1983

The Multi-Vehicle Subscriber Dial-A-Ride Problem

Bodin, Lawrence D.; Sexton, Thomas R.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/63226

NPS55-83-002

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THE MULTI-VEHICLE SUBSCRIBER

DIAL-A-RIDE PROBLEM

by

Lawrence D. Bodin

Thomas R. Sexton

February 1983

Prepared for:

Naval Postgraduate School
Monterey, CA 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral J. J. Ekelund
Superintendent

David A. Schrady
Provost

Lawrence D. Bodin
University of Maryland
College Park, Maryland
Adjunct Professor
Naval Postgraduate School
Monterey, CA 93940

Thomas R. Sexton
State University of New York
Stony Brook, N. Y.

Reviewed by:

K. T. Marshall, Chairman
Department of Operations Research

Released by:

William M. Tolles
Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS55-83-002	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE MULTI-VEHICLE SUBSCRIBER DIAL-A-RIDE PROBLEM		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lawrence D. Bodin Thomas R. Sexton		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		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)

Dial-A-Ride
Routing
Scheduling
Heuristic
Transportation

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

An algorithm is developed for solving the multi-vehicle subscriber dial-a-ride problem. This algorithm is tested on a set of data from Project Mobility in Baltimore, Maryland. Results of these tests are reported along with recommendations for future work in this area.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

An algorithm is developed for solving the multi-vehicle subscriber dial-a-ride problem. This algorithm is tested on a set of data from Project Mobility in Baltimore, Maryland. Results of these tests are reported along with recommendations for future work in this area.

1.0 Introduction

Subscriber dial-a-ride systems have been set up in many localities to transport their target populations, such as the elderly and handicapped, between their homes and special locations (hospitals, clinics, work, etc.). It is a highly subsidized program; the revenues from the program cover but a small percentage of the cost of the operation.

Each request for service in such a system has a unique pickup point, a unique delivery point, and generally either a desired time of pickup or a desired time of delivery. From a routing and scheduling standpoint, each request for service has a precedence relationship (the pickup must precede the delivery) and a one-sided time window (the delivery of the customer cannot be late if a desired delivery time is specified or the pickup of the customer cannot be early if a desired pickup time is given). It has been found that the development of effective vehicle routes and schedules in this type of situation is a complex enterprise (Bodin [1]). In Baltimore, Maryland, for example, it will take an experienced scheduler three to four hours to develop routes and schedules manually for an 85 customer subscriber service (and many of these requests for services are the same from one day to the next). Conventional automated routing and scheduling procedures cannot be applied because of the special structure of the problem. Therefore, innovative approaches to this problem are necessary. Such a procedure is presented in this paper.

This study had two major objectives. The first objective was to develop a routing and scheduling procedure for the vehicles in a subscriber dial-a-ride fleet. The second objective was to test this procedure on an actual data base. As is demonstrated in this paper, both objectives were accomplished.

In the next section, the multi-vehicle subscriber dial-a-ride algorithm is described. The core single vehicle algorithm used in this procedure is given in [10]. A special case of this algorithm was tested extensively on actual data obtained from Project Mobility in Baltimore, Maryland. These test results are given in Section 3. In Section 4, conclusions and recommendations are presented.

Other work in the dial-a-ride area includes the papers by Stein [13]-[15], Psaraftis [5]-[8] and Wilson [16]-[18]. In [1], a survey of the state-of-the-art in dial-a-ride algorithms is given.

2.0 The Multi-Vehicle Subscriber Dial-a-Ride Algorithm

We are presented with a set of m customers. Each customer has an origin node $o(i)$, a destination node $d(i)$ and a desired time of delivery τ_i. A fleet size V is also specified. The purpose of the multi-vehicle subscriber dial-a-ride algorithm is to form a set of V routes and schedules, one for each vehicle, which services all the customers and which minimizes total customer inconvenience Z. Z is defined as follows:

$$Z = \sum_{i=1}^{m} ZZ(i) \tag{1}$$

where ZZ(i) is the customer inconvenience for customer i. ZZ(i) is defined in the following manner:

$$ZZ(i) = A*DTD(i) + B*ERT(i) \quad (2)$$

where

° DTD(i) is the delivery time deviation for customer i. If y_i is the computed delivery time for customer i and τ_i is the desired delivery time for customer i, then

$$DTD(i) = \tau_i - y_i \quad (3)$$

and it is assumed that DTD(i) ≥ 0.

° ERT(i) is the excess ride time for customer i. If y_i is the computed delivery time for customer i, x_i is the computer pickup time for customer i and $d_{01}(i,i)$ is the minimum ride time from pickup location o(i) to delivery location d(i), then

$$ERT(i) = y_i - x_i - d_{01}(i,i) \quad (4)$$

where it is assumed that ERT(i) ≥ 0.

° A is the penalty for one unit of delivery time deviation.

° B is the penalty for one unit of excess ride time. It must be noted that we can always set B = 1 and interpret A as the ratio of the penalty for one unit of delivery time deviation to one unit of excess ride time as long as B > 0.

The constraints on the problem are the following:

(i) Each customer must be picked up and delivered on the same vehicle.

(ii) Each customer must be delivered no later than his desired delivery time. This condition was quantified in (3).

(iii) Each customer can be picked up no later than his latest possible pickup time ρ_i, $\rho_i = \tau_i - d_{00}(i,i)$.

(iv) The travel time from the pickup location to the delivery location for each customer must be no smaller than the minimum time it takes to travel from the pickup location to the delivery location of the customer. This condition was quantified in (4).

(v) Vehicle capacity cannot be exceeded.

(vi) Let $S(k) = \{i | \text{customer } i \text{ is assigned to vehicle } k\}$, $k = 1,2,\ldots,V$. Assume that $S(k) \cap S(\ell) = \phi$, the empty set, for $k \neq \ell$, $k = 1,2,\ldots,V; \ell = 1,2,\ldots,V$. A necessary constraint on a schedule for any vehicle is that the time between adjacent tasks on a route must be at least equal to the time it takes to travel between the locations at which these tasks are to be carried out. A task is either the pickup or delivery of a customer.

When all customers are assumed to be on one route, this problem becomes the single vehicle many to many routing and scheduling problem (SVRS) which was analyzed in great detail in Sexton [9] and Sexton and Bodin [10]. The SVRS is a nonlinear 0-1 mixed integer programming problem. Two algorithms were developed for the SVRS--one based on Benders decomposition and the other on a nearest neighbor rule with perturbed "distance" matrix. The description of the Benders algorithm can be found in Sexton and Bodin [10]. A description of the other algorithm, which we call the space-time heuristic, is given later in this section.

An algorithm for solving the many vehicle subscriber dial-a-ride problem can be outlined as follows:

Step 1: Form an initial set of vehicle clusters, i.e., partition the customers into sets $S(1), S(2), \ldots, S(V)$.

Step 2: Form a route and schedule for each vehicle cluster using the Benders procedure and the space-time heuristic. Determine total customer inconvenience for each route and schedule. Go to Step 3.

Step 3: Attempt to reassign customers. When no more improvements can be found, go to Step 4. When a customer can be reassigned, perform the reassignment forming new clusters $S(1), \ldots, S(V)$. Continue Step 3.

Step 4: Form final routes and schedules for all vehicles using the Benders procedure. Print out results. Stop.

For a given A and B, we found that the final set of routes had about the same value of Z no matter what the initial clusters were. What we noticed, however, was that the swapper algorithm was relatively slow computationally so that this procedure would run into computational problems when the number of customers became large. We also observed that if the initial set of vehicle clusters had a value of Z close to the value of Z in the final solution, the swapper required less work and was faster computationally. What we desire, therefore, is a better way to do Step 1; that is to say, a clustering procedure which gives a set of vehicle clusters whose Z value will be close to the optional solution. Most of our recent efforts have gone into the analysis of this problem. This work is as yet incomplete and will be reported on in a subsequent paper. Moreover, as described in [1], other groups studying the subscriber dial-a-ride problem have been concentrating their efforts in this area. As will be seen, the above procedure does give reasonable results for modest size problems.

We now describe the swapper algorithm and the space-time heuristic. These procedures along with the Benders procedure (which was described in [9] and [10]) make up the computational methods used in our algorithm.

2.1 <u>The Swapper Algorithm</u>

The swapper algorithm attempts to move customers among the specified vehicle clusters $S(k)$, $k = 1,\ldots,V$, in order to find a final set of vehicle clusters with reduced customer inconvenience. Input to the swapper are the number of vehicles, V, the customers in each vehicle cluster $S(k)$ and the travel time between each pair of distinct origin and destination locations.

In the swapper, we repeatedly pass through the customer list. On each pass we attempt to find customers with a ZZ(i) larger than some predetermined number (which is a function of the pass we are on) and move those customers, one at a time, to another vehicle cluster. In order to make the algorithm more efficient on the first pass, we are quite selective in determining which customers to swap, only attempting to swap a customer C_i with very bad service (i.e. a customer with a large value of ZZ(i)). On subsequent passes, we attempt to swap customers with better service since the routes and schedules have improved and most customers receive better service.

A customer C_i can be moved from vehicle cluster S(L) to vehicle cluster S(J) if the total value of customer inconvenience for clusters S(L) and S(J) before the swap $(C_i \varepsilon S(L))$ is greater than the total value of customer inconvenience for clusters S(L) and S(J) after the swap $(C_i \varepsilon S(J))$. To evaluate this swap requires the determination and evaluation of the following four routes and schedules:

(a) S(L) with $C_i \varepsilon S(L)$.

(b) S(J) with $C_i \not\varepsilon S(J)$.

(c) S(L) with $C_i \not\varepsilon S(L)$.

(d) S(J) with $C_i \varepsilon S(J)$.

(a) and (b) denote the vehicle clusters before the swap; (c) and (d) designate the vehicle clusters after the swap. Because of the computational time required of the Benders procedure, the space time heuristic was used in the analysis.

The customer $C_i \varepsilon S(L)$ can be moved to any other vehicle cluster $S(J)$, $J = 1,2,\ldots,V$ and $J \neq L$. The vehicle cluster $S(M)$ selected to receive customer C_i in the swap is that vehicle cluster offering the maximum decrease in Z over all possible vehicle clusters other than $S(L)$. If such a change is affected,

$$S(M) = S(M) \cup \{C_i\} \tag{5}$$

$$S(L) = S(L) - \{C_i\} . \tag{6}$$

To save computation time, we do not attempt to swap C_i to J from L if the value of customer inconvenience for vehicle cluster $S(J)$ with $C_i \not\varepsilon S(J)$ is large when compared to the value of customer inconvenience for vehicle cluster $S(L)$ with $C_i \varepsilon S(L)$. This check improved the efficiency of the swapper by about 30% with no degradation in results.

We have not performed any worse case analysis of the swapper algorithm and our average case analysis is restricted to comparing our results with the Mobility results (as will be reported on in Section 3). It takes about 2-3 minutes of CPU time on a Univac 1108 to solve the 85 customer problem. A reasonable rule of thumb is that the computation time is dependent on the number of attempted swaps and the number of customers in each vehicle cluster. An upper bound on the number of attempted swaps is the product of the number of passes through the customer list, the number of customers, and the number of vehicles - 1. Because of the number of checks we have placed in the swapper procedure, the number of attempted swaps is usually significantly less than this upper bound. The key element in terms of the amount of storage required by this procedure is the space needed to store the travel time matrices. Virtually all of the other storage requirements are vectors the size of the customer list. We were able to solve the 85 customer problem (with a 106×106 travel time matrix) in under 65K words on the Univac 1108.

2.2 The Space-Time Heuristic

As noted previously, a task is defined as the pickup or delivery of a customer, ρ_i is the latest possible pickup time for customer i, τ_i is the desired delivery time for customer i and $\rho_i = \tau_i - d_{00}(i,j)$. Given a vehicle cluster $S(K)$, the space time heuristic forms a route (a sequence of tasks) for all customers in $S(K)$. The scheduling procedure described in [10] in then invoked to derive a schedule for this route. The route is derived using a "nearest neighbor" type of rule with parameters called "space time separations" as the criterion function. The space time separations $\sigma_{00}(i,j)$, $\sigma_{01}(i,j)$, $\sigma_{10}(i,j)$ and $\sigma_{11}(i,j)$ are defined as follows.

$$\sigma_{00}(i,j) = d_{00}(i,j) + \rho_j - \rho_i \tag{7}$$

$$\sigma_{01}(i,j) = d_{01}(i,j) + \tau_j - \rho_i \tag{8}$$

$$\sigma_{10}(i,j) = d_{10}(i,j) + \rho_j - \tau_i \tag{9}$$

$$\sigma_{11}(i,j) = d_{11}(i,j) + \tau_j - \tau_i \ . \tag{10}$$

A space-time separation measures 1) the travel time between the two locations at which the tasks are being performed (their spatial separation), and 2) the difference between the latest feasible times at which the tasks can be performed (second task minus the first task) which we call the temporal separation. The space-time separation can be negative, zero or positive. A large positive space-time separation between task α and task β implies that task α followed by task β on a route is probably a poor sequence since either the travel time between these two points is large or the latest time to carry out task β is significantly greater than the latest time to carry out task α. On the other hand, if two task locations are nearby and these tasks are to be handled at about the same time, then task α followed by task β on a route is probably a good sequence. In this case, the spare time separations should be small (either positive or negative). A negative space time separation means that the latest time to carry out task β is significantly less than the latest time to carry out task α since the travel times are positive. If a spare time separation is a large negative number, then to have task β following task α on the route will imply that task α will have a large delivery time deviation. In this situation, task β will probably be assigned to follow task α on the route but the route will probably not be too efficient.

The space-time heuristic algorithm begins by placing the pickup of the customer with the smallest feasible pickup time, <u>i.e.</u>, $\min_i (\rho_i)$, as the first task to be serviced on the route. The routing procedure then sequentially constructs a route based on the set of space-time separations from the current last task on the route to all other feasible immediate successor tasks. Specifically, the set of tasks which are feasible immediate successors to the current task i are the following: (1) the pickup of customer j for all customers not yet picked up, and (2) the delivery of customer j for all customers currently on the vehicle. We than calculate

$\lambda_{00} = \min(\infty, \sigma_{00}(i,j))$, over all customers j such that customer j is not yet picked up; if current last task is <u>pickup</u> of customer i, (11)

$\lambda_{10} = \min(\infty, \sigma_{10}(i,j))$, over all customers j such that customer j is not yet picked up; if current last task is <u>delivery</u> of customer i, (12)

$\lambda_{01} = \min(\infty, \sigma_{01}(i,j))$, over all customers j where j is on the vehicle; if current last task is <u>pickup</u> of customer i, (13)

$\lambda_{11} = \min(\infty, \sigma_{11}(i,j))$, over all customers j where j is on the vehicle; if current last task is <u>delivery</u> of customer i, (14)

$$\lambda = \mathrm{Min}(\lambda_{00}, \lambda_{01}, \lambda_{10}, \lambda_{11}) \tag{15}$$

If $\lambda = \infty$, then the route is completed. If λ is finite, then we select as the candidate next task on the route that task which generates the minimum value of λ. We call this task the "next task on the route."

Before we accept this task as the "next task on the route," we check to see if the vehicle will pass "close" to either the pickup location or the delivery location of some customer on the vehicle. (Each customer on the vehicle has been picked up but not delivered.) This is called the DETOUR check. The purpose of this check is to guarantee that no customer essentially "rides through" his origin or destination location thus creating wasteful loops in the vehicle schedule. If, in going from the current task to the "next task on the route," the vehicle passes "close" to the delivery location of a customer on the vehicle, then the delivery of that customer becomes the "next task on the route." If the vehicle passes "close" to the pickup location of a customer on this vehicle and if the pickup of that customer has been moved forward by the DETOUR computations less than $2|S(K)|$ times, then the pickup task of the customer is moved from its earlier position in the task sequence to become the "next task on the route." (The condition on the maximum number of times the task is switched is necessary to guarantee finiteness of the algorithm.) If several tasks satisfy the above conditions, then the task closest in travel time to the current task is selected as the "next task on the route" (a tie in this case being broken arbitrarily).

We say that location c is <u>intermediate</u> between locations a and b if and only if $c \neq a$, $c \neq b$ and $TT(a,c) + TT(c,b) - TT(a,b) \leq$ DETOUR. Here $TT(a,b)$ is the travel time from location a to location b and DETOUR is a non-negative constant which represents the maximum amount of additional travel time we are willing to allow to prevent the "ride-throughs" of the customers.

3.0 Computational Results

Project Mobility in Baltimore, Maryland is a subscriber dial-a-ride system. It is set up to transport the elderly and handicapped between their homes and special locations such as hospitals, clinics, the train station and so forth, in the Baltimore area. The program is highly subsidized with the passengers paying a small percentage of the cost of the trip.

Each request for service must be filed at least 24 hours in advance. Thus, a person requesting service on Wednesday morning must call this request in by the close of business on Monday. In the morning, most passengers request that service be provided from their home to a desired location and that their delivery to this location be completed by a specified time. Most afternoon requests for service are "mirror-images" of the morning's requests for service. In the afternoon, most passengers request that service be provided to their homes from a specified location and that the pickup from this specified location be carried out no earlier than a specified time.

We received one day's data from Project Mobility. Although there were about 170 customers in the data base, we were able to partition the data into two smaller data bases (morning and afternoon) consisting of about 85 customers each. With each customer was an associated time. We could not successfully interpret the times associated with each customer in the morning data base but were able to interpret the time associated with each customer in the afternoon data base as desired time of pickup. As such, the afternoon data was chosen as the data set for this analysis.

Since the space-time heuristic algorithm and the Benders procedure used desired delivery times only, we subtracted each customer's desired pickup time from 1440 (number of minutes in a day), made this new time a desired delivery time and reversed the pickup and delivery locations. Each route, therefore, was formed from end to beginning and then reversed (when printed) to come out with routes that properly represented the data.

Each customer was assumed to have a dwell time of three minutes for pickup and delivery. Dwell time for a customer is the time it takes a customer to get on or off the vehicle at a stop. In an actual implementation, dwell time can be quite variable depending upon whether the customer is in a wheelchair, handicapped but not in a wheelchair, or not handicapped at all. In [12], we extend the Benders procedure to handle variable dwell time.

There were 106 unique locations at which the pickups and deliveries of the 85 customers took place. The (x,y) coordinates for these locations were extracted from the Baltimore Regional Planning Council Summary Address Coding Guide. The travel speed in the central core of Baltimore was assumed to be 8 mph while the travel speed in the remainder of Baltimore was assumed to be 16 mph. The shape of the central core was assumed to be a rectangle. A subroutine was written to compute the travel speed from any point in the Baltimore area to any other point. This travel speed was based on the Euclidean distance between these two points but took into account any encounter this Euclidean distance had with the rectangle. We were given seven vehicle clusters $S(1),\ldots,S(7)$ generated by the Project Mobility schedulers. We manually evaluated these seven vehicle clusters with respect to total pickup time deviation and excess ride time. This evaluation is given in Table 1. We then generated a set of vehicle routes using a fairly simple clustering routine. In each of these cases the resulting seven vehicle clusters were run through the algorithm described in Section 2. The results of this analysis is displayed in Table 2. Each of our solutions, when run through the swapper algorithm, clearly dominated the Mobility routes regardless of the choice of A and B $(A \geq B)$. The initial solution, however, did not dominate the Mobility routes and the swapper was quite dynamic in improving the results. Moreover, we have run many other cases where $A \geq B$. In every case, our routes clearly dominated the Mobility routes.

An operational dial-a-ride system can be composed of two components, a subscriber fleet and a demand responsive fleet, and it is possible to deny a customer service in the subscriber fleet and handle him in either the demand responsive fleet or by a taxi. The problem was to identify customers who should be considered as candidates for being denited service in the subscriber fleet. We defined the "marginal value" of denying customer C_i service as follows:

$MV(C_i)$ = Total customer inconvenience for route with customer C_i on route - total customer inconvenience for route with customer C_i off route.

For each route, we selected that customer to deny service to as that customer which had the maximum marginal value over all

TABLE 1

Analysis of the Mobility Routes

	Pickup Time Deviation	Excess Ride Time
Vehicle 1	121	309
Vehicle 2	0	136
Vehicle 3	17	59
Vehicle 4	477	1,143
Vehicle 5	382	119
Vehicle 6	25	244
Vehicle 7	66	86
Total	1,088	2,096

TABLE 2

Comparison of Mobility Routes with Routes

Generated by the Algorithm

	Mobility Routes	A = 9 B = 1	A = 2 B = 1	A = 5 B = 4
Delivery Time Deviation for Initial Solution	N/A	495	659	1,130
Excess Ride Time for Initial Solution	N/A	2,660	1,065	1,053
Delivery Time Deviation for Final Solution	1,088	213	427	719
Excess Ride Time for for Final Solution	2,096	2,064	1,003	900

customers on the route. We then ran each route with C_i through the Benders algorithm to find the total customer inconvenience with C_i off the route. The total customer inconvenience for the route with C_i on it had already been run through the Benders procedure.

For the case in Table 2, where $A = 9$ and $B = 1$, removing the customer with maximum customer inconvenience from each of the seven routes gave the following results (these results are presented in Table 3).

- The number of customers denied service is 8.2% (7,185).
- Total Delivery Time Deviation is reduced by 26.7% ((213-156)/213).
- Total Excess Ride Time is reduced by 41.6% ((2064-1206)/2064).

We also ran the marginal procedure for the case where $A = 5$ and $B = 4$. The reduction in service level of 8.2% reduced total delivery time deviation and excess ride time by at least 25%. (The detailed analysis of this case is not presented.) The analysis showed that the criteria of marginal value can be very effective in identifying customers whose service characteristics adversely affect the service provided the other customers on the route. In many cases, the customer C_i denied service was not the customer who had the maximum value of customer inconvenience, but had a very small value of customer inconvenience. This was due to the fact that a customer can receive very good service but be so positioned on the route that he adversely affects the service given many other customers on the route.

TABLE 3

Marginal Analysis of the Mobility Routes

when A = 9 and B = 1

Vehicle Number	1	2	3	4	5	6	7
Marginal Value	309	185	165	175	121	180	245
Customer Number C_i with maximum marginal value	44	29	22	25	9	59	49
Total Delivery Time Deviation when all customers are on the route	87	35	12	34	3	18	24
Total Delivery Time Deviation when C_i is removed from route	66	23	11	25	3	13	15
Total Excess Ride time when all customers are on route	436	183	339	327	148	386	245
Total Excess Ride time when C_i is removed from route	316	106	182	235	37	251	79

We also ran the algorithm on the 85 customer data base for six vehicles and five vehicles. The start clusters for the six vehicle case was the same as the clusters used by the Mobility staff except that two of the smaller start clusters were merged into one (in this way, we derived six start clusters rather than seven). In the five vehicle case, we used the same start clusters as in the six vehicle case except that we merged two sets of two vehicle clusters into one. We ran the algorithm with $A = 9$ and $B = 1$. For six vehicles, total customer inconvenience was 7,915 and for five vehicles, total customer inconvenience was 12,454. This compares to the total customer inconvenience of 11,888 found for the seven vehicle Project Mobility routes and schedules. Hence, with the algorithm described in this paper, we are able to service all the Project Mobility customers with one less vehicle and less total customer inconvenience with two less vehicles (five rather than seven) and with an increase in customer inconvenience of less than 5% $\left(\frac{12,454 - 11,888}{11,888} = \frac{566}{11,888} < 5\%\right)$. Hence, using our algorithm, it was possible on this set of data to reduce customer inconvenience by about 40% and the capital cost of carrying out this service by about 14%. Furthermore, it was possible to reduce the capital expenditure by about 28% at an increase in total customer inconvenience of only 5%. It must be noted that the algorithm described in Section 2 was modified slightly to derive the last set of results (i.e. the set of results when there were five and six vehicles). The Benders algorithm for the single vehicle dial-a-ride routing and scheduling system was under development at the same time as this study was being undertaken. As such, the results noted were based on an evaluation of the routes and schedules at the end of Step 3. These results can only be improved upon when run through the Beners' procedure, since the Benders' algorithm would utilize the final solution found in Step 3 as a starting solution.

4.0 Conclusions

We believe that the algorithm described in Section 2 can be utilized as a base for scheduling customers in a subscriber dial-a-ride environment. In a letter from Robert Stark of the State of Maryland Department of Transportation to Professor Bodin, Mr. Stark states that "your automated solution was significantly better than our manual solution for the same data base." Mr. Stark goes on to state:

> The single significant benefit in your series of programs is dramatically pointed out in that proper vehicle utilization can greatly diminish patron travel time. We see this as a significant break-through in dealing with handicapped individuals.

This fact can be witnessed in Table 2 since each of the solutions generated by the algorithms had less excess ride time than the Mobility routes.

This dial-a-ride algorithm was not implemented in Baltimore because of the cost that would have been encountered in converting and imbedding our code into a "user friendly" environment. This algorithm has been enhanced and is currently being embedded within the CARDS/MIS system in Dade County, Florida. The CARDS/MIS project has been going on for about two years at a cost of several hundred thousand dollars. Final implementation of our enhanced routing and scheduling system is currently being carried and system evaluation is due to begin shortly. Documentation of our procedure will be described in a subsequent paper and preliminary results were presented in [3].

As noted previously, subscriber dial-a-ride service is a highly subsidized operation. To have the facility to be able to determine the minimum number of vehicles which can service the customers at a satisfactory level can have a significant impact on the cost of the system (and, hence, the cost to the taxpayer). The latter two topics discussed in Section 3.0 (the marginal analysis and the reduction in fleet size) can become key issues in any implementable system since these considerations will allow the operators of the system to produce a set of routes and schedules which provide satisfactory service at a lower cost.

Some of the features being incorporated or being considered for incorporation into the CARDS/MIS system in Dade County, Florida are the following:

1. The development of routes and schedules with customer-dependent dwell times and customer-dependent penalties. As noted above, customer-dependent dwell times will allow us to make the procedure sensitive to customers who are in a wheelchair or handicapped but not in a wheelchair. Customer dependent penalties means that A is replaced by $A(i)$ and B is replaced by $B(i)$. This extension will allow us to penalize customers differently depending upon

whether they have to get to the destination by a certain time (for a class, train, doctor's appointment, etc.) or not (customers going to a shopping center, etc.). This analysis has been completed and was reported on in [12].

2. A robust procedure to perform the initial breakdown of customers into vehicle clusters is needed. Research into this area is currently going on. Preliminary results in this area was reported on in [3].
3. A fundamental assumption is that every customer had a desired delivery time or a desired pickup time. Of course, there are problems when some customers will have desired pickup times while others will have desired delivery times. This problem appears quite difficult to solve.
4. Many problems have multiple time periods where the number of vehicles on the street is a function of time period. This problem has been briefly discussed in Bodin [2]; however, no algorithm has ever been implemented for this problem.
5. The third component in the objective can be the ride time of the vehicle. To expand the objective in this manner will allow us to trade off between operating cost and customer inconvenience. We believe the algorithms given in this paper and in [10] can be altered to handle this third term, but such an analysis has not been carried out in the context of the assumptions of this paper. It is interesting to note that in analyzing the Mobility data, total vehicle ride time rarely varied by more than 3%,

regardless of the choice of A and B. It may, therefore, turn out that total vehicle ride time may be an insensitive variable in the context of the model described in this paper.

Acknowledgment

We wish to thank Tom Doyle and Daniel Casco of the University of Maryland for their programming assistance on this project and Art Swersey and Harry Psaraftis and some unknown referees for their critical reviews. We wish also to thank Bob Stark of the State of Maryland Department of Transportation and the Project Mobility staff for providing us with data and for evaluating the test solutions. Finally, we wish to thank Eldon Ziegler, Mike Marcowski and Ed Neigut of UMTA for their support on this project. The research was sponsored in part by UMTA Contract Number MD-06-0027 and a purchase order from the National Bureau of Standards.

REFERENCES

[1] Bodin, L., B. Golden, A. Assad and M. Ball, "The State of the Art in the Routing and Scheduling of Vehicles and Crews," to appear in Computer and Operations Research, 1983.

[2] Bodin, L., "Specifications for a Subscriber Dial-a-Ride System," unpublished report, 1980.

[3] Bodin, L., J. Greenberg and T. Sexton, "A New Algorithm for the Multi-Vehicle Subscriber Dial-a-Ride Problem," presented at the ORSA/TIMS 1982 meeting in San Diego, California, October 1982.

[4] Lin, S., "Computer Solutions of the Travelling Salesman Problem," Bell System Technical Journal, 44, pp. 2245-2269, 1965.

[5] Psaraftis, H., "Dynamic Programming Algorithms for Specially Structured Sequencing and Routing Problems in Transportation," Ph.D. Thesis, M.I.T., Cambridge, Massachusetts, 1978.

[6] Psaraftis, H., "A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem," Transportation Science, 2, pp. 130-154, 1980.

[7] Psaraftis, H., "Two Heuristic Algorithms for the Single Vehicle Many-to-Many Dial-a-Ride Problem," to appear in Transportation Research.

[8] Psaraftis, H. and G. Tharakan, "A Dynamic Programming Approach to the Dial-a-Ride Problem: An Extension to the Multi-Vehicle Case," Report R79-39, Dept of Civil Engineering, M.I.T., August 1979.

[9] Sexton, T. R., "The Single Vehicle Many-to-Many Routing and Scheduling Problem," Ph.D. dissertation, SUNY at Stony Brook, 1979.

[10] Sexton, T. R. and L. D. Bodin, "The Single Vehicle Many-to-Many Routing and Scheduling Problem with Desired Delivery Times," submitted to Transportation Science, 1981.

[11] Sexton, T. R. and L. D. Bodin, "The Single Vehicle Many-to-Many Routing and Scheduling Problem with Customer-Dependent Dwell Times," Working Paper, SUNY at Sony Brook, 1981.

[12] Sexton, T. R., Y. M. Choi and L. D. Bodin, "Extensions of the Single Vehicle Many-to-Many Routing and Scheduling Algorithm," presented at the ORSA/TIMS 1982 meeting in San Diego, California, October 1982.

[13] Stein, D., "Scheduling Dial-a-Ride Transportation Systems," Transportation Science, 12, 232-249, 1978.

[14] Stein, D., "Scheduling Dial-a-Ride Transportation Systems," Ph.D. Thesis, Harvard University, 1977.

[15] Stein, D., "An Asymptotic Probabilistic Analysis of a Routing Problem," Mathematics of Operations Research, 3, 89-101, 1978.

[16] Wilson, N., "The Effect of Driver Schedules on Dial-a-Ride Performance," Preprints, Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services, L. Bodin and D. Bergmann, eds., Chicago, Illinois, April 1975.

[17] Wilson, N. and E. Miller, Advanced Dial-a-Ride Algorithms Research Project, Phase II: Interim Report, Report R77-31, Cambridge, Massachusetts, M.I.T., Dept of Civil Engineering, July 1977.

(18) Wilson, N., R. Weissberg and J. Hauser, "Advanced Dial-a-Ride Algorithms," Research Project Final Report, Report R76-20, Cambridge, Massachusetts, M.I.T., Dept of Civil Engineering, March 1976.

DISTRIBUTION LIST

	NO. OF COPIES
Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	4
Dean of Research Code 012A Naval Postgraduate School Monterey, CA 93940	1
Library, Code 55 Naval Postgraduate School Monterey, VA 93940	2
Adjunct Professor L. D. Bodin Code 55Xb Naval Postgraduate School Monterey, CA 93940	45