this area can be found in [Ref. 2] and [Ref. 4]. Thus, the question arises whether it is better to develop a procedure that simultaneously schedules vehicle blocks and driver pieces or to develop vehicle schedules which anticipate the potential run cut. This analysis has been attempted [Ref. 4] but has not yet been completed. Preliminary results indicate that such an approach can be very effective.

C. OVERVIEW OF THESIS

The procedures investigated in this thesis were tested on a database from the Monterey portion of the Monterey-Salinas Transit (MST) system and one other database that was artifically constructed. These approaches only dealt with vehicle scheduling, not driver scheduling. By appropriately selecting the weights for a weighted sum cost function and creating a network with the feasible arcs having the costs associated in these procedures, the scheduling of the vehicles can be manipulated so as to reduce the total deadhead time of the vehicles, to reduce the total waiting time of the vehicles, to reduce the average number of lines a vehicle can traverse and to increase, as much as possible, the average number of trips per vehicle block.

The initial work on the vehicle scheduling from the MST database was carried out by another Naval Postgraduate School student, LCDR M. L. Mitchell, and his results are compared to the results obtained from the two approaches investigated here. This initial study has significantly influenced the work in this thesis.

In Chapter II, the definitions of the mass transit vehicle scheduling problem are given and in Chapter III, the vehicle scheduling problem is formulated mathematically. In Chapter IV, the heuristics for solving the problem are developed and in Chapter V, the computational results are displayed. The conclusions and suggested areas for further investigation are in Chapter VI.

II. PROBLEM DEFINITION

A. THE VEHICLE SCHEDULING PROBLEM

The problem studied in this thesis is the vehicle scheduling problem. This problem's principal characteristic is that each task to be serviced has a specific time of day when the task is to begin and a specific time of day when it is to end. This should be contrasted with vehicle routing problems where the tasks to be serviced do not have _a priori_ specified times for beginning and ending service. A vehicle schedule must be feasible in both time and space since a vehicle cannot be at two locations at the same time. A discussion of the contrast between vehicle routing and vehicle scheduling problems is given in [Ref. 4].

The sequencing of the vehicle activities in both time and space is at the heart of vehicle scheduling problems. The real-world constraints that commonly determine the complexity of vehicle scheduling problem are the following:

1. The length of time a vehicle may be in service before returning to the depot for servicing or refueling.
2. The servicing of certain tasks by specific vehicle types.
3. The number of depots where vehicles may be housed.

The assumptions for this thesis which are deemed reasonable in the context of a mass transit system are the following:

1. There is no upper bound on the length of a vehicle schedule.
2. All vehicles are identical.
3. All vehicles are housed at the same depot.
4. Any task can be serviced by any vehicle.

The objective used in most analyses of this type is to minimize total number of vehicles or total deadhead time (deadhead time is defined in the next section). Because of the side constraints that must be satisfied, other objectives may be more applicable. In this study, the objective used attempts to minimize a modified linear combination of vehicle deadhead time, waiting time, average number of lines in a vehicle block and average number of lines/number of trips in a vehicle block. This objective is defined in more detail in Chapter III.

B. DEFINITIONS

Mass transit systems are made up of lines and line schedules. A <u>line</u> is defined to be a specification of a start location, an end location and any intermediate stop locations over which service is to be provided. A <u>trip</u> is a one-way traversal of the line from the start location to the end location through the intermediate stops in the order specified. A <u>line schedule</u> is defined to be the time schedule for a line over a single day which may consist of one or more trips. Figure 2.1 is an example of a line schedule for a line. The <u>timetable</u> for the transit system is the collection of line schedules for all lines in the transit system.

Typically, a vehicle will depart the garage or depot in the morning, travel to a start location of a line, traverse several trips, travel to the start location of another line, traverse several trips of this line and continue in this manner until it finally returns to the garage or depot. Figure 2.2 is an example of a vehicle schedule (Note: WT and DHT are defined below). Most, if not all, mass transit systems have a high period of requirements for service in the morning rush (called the <u>AM peak</u>), a reduced level for

LINE 22 BIG SUR
Monterey/Carmel to Big Sur

Trip No	Monterey Transit Plaza	Carmel 6th & Mission	Point Lobos State Reserve	Pfeiffer Big Sur State Park	Nepenthe
1	6.00a	6.15a	6.45a	7.25a	7.35a
3	7.45a	8.00a	8.30a	9.10a	9.20a
5	9.30a	9.45a	10.15a	10.55a	11.05a
7	11.15a	11.30a	12.00p	12.40p	12.45p
9	1.00p	1.15p	1.45p	2.25p	2.35p
11	2.45p	3.00p	3.30p	4.10p	4.20p

LINE 22 Monterey
Big Sur to Carmel/Monterey

Trip No	Nepenthe	Pfeiffer Big Sur State Park	Point Lobos State Reserve	Carmel 6th & Mission	Monterey Transit Plaza
2	7.45a	7.55a	8.35a	9.05a	9.15a
4	9.30a	9.40a	10.20a	10.50a	11.05a
6	11.15a	11.25a	12.05p	12.35p	12.45p
8	1.00p	1.10p	1.50p	2.20p	2.35p
10	2.45p	2.55p	3.35p	4.05p	4.15p
12	4.30p	4.40p	5.20p	5.50p	6.10p

Figure 2.1 Example of a Line Schedule for Line 22.

service in the middle part of the day and then a period of high requirements in the late afternoon (called the _PM peak_). As a result of the AM and PM peaks, a vehicle can return to the garage during the day and reappear on the streets later in the day.

For the remainder of this thesis, the following definitions are used:

Vehicle Block: the work performed by a vehicle between traveling to and from the depot; i.e., if a vehicle left and returned to the depot several times during a day, its schedule for that day would consist of several vehicle blocks. The makeup of the vehicle blocks is important because, in the traditional way of scheduling drivers, these

Assignment for Vehicle 21

Block No	Line No	Trip No	Start Time	End Time	Start Location	End Location	WT	DHT
1	--	--	6.45a	6.57a	Garage	Plaza	0	12
1	7	4	6.57a	8.21a	Plaza	Plaza	0	0
1	5	4	8.30a	9.25a	Plaza	Plaza	9	0
1	7	11	9.27a	11.06a	Plaza	Plaza	2	0
1	--	--	11.06a	11.18p	Plaza	Garage	0	12
2	--	--	2.18p	2.30p	Garage	Plaza	0	12
2	5	16	2.30p	3.25p	Plaza	Plaza	0	0
2	7	24	3.27p	5.06p	Plaza	Plaza	2	0
2	4	20	5.15p	6.10p	Plaza	Plaza	9	0
2	--	--	6.10p	6.22p	Plaza	Garage	0	12

Figure 2.2 Example of a Vehicle Schedule.

vehicle blocks are broken up into pieces by a process called run cutting and these pieces are then used in driver scheduling. In forming the vehicle blocks, a scheduler should never lose sight of the eventual run cut to be performed. A poor vehicle schedule in terms of how the vehicle schedule, when cut, satisfies the constraints on the driver schedules can lead to a poor run cut and, hence, an expensive driver scheduling configuration. A description of run cutting can be found in [Ref. 1].

Vehicle Schedule: the combination of vehicle blocks that makes up a schedule for a single vehicle.

Deadhead Time (DHT): the time during which a vehicle is traveling but is not involved in revenue producing service,i.e., not traversing a trip on a line.

Waiting Time (WT): idle time before or after a trip when a vehicle is neither traversing a trip nor deadheading. In some transit agencies, waiting time is referred to as layover time.

Average Number of Lines per Vehicle Block (L/B): summation of the total number of lines for each vehicle block divided by the total number of vehicles blocks required.

Average Number of Trips per Vehicle Schedule (T/V): summation of the total number of trips for each vehicle schedule divided by the total number of vehicle schedules required.

Maximum Number of Lines (ML): maximum allowable number of lines any vehicle block can traverse in a given day. ML is a value that can be specified by the scheduler for the purpose of producing vehicle schedules which can be configured to assist in solving the driver scheduling problem.

Maximum Allowable Deadhead Time (MDHT): maximum deadhead time allowed for a vehicle between two trips. MDHT is a value specified by the scheduler for the purpose of preventing an excess amount of deadhead time in a vehicle block.

Maximum Allowable Waiting Time (MWT): maximum waiting time allowed for a vehicle between two trips. MWT is a value specified by the scheduler for preventing an excess amount of waiting time in a vehicle block.

A vehicle block is said to interline between two lines A and B if the vehicle is to traverse a trip on line A followed by a trip from line B. The vehicle schedule in Figure 2.2 is made up of two vehicle blocks. Thus, in this vehicle schedule, there is interlining between line 7, trip 4 and line 5, trip 4 and interlining between line 5, trip 4 and line 7, trip 11, etc. Vehicle block 1 covers trips from two lines, lines 5 and 7 and vehicle block 2 covers trips from three lines, lines 4, 5 and 7. To restrict the number of lines that a vehicle can service in a vehicle block is to restrict the degree of interlining in the block. The vehicle schedule displayed in Figure 2.2 would be infeasible if the constraint of traversing trips from only two lines on a vehicle block was applied.

In the procedures developed in this thesis, the maximum number of lines in a vehicle block is explicitly constrained. To the author's knowledge, no other published procedure developed so far tries to handle this constraint. Most, if not all, of the procedures have not handled this type of constraint since the interlining constraints destroy the simple network flow structure of the vehicle scheduling problem. However, these constraints should not be ignored since a very restrictive set of interlining constraints can significantly cost the transit agency in both the number of vehicles required and total deadheading and waiting time. Such results will be seen later in this thesis.

III. FORMULATION OF THE VEHICLE SCHEDULING PROBLEM

For the formulation of a vehicle scheduling problem, two types of scheduling problems should be understood as they apply to a network. The two types are those without side contraints and those with side constraints.

A. SCHEDULING PROBLEM WITHOUT SIDE CONSTRAINTS

Most procedures for solving the single depot vehicle scheduling problem partition the nodes (tasks) of an acyclic network into a set of paths in such a way that a specified cost (objective) function is minimized. This cost function is constructed to be additive over the arcs in the network. Each path in the network corresponds to the schedule for a single vehicle [Ref. 3]. The cost function generally minimizes either the number of paths since the number of required vehicles equals the number of paths or the total deadhead times of the vehicles.

An assumption is made that a timetable is given and for each trip i in the timetable, the start time, ST(i), end time, ET(i), start location, SL(i), end location, EL(i), line number, L(i), trip number, T(i), and a depot are specified. The location of the depot is given by the letters s and t (s and t are defined below). Figure 3.1 is an example of a timetable sorted by the start time of the tasks. Additionally, a deadhead time matrix, D=DD(EL(i),SL(j)), is given where DD(EL(i),SL(j)) is the deadhead time required to go from the end location of trip i, EL(i), to the beginning location of trip j, SL(j). Since a vehicle must go from the depot to the beginning of a trip or from the ending of a trip to the depot, there is an extra row and column in the

Start Time	End Time	Start Location	End Location	Line Number	Trip Number
5:45	6:40	2	1	12	1
6:00	7:06	2	1	12	2
6:09	6:40	3	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮
19:05	19:33	1	9	4	22

Figure 3.1 Timetable Input Example.

deadhead matrix corresponding to the deadhead time into and out of the depot.

For a vehicle scheduling problem without side constraints, it is possible to create a network from this data and solve the scheduling problem using a minimum cost flow algorithm. A mathematical formulation of the minimum cost flow model can be found in **[Ref. 5]**. The structure of this network is given as follows:

1. Each trip i in the timetable is represented by nodes +i and -i.
2. A supersource s and supersink t are defined.
3. Table I displays the arcs in the network. It is important to note that not all -i to +j arcs are feasible as discussed below.

In order for node i (trip i) to be scheduled, flow from node +i to -i is required. Thus, the lower bound (LB) and the upper bound (UB) on arc (+i,-i) equals 1. The cost/unit flow is 0. The cost D of leaving node -i and going to the depot (supersink t) is equal to the deadhead time from EL(i) to the depot. Also, arc (-i,t) has a lower bound on flow of 0 and an upper bound on flow of inf(infinity). Similarly, the cost E of going from the depot (supersource s) to node

TABLE I

Definition of Arcs

From	To	LB	UB	COST
+i	-i	1	1	0
-i	t	0	inf	D
s	+i	0	inf	E
-i	+j	0	inf	C'
t	s	A	B	C

+i is equal to the deadhead time from the depot s to SL(i) and arc (s,+i) has a lower bound on flow of 0 and an upper bound on flow of inf(infinity). The cost C associated with going from t to s is 0 if minimizing deadhead time only or is equal to the capital cost of the vehicle if the objective is to minimize a modified linear combination of operating and capital cost. The lower bound for t to s, A, is either 0 or the minimum number of vehicles required. The upper bound value B is either the maximum number of vehicles allowed or infinity.

An arc from -i to +j is defined only if it is feasible to go from the end of trip i to the beginning of trip j. This will be feasible if Equation (3.1) is satisfied.

$$ST(j) - ET(i) - DD(EL(i), SL(j)) \geq 0 \qquad \text{Eqn}(3.1)$$

With no additional constraints, The cost C' from node -i to node +j is generally the deadhead time or the waiting time encountered in going from EL(i) to SL(j).

B. SCHEDULING WITH INTERLINING CONSTRAINTS

With side constraints such as maximum number of lines that can be traversed in a block, a different approach must be used since the mathematical structure of the minimum cost flow model is destroyed. A basic property of flow algorithms is that they do not have memory; i.e. they are only interested in the existence, quantity and cost of flow and not in the additional conditions such as length of flow path. In attempting to set up the above network when the problem has side constraints, it is impossible to know a priori if the arc from -i to +j will exist since the existence of this arc is dependent on the conditions on the path up to node i and the conditions on the path leading out of node j. To illustrate, suppose the path through -i includes trips from 3 different lines; trip j is a vehicle block by itself; trip j is from a different line; and there is an upper bound of 3 lines on any vehicle block. Then, an arc from node -i to node +j will not be feasible. If trip j is a trip from one of the lines already on the path to node -i, then this arc would be feasible. Flow algorithms are not able to test for these conditions.

In Chapter IV, two basic iterative algorithms are developed to solve the vehicle scheduling problem. The feature of these algorithms is that, unlike the vehicle scheduling problem without side constraints, the procedures define a sequence of networks, each network dependent upon the algorithm used (matching-based or time increment) and an arc between trips i and j is allowed only if interlining conditions are met in addition to the feasibility requirement of Equation (3.1). Additionally, if, out of trip i, there exists more than one feasible trip j, then a decision has to be made as to which trip j to select. The two algorithms utilize the same basic cost function for concatenating trips into partial vehicle blocks which allows for a decision to select the trip j that minimizes the cost if more than one trip is feasible. For the discussion on the cost function, it will be assumed that one vehicle block ends with trip i and another vehicle block begins with trip j.

C. COST FUNCTION FOR CONCATENATION OF TRIPS

For the vehicle scheduling problem with interlining constraints, the cost function that is minimized is a modified linear weighted sum of vehicle deadhead time, vehicle waiting time, number of lines traversed in a vehicle block and number of lines/number of trips in a vehicle block. The iterative algorithms require the costs to be tabulated as each new trip j is added to a vehicle block ending with trip i. It is done in the following manner. Assume that there exists a partial vehicle block associated with trip i where trip i is the last trip on the vehicle block and a partial vehicle block with trip j, where trip j is the first trip on the vehicle block. Then, the cost for concatenating the partial vehicles blocks for trip i and trip j is given in Equation (3.2).

$$\begin{aligned} C'(i,j) = {} & A1 \times LC + A2 \times TC + A3 \times DD(EL(i),SL(j)) \\ & + A4 \times \max(ST(j)-ET(i)-DD(EL(i),SL(j))-5\ ,0) \qquad \text{Eqn (3.2)} \end{aligned}$$

where A1 is the weight factor for number lines traversed.
LC is the total number of lines in the concatenated block.
A2 is the weight factor for TC.
TC is the LC time 100 divided by number of trips in the concatenated block.
A3 is the weight factor for deadhead time.
A4 is the weight factor for waiting time.

Equation (3.2) used the four factors for the reasons described below.

If cost was not important, the ideal schedule would have each vehicle block traverse only one line during a day. Such a schedule could be extremely costly since it would require a large number of vehicle blocks and/or drivers and probably create a situation with extreme amounts of deadhead time for going back and forth to the depot or waiting time. This situation will be seen in the computational results. This ideal schedule is probably not practical because of budgeting considerations. Thus, a solution procedure should insure that the average number of lines in each vehicle block will be as small as possible. In this way, when a run cut is administered to each vehicle block, driver costs are reduced. Therefore, in the procedures developed here, a penalty is incurred if a vehicle block traverses more than one line. This penalty increases linearly with the number of different lines the concatenated vehicle block would traverse. This factor has a weight of A1.

If keeping the number of lines traversed in a vehicle block to a minimum is a goal, a weight factor is designed which takes into account the number of lines traversed divided by the number of trips for a vehicle block. If a vehicle block traverses only one line on a given day then the ratio is small unless the vehicle block traverses only one trip. A vehicle block which traverses two or three lines while covering ten trips may be better than a vehicle block that only covers one line with one trip. Since this number is small when compared to the total deadhead and waiting time terms in Equation (3.2), this number is scaled by multiplying by 100. This factor has a weight of A2.

One of the most undesirable features of a vehicle schedule for a mass transit system is excess deadhead time. Deadhead time is costly in terms of driver pay hours and vehicle operating costs. Of course, some deadhead time is an unavoidable cost such as when a vehicle leaves the depot for the first time of the day and when the vehicle returns to the depot at the end of the day. However, one wants to prevent as much as possible the deadhead between the end location of one trip and the start location of the next trip in a vehicle block or the return of the vehicle to the depot during the day. A weight of A3 is assigned to the deadhead time to go from EL(i) to SL(j).

Waiting time is another expense that the transit agency wishes to hold to a minimum since the vehicle is not producing revenue when waiting and the driver has to be paid when the vehicle is idle. However, a little waiting time can be of benefit to the system. A layover of less than five minutes is not considered a penalty. An example is the following. If the waiting time is small, it might be possible to connect two trips with the same line number into a vehicle block or it might be possible to use waiting time to have a driver relieved by another driver. Also, waiting time can be used as a period of time for the driver to have a break or for the schedule to be caught up (if the vehicle is running late). A factor was assigned to waiting time and carried a weight of A4.

IV. HEURISTICS FOR THE VEHICLE SCHEDULING PROBLEM

In this chapter, two basic procedures for creating an initial set of vehicle blocks under interlining constraints are described. These procedures are the "time increment procedure" and the "matching procedure". Also, in this chapter, the block improvement procedure for forming an improved set of vehicle blocks from the initial set of vehicle blocks is given. Additionally, in this chapter, two procedures for concatenating the vehicle blocks into a final vehicle schedule are presented. These procedures (outside of the block improvement procedure) repeatedly concatenate vehicle blocks into larger vehicle blocks or vehicle schedules. A description of how these procedures fit together into a set of algorithms for solving the problem concludes this chapter.

A. INITIAL VEHICLE BLOCK PROCEDURES

1. Time Increment Procedure

In the time increment method, a time interval (t1,t2) is first defined. Each pair of trips, i and j, in the timetable is examined in a specified order in an attempt to find a combination of trips that minimizes the arc cost (Equation (3.2)) and that satisfies the following conditions:

$$ST(j)-ET(i) \leq t2 \qquad \text{Eqn (4.1)}$$

$$ST(j)-ET(i) \geq t1 \qquad \text{Eqn (4.2)}$$

$$ST(j)-ET(i)-DD(EL(i),SL(j)) \geq 0 \qquad \text{Eqn}(4.3)$$

where DD(EL(i),SL(j)) is the deadhead time to go from the end location of trip i to the start location of trip j. At node i, the arc cost C'(i,j) for all trips j which satisfy equations (4.1) to (4.3) is calculated. Trip k is selected to follow trip i on a partial vehicle schedule if

$$C'(i,k) = \min C'(i,j). \qquad \text{Eqn}(4.4)$$

Of course, if trip i is the end trip of one partial vehicle block and trip k is the beginning trip of a second partial vehicle block, this operation concatenates the vehicle block beginning with trip k to the end of the vehicle block ending with trip i, creating a longer vehicle block.

For a trip i, the procedure starts with an interval (t1,t2) where t1=1 and t2=DELTA and examines all trips j which satisfy Equations (4.1), (4.2) and (4.3). Upon completing the examination of all trips j, a new time interval(t1,t2) is formed where t1=t2+1, t2=t1+DELTA-1 and the process is repeated. The initial study done on the MST database by LCDR Mitchell is a special case of the method. The initial study looked at increments of 1 minute, i.e. t2=t1. The first trip j which satisfied Equations (4.1), (4.2) and (4.3) was concatenated with trip i and any other possible trip was ignored. Chapter V will discuss the sensitivity to DELTA.

The basic steps of the time increment procedure is as follows:

Step 1: Input a timetable, let DELTA and VALUE be specified and sort the timetable by the start time of each trip. (DELTA is the length of the time interval and VALUE is the maximum amount of acceptable waiting time or deadhead time. DELTA is less than or equal to VALUE.)

Step 2: Set lower limit for time interval t1=1 and set upper limit for time interval t2=min(DELTA, VALUE). Let i=1.

Step 3: For trip i,

3.a. If trip i is not the last trip in a vehicle block, increment i and go to Step 3 unless i equals the number of trips in the timetable, then go to Step 8.

3.b. Compute the number of different lines traversed on the partial vehicle block associated with trip i. Let j = i + 1.

Step 4: For trip j,

4.a. If trip j not the start of a partial vehicle block, increment j and go to Step 4 unless j greater that the number of trips in the timetable, then go to Step 5.

4.b. If trip j does not satisfy Equations (4.1),(4.2) and (4.3), increment j and go to Step 4 unless j greater than the number of trips in the timetable, then go to Step 5.

4.c. Compute the number of different lines and number of trips that would be traversed if the partial vehicle block up to and including trip i was concatenated with the partial vehicle block commencing with trip j.

4.d. If the total number of lines traversed is greater than maximum number of lines permitted, increment j and go to Step 4 unless j greater that the number of trips in the timetable, then go to Step 5.

4.e. If the deadhead time or the waiting time is greater than the maximum allowed, increment j and go to Step 4 unless j greater that the number of trips in the timetable, then go to Step 5.

4.f. Compute cost of an arc using Equation (3.2).

4.g. If cost is greater than or equal to previous minimum cost, increment j and go to Step 4 unless j greater that the number of trips in the timetable, then go to Step 5.

4.h. Replace previous minimum cost by cost, let j*=j, (j* is the current candidate to concatenate with trip i) increment j and go to Step 4 unless j greater that the number of trips in the timetable, then continue.

Step 5: Let j=i+1. Let t1=t2+1. Let t2=min(t1+DELTA-1, VALUE) and go to step 4 unless t1 is greater than VALUE. Then, continue.

Step 6: If a trip j* found, connect trip i to trip j* on the vehicle block.

Step 7: Increment i and go to step 3.

Step 8: Stop. (Vehicle blocks have been completed.)

2. Matching Procedure

The matching method adapts some of the procedures in [Ref. 4]. Unlike the time increment method which only looks at one trip at a time, the matching algorithm examines more than one trip at a time. The matching procedure requires that the level of each node (trip) be specified. The level of a node is the relative depth of the node in respect to a particular starting node, that is, the maximum number of trips which could precede it in a vehicle block. Let trip i be represented by node i in a network. An arc exists from node i to node j if Equation (3.1) is satisfied and the arc is given a cost of 1. Each node is connected to a supernode s. The level of each node i is defined by the longest path from s to i. Since the network is acyclic, all longest paths can be easily found [Ref. 3].

The algorithm procedes as follows. All trips at level 1 are called partial vehicle blocks. A matching problem with self loops (a node can be matched to itself) is defined where the partial vehicle blocks on level 1 are one set of nodes and the trips on level 2 are a second set of nodes. The arc costs are as defined in Equation (3.2), and the self loop costs (the cost of not concatenating a trip with another trip) are set to a large number. The solution to the matching problem finds the "best" set of partial vehicle blocks for all trips on levels 1 and 2. The partial vehicle blocks on levels 1 and 2 now become a set of nodes and the trips on level 3 become a set of nodes. The arc costs are again defined by Equation (3.2) and the self loop costs are set to a large number. The solution to the matching problem finds the "best" set of partial vehicle blocks on levels 1 through 3. This procedure continues until all levels have been examined.

Some of the nodes may not be matched in the matching problem defined on level k for a variety of reasons. For example, the number of partial vehicle blocks on levels 1 through k-1 may be less than (greater than) the number of trips on level k. If a node (partial vehicle block) is not matched on a lower level, its level is raised and it is included in the matching for the next level. For example, if level 1 had 3 nodes and level 2 had only 2 nodes, then the one unmatched node from level 1 is added to the list of possible candidates for the matching problem from level 2 to level 3. This upgrading of the partial vehicle blocks continues until it is not possible to match this node with any node on the next level because the maximum deadhead or maximum waiting time constraints are violated.

The basic steps for the matching procedure are:

Step 1: Input a timetable and set up a network where each arc has a cost of 1.

Step 2: Find longest path from s to each node i and assign each node the appropriate level.

Step 3: Let all trips on level 1 be called partial vehicle blocks and let i=1.

Step 4: For level k= i+1,

4.a. Find all feasible arcs from the partial vehicle blocks on levels 1,2,..., k-1 to the trips on level k by using Equation (3.1) and find the associated arc cost using Equation (3.2).

4.b. Set self looping cost as a large number.

4.c. Find minimal matching cost solution and update the partial blocks.

Step 5: Increment i and go to step 4 unless all levels have been matched.

Step 6: Stop. (Initial vehicle blocks have been found.)

3. Block Improvement Procedure

The block improvement procedure is based on the concept that the vehicle blocks might be improved in the sense of reducing the number of different lines in a vehicle block. This is accomplished by "freeing up" the ends of each vehicle block and solving a matching problem. The procedure finds the start of a vehicle block and compares the line number for the first two trips. If these trips are from the same line schedule, the second and third trips are compared for the same line number. If these trips are from the same line schedule, the third and fourth trips are similarly compared. If all four trips are from the same line schedule, the vehicle block is removed from consideration for the block improvement procedure.

There are two ways to have a vehicle block become eligible for this procedure; first, if the vehicle block contains less than four trips, or second, the vehicle block has interlining within the first four trips. If a vehicle block has less than four trips and each of the trips are from the same line schedule, the vehicle block is defined as a node in LIST1 which is a partial listing of nodes to be used for a matching problem. If a vehicle block has interlining within the first four trips, the vehicle block is split between the two trips where the interlining occurs. The partial vehicle block up to the interlining is defined as a node in LIST1 and the split off portion is defined as a node in LIST2 which is a partial listing of the second set of nodes for a matching problem.

After all the nodes in LIST1 and LIST2 have been defined, feasible arcs are determined by Equation (3.1) between nodes in LIST1 and LIST2 with arc costs determined by Equation (3.2). Self looping cost are defined as zero for nodes in LIST1 and as a large number for nodes in LIST2. This is to insure that at a minimum of the old vehicle blocks will be redefined as a vehicle block and not create more vehicle blocks than originally started with. A minimum cost matching problem is solved and the vehicles blocks are updated by the solution. As soon as the procedure is complete on the front end of each vehicle, the procedure is applied to the tail end of each vehicle block. For the tails, a total of four trips can be examined, just as with the front of each vehicle block, and the procedure examines the last four trips from the last trip of the vehicle block backwards. Upon completing the examination of the tails, the front portion of the updated vehicle blocks is looked at again. The procedure continues until no changes are found for either end and then stops. To prevent the possibility of an endless loop or an excessive amount of time being taken, the procedure only allows a fixed number of iterations. Chapter V discusses the sensitivity to the number of trips into a vehicle block that the procedure checks for the same line being traversed.

The steps of the block improvement procedure are as follows:

<u>Step 1</u>: Input an initial set of vehicle blocks. Let i=1.

<u>Step 2</u>: For trip i,

<u>2.a</u>. If trip i is not the first trip on a vehicle block, increment i and go to Step 2 unless i equals the number of trips in the timetable, then go to Step 5.

2.b. If trip i is a single trip, place trip i in LIST1, increment i and go to Step 2 unless i equals the number of trips in the timetable, then go to Step 5.

2.c. Let trip j be the successor to trip i and let i*=i.

Step 3: For trip j,

3.a. If trip i* and trip j are not trips from the same line schedule, place trip i* into LIST1 and trip j into LIST2 and go to Step 4.

3.b. If trip j is the last trip in a vehicle block, place trip j into LIST1 and go to Step 4.

3.c. Let trip i* = trip j and let trip j be the successor trip to trip i* and go to Step 3 unless trip i* is the fourth trip in a vehicle block, then continue.

Step 4: Increment i and go to Step 2 unless i equals the number of trips in the timetable.

Step 5: Find all feasible arcs from nodes in LIST1 to nodes in LIST2 by Equation (3.1) and the arc costs by Equation (3.2).

Step 6: Set self looping costs for LIST1 equal to zero and for LIST2 equal to a large number.

Step 7: Find the minimum cost matching solution and update the vehicle blocks.

Step 8: Repeat the procedure for the ends of each vehicle block using the reverse of the procedure, i.e. find the last trip in each vehicle block and find its predecessor, etc.

Step 9: If there was a change in the vehicle blocks after both ends of the vehicle blocks have been through the procedure, let i=1 and go to Step 2.

Step 10: Stop. (Vehicle blocks have been updated.)

E. CONCATENATING BLOCKS FOR VEHICLE SCHEDULES

1. The Greedy Approach

The previous procedures form a set of feasible vehicles blocks. This procedure forms a set of vehicle schedules from the vehicle blocks. This procedure is greedy in that it concatenates the very first feasible vehicle block to the end of the vehicle block being examined. This concatenated vehicle block is termed a longer vehicle block. At the end of this procedure, the set of longer vehicle blocks are the vehicle schedules.

The steps in the greedy approach are as follows:

<u>Step 1</u>: Input a set of initial blocks. Let i=1.

<u>Step 2</u>: For trip i,

<u>2.a</u>. If trip i is not the end of a vehicle block, increment i and go to Step 2 unless i equals the number of trips in the timetable, then go to Step 5.

<u>2.b</u>. Let j=i+1.

<u>Step 3</u>: For trip j,

<u>3.a</u>. If trip j is not the start of a vehicle block or is previously assigned, increment j and go to Step 3 unless j greater than the number of trips in the time table, then go to Step 4.

<u>3.b</u>. If Equation (3.1) is not satisfied, increment j and go to Step 3 unless j greater than the number of trips in the time table, then go to Step 4.

<u>3.c</u>. Concatenate the vehicle block that starts with trip j to the vehicle block that ends with trip i to form a longer vehicle block.

<u>Step 4</u>: Increment i and go to Step 2 unless i equals the number of trips in the timetable.

<u>Step 5</u>: Stop. (Final set of longer vehicle blocks are the final vehicle schedules.)

2. The Matching Approach

The matching approach forms vehicle schedules from vehicles blocks by a repeated solution of a matching problem. The nodes in the matching procedure are the end trips and the beginning trips for each vehicle block. The procedure defines all feasible arcs by Equation (3.1) and their associated costs by Equation (3.2) and solves a minimum cost matching problem.

The steps to the matching approach are as follows:

Step 1: Input an initial set of vehicle blocks.

Step 2: Find the ending nodes of all vehicle blocks and the starting nodes of all vehicle blocks.

Step 3: Find all feasible arcs by Equation (3.1) between the end nodes of each vehicle block and the start nodes of each vehicle block. Determine the arc cost by Equation (3.2).

Step 4: Set self looping cost equal to a large number.

Step 5: Find the minimum cost matching solution.

Step 6: Stop. (All nodes matched in the solution are concatenated to form vehicle schedules.)

C. UTILIZATION OF THE HEURISTICS

The methods to derive vehicle blocks can be efficiently combined with the approaches used to create vehicle schedules. To generate results shown in Chapter V, all of the possible combinations were examined initially and the ones that derived the best solutions were examined more closely. Table II lists the possible combinations with the associated index that will be used to display results in tables in Chapter V. For example, if in examining a table located in Chapter V and the method used was MTD=1, then the vehicle blocks were created by the time increment method and the vehicle schedule was derived by using the greedy approach.

TABLE II

Definitions of Ways of Deriving Vehicle Schedule

METHOD	VEHICLE BLOCKS			VEHICLE SCHEDULE	
MTD	TIME	MATCH	BLOCK	GREEDY	MATCHING
1	YES	NO	NO	YES	NO
2	YES	NO	YES	YES	NO
3	YES	NO	NO	NO	YES
4	YES	NO	YES	NO	YES
5	NO	YES	NO	YES	NO
6	NO	YES	YES	YES	NO
7	NO	YES	NO	NO	YES
8	NO	YES	YES	NO	YES

Procedure for deriving vehicle blocks

Time refers to the Time Increment Method
Match refers to the Matching Method
Block refers to the Block Improvement Procedure

Procedure for deriving vehicle schedule from vehicle blocks

Greedy refers to greedy approach for the concatenating of vehicle blocks
Matching refers to matching approach for the concatenating of vehicle blocks

V. RESULTS

In this chapter, the methods shown in Table II for creating vehicle blocks and schedules were applied to two different databases - the MST database and one database artificially generated - and the results are displayed and evaluated. The procedures are evaluated as to their validity for use in solving the vehicle scheduling problem.

A. MODEL VALIDATION

1. Table Notation

The abbreviations used in the columns of the tables are:

ML: maximum number of lines in a vehicle block.

MTD: heuristic methods to derive the vehicle schedule. The number in the column identifies the method used and corresponds to that row in Table II, e.g., if MTD is 8, the vehicle blocks are derived by the matching procedure and block improvement procedure combined then the matching approach is used to concatenate the vehicle blocks into vehicle schedules.

WT: total waiting time for a vehicle schedule.

DHT: total deadhead time for a vehicle schedule.

NB: number of vehicle blocks required.

L/B: average number of lines traversed per vehicle block.

VS: total number of vehicle schedules.

T/V: average number of trips per vehicle schedule.

2. MST System

The Monterey portion of the MST system is composed of 16 different lines and 240 trips. The timetable utilized was dated effective 20 November, 1982. This timetable was used since it was readily available, not tremendously complex and easily reproduced since the Rider's Guide, **[Ref. 6]**, contained the predecessor and successor trips for each trip. The MST timetable dealt only with the Monday through Friday schedule. This schedule had a large AM peak and a somewhat smaller PM peak. It also had 20 different starting locations excluding the depot. In the MST data, the maximum number of lines a vehicle was found to traverse in a vehicle block was 7. The deadhead time was approximately 1105 minutes, the waiting time was 1417 minutes, the number of vehicle blocks was 35, the number of vehicles scheduled was 26 and the number of lines/vehicle block was 2.94. Table III tabulates this information.

TABLE III

MST System Schedule

ORIGINAL SCHEDULE					
WT	DHT	NB	L/B	VS	T/V
1417	1105	35	2.94	26	9.23

3. Initial Study

The initial study by LCDR Mitchell was carried out as a term project in a seminar class on Routing and Scheduling at the Naval Postgraduate School, October to December, 1982. The approach taken was to generate the vehicle blocks using a time increment method with the increment being 1 minute. When a trip j was found that could be concatenated with a trip i, the trips were joined together to form a partial vehicle block. These partial vehicle blocks did not necessarily have to begin at the depot; the vehicle scheduling procedure took this into account when forming the vehicle schedules. The vehicle scheduling procedure tried to exchange trips between the vehicle blocks to reduce deadhead or waiting times or number of lines traversed. Table IV shows the results for the

TABLE IV

Initial Study Results - MST Database

ML	WT	DHT	NB	L/B	T/V
1	2914	1373	34	1.00	7.06
	2914	1373	34	1.00	7.06
2	1988	1690	51	1.98	4.71
	2782	1235	34	1.85	7.06
3	1125	1848	49	2.91	4.90
	2842	1147	28	2.53	8.57
4	1042	1639	41	3.78	5.85
	2852	1123	28	2.75	8.57
5	1044	1434	37	4.59	6.49
	2823	1123	29	2.69	8.27
7	1113	1291	29	6.10	8.27
	2844	1123	28	2.82	8.57
10	2844	1123	28	2.82	8.57
	1195	1099	26	6.88	9.23
15	2844	1123	28	2.82	8.57
	1193	1099	26	6.85	9.23

initial study. For each maximum line (ML), the table has two rows associated with it: the first row is the initial vehicle blocks and the second row is the number of vehicles required after the concatenating procedure was applied. As will be seen, this method gave inferior results.

TABLE V

Time Increment Results - MST Database

Parameters:

A1= 1 A2= 1 A3= 1 A4= 1 MDHT= 20 MWT= 30

MI	MTD	WT	DHT	NB	L/B	VS	T/V
1	1	2634	1643	55	1.00	29	8.27
1	2	2634	1643	55	1.00	29	8.27
1	3	2634	1643	55	1.00	30	8.00
1	4	2634	1643	55	1.00	30	8.00
2	1	2601	1425	45	1.60	29	8.27
2	2	2594	1425	45	1.48	29	8.27
2	3	2601	1425	45	1.60	29	8.27
2	4	2594	1425	45	1.48	29	8.27
3	1	2604	1349	42	1.90	29	8.27
3	2	2532	1325	42	1.71	31	7.74
3	3	2604	1349	42	1.90	29	8.27
3	4	2532	1325	42	1.71	31	7.74
4	1	2493	1329	41	2.02	29	8.27
4	2	2455	1329	41	1.87	29	8.27
4	3	2493	1329	41	2.02	29	8.27
4	4	2455	1329	41	1.87	29	8.27
5	1	2519	1281	39	2.07	29	8.27
5	2	2501	1269	39	1.92	30	8.00
5	3	2519	1281	39	2.07	29	8.27
5	4	2501	1269	39	1.92	30	8.00
7	1	2519	1281	39	2.07	29	8.27
7	2	2501	1269	39	1.92	30	8.00
7	3	2519	1281	39	2.07	29	8.27
7	4	2501	1269	39	1.92	30	8.00
10	1	2519	1281	39	2.07	29	8.27
10	2	2501	1269	39	1.92	30	8.00
10	3	2519	1281	39	2.07	29	8.27
10	4	2501	1269	39	1.92	30	8.00
15	1	2519	1281	39	2.07	29	8.27
15	2	2501	1269	39	1.92	30	8.00
15	3	2519	1281	39	2.07	29	8.27
15	4	2501	1269	39	1.92	30	8.00

4. Time Increment

For the runs on the MST database, the maximum allowable deadhead time was set equal to twenty minutes and the maximum allowable waiting time was set equal to thirty minutes. Table V displays the results.

In this analysis, with A1= A2= A3= A4= 1 and ML= 1, the results from the methods, MTD = 1, 2, 3 and 4, were the same. This was expected since, with ML= 1, the time increment procedure is basically a first in - first out algorithm. The results differed greatly when ML was changed from 1 to 2 and then to 3. When ML was increased, the total deadhead time and waiting time was decreased. With ML $\geq$ 4, the results stabilized. Other than for ML of 1, the block improvement procedure reduced the average number of lines traversed per vehicle block (L/B) and generally reduced both waiting and deadhead time. The concatenation of the vehicle blocks into vehicle schedules was affected by the method used to derive the initial vehicle blocks. The block improvement procedure generally increased the number of vehicles required but reduced the average number of lines per vehicle block. Since the four methods (MTD= 1, 2, 3 or 4) produced equivalent results, the analysis in the remainder of this paper concentrated on method 4, time increment procedure with block improvement and matching approach for the concatenating of vehicle blocks into vehicle schedules.

The sensitivity of the time increment procedure to the size of the interval (t1,t2) was examined. By holding all the other parameters A1, A2, A3, A4, MDHT and MWT constant, Table VI shows the results of varying DELTA from one minute up to thirty minutes. In general terms, the bigger the DELTA, the smaller the average number of lines traversed in a vehicle block. For minimizing deadhead or waiting time, a value of DELTA between five to ten minutes appear to give the best result. Since one objective of this work was to minimize the number of lines traversed in a vehicle block, a thirty minute DELTA was used, but with the understanding that it does not result in the minimum waiting and deadhead time.

TABLE VI

Time Interval Sensitivity

Parameters:

A1= 1 A2= 1 A3= 1 A4= 1 MDHT= 20 MWT= 30

ML	DELTA	WT	DHT	NB	L/B	VS	T/V
2	1	2486	1452	48	1.77	30	8.00
2	3	2486	1452	48	1.77	30	8.00
2	5	2447	1413	46	1.65	29	8.27
2	10	2447	1413	46	1.65	29	8.27
2	15	2275	1466	48	1.75	29	8.27
2	20	2285	1559	52	1.53	30	8.00
2	25	2517	1425	45	1.53	29	8.27
2	30	2594	1425	45	1.48	29	8.27
3	1	2062	1378	45	2.31	28	8.57
3	3	2062	1378	45	2.31	28	8.57
3	5	2021	1345	43	2.41	27	8.88
3	10	2090	1289	40	2.32	27	8.88
3	15	2056	1403	45	2.17	27	8.88
3	20	2202	1320	41	2.09	28	8.57
3	25	2530	1293	40	1.85	30	8.00
3	30	2532	1325	42	1.71	31	7.74
4	1	2223	1237	39	2.76	27	8.88
4	3	2223	1237	39	2.76	27	8.88
4	5	2095	1129	34	2.85	26	9.23
4	10	2167	1152	36	2.77	28	8.57
4	15	2126	1238	39	2.71	28	8.57
4	20	2254	1250	39	2.58	28	8.57
4	25	2410	1269	39	2.12	30	8.00
4	30	2455	1329	41	1.87	29	8.27

5. Matching

Table VII shows the results of using the matching procedure to generate the initial set of vehicle blocks for the MST timetable (MTD = 5, 6, 7, 8). The results indicate that the matching procedure is independent of whether or not block improvement was used. This result was expected since the matching procedure attempts to find the minimum cost matching solution while forming the initial vehicle blocks. Again, the potential cost of the system increases as the value of ML gets smaller. Since there is no real difference in the method used, method 8, the matching procedure with block improvement for the vehicle blocks and matching

TABLE VII

Matching Results - MST Database

Parameters:

A1= 1 A2= 1 A3= 1 A4= 1 MDHT= 20 MWT= 30

MI	MTD	WT	DHT	NB	L/B	VS	T/V
1	5	2647	1621	53	1.00	29	8.27
1	6	2647	1621	53	1.00	29	8.27
1	7	2647	1621	53	1.00	30	8.00
1	8	2647	1621	53	1.00	30	8.00
2	5	2343	1338	41	1.75	27	8.88
2	6	2361	1326	41	1.75	28	8.57
2	7	2343	1338	41	1.75	27	8.88
2	8	2361	1326	41	1.75	28	8.57
3	5	1946	1193	36	2.22	27	8.88
3	6	1946	1193	36	2.22	27	8.88
3	7	1946	1193	36	2.22	27	8.88
3	8	1946	1193	36	2.22	27	8.88
4	5	1689	1136	33	2.63	26	9.23
4	6	1689	1136	33	2.63	26	9.23
4	7	1689	1136	33	2.63	26	9.23
4	8	1689	1136	33	2.63	26	9.23
5	5	1432	1136	33	2.93	26	9.23
5	6	1432	1136	33	2.93	26	9.23
5	7	1432	1136	33	2.93	26	9.23
5	8	1432	1136	33	2.93	26	9.23
7	5	1421	1136	33	2.87	26	9.23
7	6	1421	1136	33	2.87	26	9.23
7	7	1421	1136	33	2.87	26	9.23
7	8	1421	1136	33	2.87	26	9.23
10	5	1421	1136	33	2.87	26	9.23
10	6	1421	1136	33	2.87	26	9.23
10	7	1421	1136	33	2.87	26	9.23
10	8	1421	1136	33	2.87	26	9.23
15	5	1421	1136	33	2.87	26	9.23
15	6	1421	1136	33	2.87	26	9.23
15	7	1421	1136	33	2.87	26	9.23
15	8	1421	1136	33	2.87	26	9.23

approach for the concatenation of the vehicle blocks into vehicle schedules, was used for the remainder of the analyses on the MST database.

6. Comparison of the Scheduling Methods

Since the Monterey portion of the MST system was constrained to have no more than seven lines per vehicle block, the validation of these procedures was carried out with ML = 7. Table VIII shows the results with the different scheduling methods. The deadhead time for the present MST schedule, for the initial study and for the matching procedure all are very close. Since the number of vehicle blocks for the time increment method is larger than the other methods, the time increment method has more deadhead time because of the larger number of trips to and from the garage. In addition, the time increment method had more waiting time, number of vehicle blocks, and number of vehicle schedules but a smaller average number of lines traversed per vehicle block. The time increment method is higher in the areas discussed but it is felt that by changing the weights for each factor that this approach will produce good solutions. Since the initial method gives inferior results, it will not be considered any further in this thesis.

TABLE VIII

Comparison - MST Database

Method	WT	DHT	NB	L/B	VS	T/V
MST	1417	1105	35	2.94	26	9.23
Initial	2844	1123	29	2.82	28	8.57
Time	2501	1269	39	1.92	30	8.00
Matching	1421	1136	33	2.87	26	9.23

MST refers to the present MST data
Initial refers to the initial study
Time refers to MTD 4
Matching refers to MTD 8

With the same parameters, it appears that the matching method generates a soluton which requires less deadhead time, fewer vehicle blocks, a smaller number of vehicle schedules and more waiting time when compared with the time increment procedure. From these results, one would probably prefer the matching method to the time increment method for generating the initial vehicle blocks for the MST system. The block improvement procedure has little effect on the results.

E. EFFECTS OF CHANGING THE FACTORS WEIGHTS

To show the effects of changing the weight factors, two of the four factors (A1, A2, A3 and A4) were held constant and the other factors varied. This approach was selected since the number of lines per vehicle block and the number of lines divided by the number of trips were related and total deadhead time and waiting time were related. The methods examined are *MTD 4*, the time increment method with block improvement and matching used for concatenating the vehicle blocks into vehicle schedules, and MTD 8, the matching method with block improvement and matching used for concatenating the vehicle blocks into vehicle schedules.

1. Weight Factors A1 and A2

Weight factor A1 is the factor associated with the number of lines traversed in a vehicle block. The weight was considered to be small and used only as a tie breaker when seeking to find the "best" trip j to concatenate with trip i. Weight factor A2 is the factor associated with the number of lines divided by the number of trips in a vehicle block. The results in Table IX demonstrate that for both the time increment and the matching procedures as one increased either A1 or A2, the number of lines/vehicle block

TABLE IX

Weight Factors A1 and A2

Parameters:

A3= 1 A4= 1 MDHT= 20 MWT= 30

ML	MTD	WT	DHT	NB	L/B	VS	T/V	A1	A2
2	4	2594	1425	45	1.48	29	8.27	1	1
2	4	2386	1548	53	1.81	31	7.74	0	0
2	4	2466	1401	46	1.65	30	8.00	1	0
2	4	2587	1413	45	1.48	30	8.00	2	1
2	4	2652	1392	44	1.40	29	8.27	10	1
2	4	2594	1425	45	1.48	29	8.27	0	1
2	4	2667	1377	43	1.44	29	8.27	1	2
2	4	2679	1377	43	1.44	29	8.27	1	10
2	8	2361	1326	41	1.75	28	8.57	1	1
2	8	2000	1530	50	1.94	28	8.57	0	0
2	8	2166	1405	44	1.90	28	8.57	1	0
2	8	2361	1326	41	1.75	28	8.57	2	1
2	8	2482	1289	40	1.75	28	8.57	10	1
2	8	2316	1338	41	1.80	27	8.88	0	1
2	8	2534	1280	40	1.75	28	8.57	1	2
2	8	2491	1329	40	1.70	27	8.88	1	10
3	4	2532	1325	42	1.71	31	7.74	1	1
3	4	1975	1391	45	2.62	28	8.57	0	0
3	4	1976	1321	42	2.35	27	8.88	1	0
3	4	2565	1292	40	1.80	29	8.27	2	1
3	4	2673	1305	40	1.60	29	8.27	10	1
3	4	2528	1353	42	1.76	29	8.27	0	1
3	4	2693	1305	40	1.57	29	8.27	1	2
3	4	2712	1305	40	1.57	29	8.27	1	10
3	8	1946	1193	36	2.22	27	8.88	1	1
3	8	1803	1299	40	2.75	27	8.88	0	0
3	8	1740	1266	38	2.57	27	8.88	1	0
3	8	1933	1206	36	2.22	27	8.88	2	1
3	8	1973	1193	36	2.22	27	8.88	10	1
3	8	1946	1193	36	2.22	27	8.88	0	1
3	8	2145	1245	36	2.22	27	8.88	1	2
3	8	2128	1171	34	2.00	27	8.88	1	10

(L/B) decreases, the total deadhead time and total vehicles required decreases and the total waiting time increases.

2. Weight Factors A3 and A4

The weight factor on deadhead time is A3 and the weight factor on waiting time is A4. The results in Table X indicate that when A3= A4= 0 gives very reasonable results except with respect to total waiting time of the vehicles.

TABLE X

Weight Factors A3 and A4

Parameters:

A1= 1 A2= 1 MDHT= 20 MWT= 30

ML	MTD	WT	DHT	NB	L/B	VS	T/V	A3	A4
2	4	2594	1425	45	1.48	29	8.27	1	1
2	4	2682	1377	43	1.44	29	8.27	0	0
2	4	2700	1386	44	1.43	30	8.00	1	0
2	4	2594	1425	45	1.48	29	8.27	2	1
2	4	2466	1401	46	1.65	30	8.00	15	10
2	4	2594	1425	45	1.48	29	8.27	0	1
2	4	2446	1418	46	1.56	29	8.27	1	2
2	4	2253	1487	49	1.77	29	8.27	10	15
2	8	2361	1326	41	1.75	28	8.57	1	1
2	8	2658	1283	38	1.63	28	8.57	0	0
2	8	2641	1270	38	1.63	28	8.57	1	0
2	8	2359	1313	41	1.73	28	8.57	2	1
2	8	2136	1381	43	1.86	29	8.57	15	10
2	8	2345	1339	41	1.75	28	8.57	0	1
2	8	2241	1364	42	1.80	27	8.88	1	2
2	8	2120	1394	43	1.86	28	8.57	10	15
3	4	2532	1325	42	1.71	31	7.74	1	1
3	4	2715	1305	40	1.57	29	8.27	0	0
3	4	2728	1290	40	1.70	29	8.27	1	0
3	4	2532	1325	42	1.71	31	7.74	2	1
3	4	2021	1345	43	2.41	27	8.88	15	10
3	4	2532	1325	42	1.71	31	7.74	0	1
3	4	2403	1298	41	2.04	29	8.27	1	2
3	4	2032	1345	43	2.39	27	8.88	10	15
3	8	1946	1193	36	2.22	27	8.88	1	1
3	8	2272	1269	37	2.05	27	8.88	0	0
3	8	2283	1256	37	2.05	27	8.88	1	0
3	8	1835	1259	39	2.23	27	8.88	2	1
3	8	1810	1219	38	2.55	27	8.88	15	10
3	8	1931	1206	36	2.25	27	8.88	0	1
3	8	1899	1210	36	2.33	27	8.88	1	2
3	8	1689	1214	37	2.62	27	8.88	10	15

Large values of A3 and A4 give the smallest waiting time but the largest value of lines per vehicle block.

C. EFFECTS ON TRIP DEPTH FOR BLOCK IMPROVEMENT

The purpose of the block improvement procedure is to reduce the number of lines per vehicle block (L/B). The procedure allows the removal of up to four trips from a vehicle block. The question is to determine the proper depth into a vehicle block (the depth of a vehicle block is defined to be the number of trips into a vehicle block)

TABLE XI
Trip Depth Affects

Parameters:

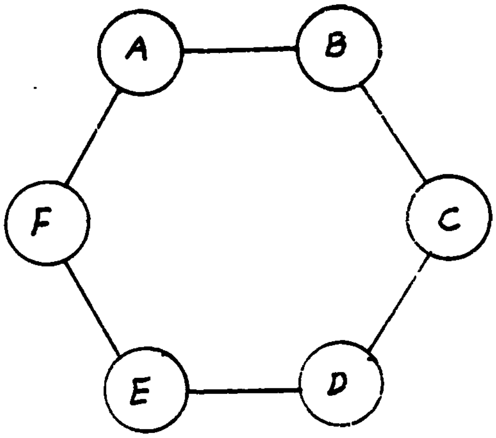
A1= 1 A2= 1 A3= 1 A4= 1 MDHT= 20 MWT= 30

ML	MTD	WT	DHT	NB	L/B	VS	T/V	DEPTH
2	4	2598	1425	45	1.60	29	8.27	2
2	4	2594	1425	45	1.48	29	8.27	4
3	4	2541	1337	42	1.78	30	8.00	2
3	4	2532	1325	42	1.71	31	7.74	4
4	4	2490	1329	41	1.97	29	8.27	2
4	4	2455	1329	41	1.37	29	8.27	4
2	8	2343	1338	41	1.75	27	8.88	2
2	8	2361	1326	41	1.75	28	8.57	4
3	8	1946	1193	36	2.22	27	8.88	2
3	8	1946	1193	36	2.22	27	8.88	4
4	8	1689	1136	33	2.66	26	9.23	2
4	8	1689	1136	33	2.63	26	9.23	4

which can be split apart. Table XI is a comparison of the methods when the depth is changed from two trips to four trips. The column in Table XI that is labeled DEPTH refers to the maximum number of trips that was checked for traversing the same line. The time increment method was somewhat sensitive to the depth of a vehicle block and the greater the depth the smaller was the value of lines per block, waiting time and deadhead time. The matching-based procedure was not affected by this procedure.

D. APPLICATION TO AN ARTIFICIAL DATABASE

To determine the generalities of these results, a second, artifical database was generated. Figure 5.1 shows how a timetable was constructed. The depot was located at node A. To illustrate, the first trip associated with line



Line	Start Time	End Time	Start Location	End Location	Headway
1	7.20	8.15	A	A	9 min
2	7.05	8.10	B	D	17 min
3	7.10	8.00	D	D	13 min
4	7.05	8.20	B	E	6 min
5	7.20	8.15	E	A	18 min
6	7.10	8.45	E	D	14 min
7	7.40	9.00	A	D	15 min
8	7.30	8.20	C	F	21 min
9	7.25	8.25	F	C	17 min
10	7.20	8.20.	C	A	14 min

Deadhead matrix (depot located at A)

	A	B	C	D	E	F
A	0	10	18	18	18	10
B	10	0	10	18	18	18
C	18	10	0	10	18	18
D	18	18	10	0	10	18
E	18	18	18	10	0	10
F	10	18	18	18	10	0

Figure 5.1 Database 2 Route Structure.

1 would begin at 7.00 and have a duration of 75 minutes. The second trip associated with line 1 would begin at 7.09 and have a duration of 75 minutes, etc. The headway is the interval of time between successive trips leaving a specified starting node for a specified line. The period of time that was covered by the timetable was 5 hours and resulted in a timetable which consisted of 230 trips. Table XII shows the initial results for methods 4 and 8. Method 4 again stabilized very early and continued to have a very low number of lines per vehicle block but did "pay" for this in higher waiting time, deadhead time and number of vehicle blocks. In contrast, method 8 (matching) ended with a higher L/B but significantly reduced WT, DHT, NB and VS. These solutions again seem to indicate that the matching-based procedure is preferable to the time increment procedure.

TABLE XII
Database 2 Initial Results

Parameters:

A1= 1 A2= 1 A3= 1 A4= 1 MDHT= 20 MWT= 30

MI	MTD	WT	DHT	NB	L/B	VS	I/V
1	4	923	4018	79	1.00	70	3.28
2	4	1005	3906	72	1.04	70	3.28
3	4	1020	3888	72	1.05	70	3.28
4	4	1020	3888	72	1.05	70	3.28
5	4	1020	3888	72	1.05	70	3.28
1	8	1076	3928	70	1.00	70	3.28
2	8	1057	2870	68	1.57	67	3.43
3	8	862	2274	63	2.19	63	3.65
4	8	868	2168	61	2.34	61	3.77
5	8	868	2168	61	2.34	61	3.77

Varying the weighting factors showed similar results as those obtained from the MST database. Table XIII shows the effects of varying A1 and A2. For method 4, there was a

TABLE XIII

Database 2 - Varying A1 and A2

Parameters:

A3= 1 A4= 1 MDHT= 20 MWT= 30

ML	MTD	WT	DHT	NB	L/B	VS	T/V	A1	A2
2	4	1005	3906	72	1.04	70	3.28	1	1
2	4	851	2968	81	1.91	69	3.33	0	0
2	4	851	2968	81	1.91	69	3.33	1	0
2	4	1005	3906	72	1.04	70	3.28	0	1
2	8	1057	2870	68	1.57	67	3.43	1	1
2	8	1076	2460	68	1.89	68	3.38	0	0
2	8	1079	2470	68	1.82	68	3.38	1	0
2	8	1057	2870	68	1.57	67	3.43	0	1
3	4	1020	3888	72	1.05	70	3.28	1	1
3	4	798	2572	69	2.57	68	3.38	0	0
3	4	791	2496	68	2.54	67	3.43	1	0
3	4	1020	3888	72	1.05	70	3.28	0	1
3	8	862	2274	63	2.19	63	3.65	1	1
3	8	823	2156	64	2.73	64	3.59	0	0
3	8	875	2128	64	2.39	64	3.59	1	0
3	8	849	2328	63	2.12	63	3.65	0	1

significant decrease in WT and DHT and a significant increase in average lines per vehicle block when A2 equalled 0 as compared to when A2 = 1. Also, values of WT, DHT, and L/B appear independent of the weight on A1. A similar set of results was noted for method 8. Again, these results seem to indicate that only A2 needs a weight.

Table XIV shows the results of varying the weights on A3 and A4. Increasing the weight on A3 and A4 decreases the deadhead time but increases the number of lines per vehicle block and number of vehicle schedules and, for the time increment method, increased the waiting time.

For the same parameter settings, the matching procedure clearly dominated the time increment procedure with respect to total deadhead time and number of vehicle schedules and was clearly inferior in terms of lines per vehicle block. For MI = 3, the matching procedure had less waiting time

TABLE XIV

Database 2 - Varying A3 and A4

Parameters:

A1= 1 A2= 1 MDHT= 20 MWT= 30

ML	MTD	WT	DHT	NB	L/B	VS	T/V	A3	A4
2	4	1005	3906	72	1.04	70	3.28	1	1
2	4	985	3952	73	1.04	70	3.28	0	0
2	4	1023	3826	72	1.08	70	3.28	0	1
2	4	1009	3826	73	1.08	70	3.28	1	0
2	4	1194	2722	82	1.85	72	3.19	30	15
2	8	1057	2870	68	1.57	67	3.43	1	1
2	8	1010	2812	65	1.69	65	3.53	0	0
2	8	928	3072	66	1.60	66	3.48	0	1
2	8	1223	2834	67	1.53	66	3.48	1	0
2	8	1218	2364	69	1.84	68	3.38	30	15
3	4	1020	3888	72	1.05	70	3.28	1	1
3	4	1018	3934	73	1.05	70	3.28	0	0
3	4	1008	3782	71	1.14	70	3.28	0	1
3	4	1047	3816	72	1.03	70	3.28	1	0
3	4	1334	2163	70	2.48	70	3.28	30	15
3	8	862	2274	63	2.19	63	3.65	1	1
3	8	927	2252	62	2.19	62	3.70	0	0
3	8	795	2372	62	2.16	62	3.70	0	1
3	8	906	2254	62	2.17	62	3.70	1	0
3	8	857	2080	65	2.29	64	3.59	30	15

than the time increment method but, for ML = 2, no other conclusions could be reached.

VI. CONCLUSIONS AND AREAS FOR FURTHER INVESTIGATION

The procedures developed attempted to minimize the average number of lines traversed in a vehicle block while maintaining a high average number of trips per vehicle schedule. Both procedures used a modified linear weighted sum cost function, Equation (3.2), to derive the costs of concatenating trip i to a trip j. The factors included the number of lines traversed in a vehicle block, the number of lines traversed divided by the number of trips in a vehicle block, the deadhead and waiting times.

A. CONCLUSIONS

These procedures can be applied to obtain a reasonable solution for a vehicle scheduling problem with interlining constraints. The matching-based procedure consistently obtained a better solution in terms of waiting and deadhead times, number of vehicle blocks and average number of trips per vehicle schedule. The time increment procedure did result in a lower average number of lines traversed per vehicle block. However, it is possible to increase the value of A2 so as to have the matching-based procedure produce results comparable to the time increment procedure. For example, when A2 equaled 10, the results for both methods on the MST data set were nearly identical for L/B. The CPU time for both of these procedures averaged less than 5 seconds on an IBM 3033 so computational time was not considered a factor in the comparison of the two methods.

Based on the assumptions and the results of this study, it is possible to find reasonable solutions to the vehicle scheduling problem with interlining constraints. One can vary the solution by increasing or decreasing the values associated with each factor and some general guidelines can be established. A1, the factor for the number of lines traversed is not a dominant factor but it can be useful to break ties. A2, the factor associated with the number of lines traversed divided by the number of trips, can dominate the solution and, by increasing or decreasing its weight, a desired value can be found. The use of the factor A3, for deadhead time, is obvious as is A4, the factor for waiting time. By varying A3 and A4, solutions can be changed dramatically but it is obvious that the deadhead time factor should have a weight greater than the waiting time factor weight. Constraining the maximum number of lines traversed in a vehicle block increases the costs of the system in terms of deadhead time, waiting time and number of vehicles needed. For the time increment methods, the block improvement procedure can reduce the average number of lines traversed in a vehicle block.

B. AREAS FOR FURTHER INVESTIGATION

This study used a modified linear function to develop the cost of an arc based on the number of lines traversed, the number of lines traversed divided by the number of trips, deadhead and waiting times. The procedures did not attempt to find the optimal values for the weights of each factor, rather, examined to see if this type of approach could lead to feasible solutions. The true sensitivity to each of the factors has not been fully determined, instead, guidelines were given as to whether or not a factor should have a non-zero weight. An investigation into finding out if a ratio of one weight to another might lead to better solutions, e.g., should the weight for deadhead time to waiting time be 2:1, 3:1, or what?

Another consideration is the use of the modified linear function. The objective function might be better represented by a non-linear form. An example is the factor concerning waiting time. Some waiting time could be considered an asset so as to catch the schedule up if the vehicle falls behind schedule or to allow for driver relief, etc. The modified linear function assumed a value of 0 for the waiting time if less than a given value (for the study it was 5 minutes). It does not necessarily seem logical that waiting time as a cost would be linear after that time since it would become more of a cost and the relationship to the cost might not be linear. Similar logic could be applied to deadhead time. Some deadhead time is a fixed cost but the deadhead time from one location to another might be beneficial somewhere down the schedule since it might make a valid connection which could end up saving more than the one cost. One approach to answer this might be to derive a tentative vehicle block then apply a savings type algorithm to it. This type of approach could take the place of or be in addition to the block improvement procedure. Another possible solution might be to derive a strictly non-linear function.

The procedures looked at dealt only with vehicle scheduling and ignored the driver scheduling problem. If the problem is to be solved using a simultaneous method for both the driver/vehicle scheduling, how can these procedures be applied? One could add more constraints so as to satisfy the driver scheduling problem but will it still lead to a reasonable solution? Question in this area of driver/vehicle scheduling can lead into other areas for further research.

The procedures developed were applied only to two small timetables. The application to a larger timetable and the resultant solution should be investigated to get a better feel for the values of the weights for the different factors. The effect of constraints on how long a vehicle can preform a schedule should be investigated, along with the effect of heterogeneous vehicles and/or multiple depots.

Beyond the questions of deriving vehicle blocks is the question of concatenating the blocks into feasible and reasonable vehicle schedules. The procedures used in this study were fairly simple. Concatenating the vehicle blocks could be approached using a savings type approach which could interchange trips from one vehicle block to another vehicle block in order to reduce some desired factor. Additional approaches are as many as there are people to derive them.

C. COMMENTS

As any individual who has attempted to solve a vehicle scheduling or a driver scheduling problem can attest, there may not be a common answer as to the best solution or to how to derive a reasonable solution. This study has shown that reasonable solutions to the vehicle scheduling problem with interlining constraints can be obtained using the two procedures described. The matching-based procedure does produce better solutions and can give comparable solutions to a schedule that is in existence for a mass transit system.

LIST OF REFERENCES

1. Bennington, G. and Rebibo, K., "Overview of RUCUS Vehicle Scheduling Program (BLOCKS)," in *Preprints: Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services*, D. Bergmann and L. Bodin, eds. 1975.

2. Wren, A., ed. *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, North-Holland Publishing Co. Amsterdam, 1981.

3. Bodin, L.D., Golden, B, Assad, A. and Ball, M.O., "The State of the Art in the Routing and Scheduling of Vehicles and Crews", to appear in *Computers and Operations Research*, June, 1983.

4. Ball, M.O., Bodin, L.D., and Dial, R., "A Matching Based Heuristic for Scheduling Mass Transit Crews and Vehicles", *Transportation Science*, Volume 17, No. 1, 1983

5. Ford, L. and Fulkerson, D., *Flows in Networks*, Princeton University Press, Princeton, New Jersey, 1962

6. "Rider's Guide to the MST System", Published by the Monterey-Salinas Transit system, effective 20 November, 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, Ca. 93940	2
3.	LCDR M.L. Mitchell 326 Parson Circle Marina, Ca 93933	1