Socio-Econ. Plan. Sci. Vol. 3, pp. 75–85 (1969). Pergamon Press. Printed in Great Britain

DESIGN OF SCHOOL BUS ROUTES BY COMPUTER

RITA M. NEWTON and WARREN H. THOMAS

Department of Industrial Engineering, State University of New York at Buffalo, Buffalo, New York 14214

(*Received* 1 *October* 1968)

A practical method for generating school bus routes and schedules on a digital computer has been developed. Routing is accomplished by a two step procedure. First, the shortest route a bus of infinite capacity traverses in order to visit all of the stops is determined. This route, the solution of the traveling-salesman problem associated with the set of bus stops, is obtained by an efficient heuristic procedure which yields near-optimal solutions to problems of a realistic size. This single route is then partitioned to provide individual bus routes and schedules which satisfy bus capacity, bus loading policy, and passenger riding time constraints.

INTRODUCTION

ONE ASPECT of school administration which is becoming more difficult and more costly is the operation of the large bus fleet required to transport students who live beyond walking distance from their assigned schools. As the population grows, as school districts become larger, and as the number of students who must be transported increases, the school bus problem becomes one of major importance.

School transportation personnel are concerned primarily with the problems of bus scheduling, *i.e.*, the determination of a route and a time schedule for each bus. At present, most schools prepare bus routes and schedules manually by using a large map of the district and a listing of the school census. This procedure is not only time consuming but also requires an excessive amount of administrative talent which could be better utilized in other endeavours. Moreover, the quality of the bus routes prepared by hand is a function of the scheduler's experience, *i.e.*, the best routes are usually prepared by the most experienced schedulers.

In addition to the bus scheduling problem, school administrators are also interested in evaluating the sensitivity of current fleet operations to various changes. For example, it would be useful to examine how population fluctuations, the addition of buses to the fleet, the location of new schools, or a change in policy concerning the minimum bussing distance would affect bus operations. Therefore, a great need exists for some means of generating school bus routes and schedules with minimum effort.

PROBLEM FORMULATION

This paper describes the first of a series of continuing research projects [5] concerned with the development of practical methods for designing school bus routes by means of a digital computer. A heuristic procedure has been developed and programmed in

FORTRAN IV to generate efficient bus routes and schedules when the school bus problem has the following definition:
Given:

1. a single school location
2. the identification of each bus stop which must be visited
3. the number of students assigned to each stop
4. the matrix of interstop travel times (this matrix is not necessarily symmetric because of one-way streets and turning restrictions)
5. the bus capacity
6. the number of students who can board a bus in one unit of time.

Determine:

1. the set of bus routes required to provide transportation for all students to and from the school
2. the time schedule for each bus route.

Subject to possible constraints concerning:

1. the maximum allowable riding time of the students picked up at the first stop of any route
2. bus loading policy specifying whether or not all students assigned to a stop must always be picked up by the same bus.

It is assumed that:

1. the bus load for any route must not exceed the stated capacity
2. all buses assigned to a particular school have the same capacity
3. all routes originate and terminate at the school.

SOLUTION APPROACH

Bus routing is accomplished by a two step procedure. First, a single near-optimal route which starts at the school, visits every stop once, and terminates at the school is determined. This route, the solution of the traveling-salesman problem associated with the given set of bus stops, is the route which a bus of infinite capacity would traverse. This single route is then partitioned into individual bus routes which satisfy the bus capacity, bus loading policy, and passenger riding time constraints. The order of the bus stops determined in step one is preserved during the partitioning process.

After bus routing has been completed, the schedule or timetable for each individual route is calculated. Each timetable reflects the time required to load the bus at each stop and the time required to unload the bus at the school in addition to the traveling time between stops.

A. *Initial solution*

The initial feasible solution to the traveling-salesman problem is determined by the nearest-city approach, in which the salesman selects as his next stop that city which is nearest to his present location and which has not yet been visited. Although sub-optimizations at each stop do not insure overall optimization, an initial tour selected in this manner is usually better than one selected at random.

B. *Algorithm* 1

Algorithm 1, a systematic procedure for decreasing the total time required to traverse the traveling-salesman tour, is applied to the initial route. This algorithm determines sets of three path segments which can be changed simultaneously without destroying the continuity of the closed tour and without affecting the direction of the unchanged links of the route. The latter constraint is necessary, because the procedure is applicable to non-symmetric as well as symmetric problems.

If the time required to traverse these three new path segments is less than the time required to traverse the segments which they would replace, the route is altered to incorporate the change.

For example, consider a seven stop problem with current tour R_0 = 1–2–3–4–5–6–7–1. A possible three link change, shown in Fig. 1 by the dashed segments, would yield a new tour, R_1 = 1–2–4–5–3–6–7–1.

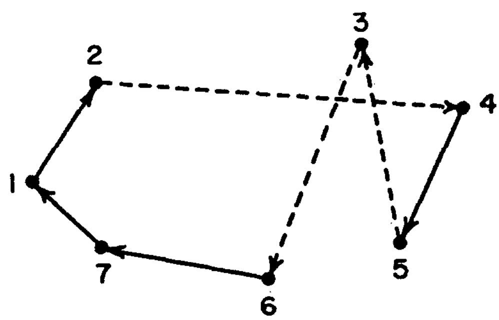


FIG. 1.

If the time required to traverse new links 2–4, 5–3 and 3–6 is less than the time required to traverse old links 2–3, 3–4 and 5–6, tour R_1 replaces R_0 and the algorithm is restarted. If the new tour is not better, the old tour is retained and another set of three link changes is examined. Algorithm 1 provides for the examination of all permissible three link changes. Its execution is complete when no further improvement by any three segment change is possible.

C. *Algorithm* 2

Algorithm 2, a procedure for the possible further reduction of the total time required to traverse the traveling-salesman route determined by Algorithm 1, is applied whenever the time matrix is symmetric. This algorithm determines sets of two path segments which can be changed simultaneously without destroying the continuity of the closed tour. Since such a change requires the reversal of direction of a portion of the links not under consideration, symmetry is a necessary condition.

Consider again a seven stop problem with current tour R_0 = 1–2–3–4–5–6–7–1. A permissible two segment change would yield R_1 = 1–7–4–5–6–3–2–1 as shown in Fig. 2.

If the time required to traverse path segments 7–4 and 6–3 is less than the time to

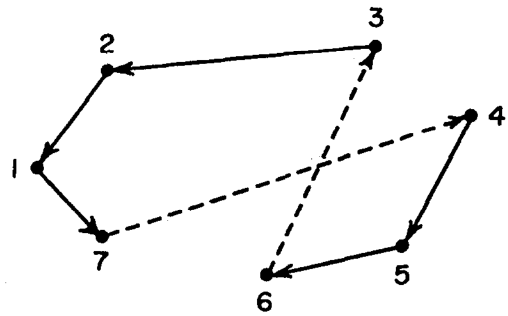


FIG. 2.

traverse 3–4 and 6–7, R_1 replaces R_0 and Algorithm 1 is reapplied. Otherwise, a new set of two path changes are examined.

Algorithm 2 provides for the examination of all permissible two link changes and is completed when no further improvement is possible.

D. *Partitioning*

After the best obtainable traveling-salesman route has been determined by Algorithm 1 and, if appropriate by Algorithm 2, the partitioning procedure is applied. This procedure requires the following additional information:

1. a student load vector specifying the number of students assigned to each stop
2. maximum permissible riding time of the students picked up at the first stop of any route.
3. the number of students who can board a bus in one unit of time
4. bus capacity
5. policy concerning the assignment of any bus stop to more than one bus route.

The partitioning procedure generates a set of bus routes which start at the school, visit the stops of the traveling-salesman tour in the order previously determined until the bus is loaded properly, and return to the school. At each stop, the bus load count is incremented by the appropriate element of the student load vector and the time tally is incremented by both the traveling time from the previous stop and the loading time. If the riding time constraint is exceeded, the previous stop of the traveling-salesman tour is the last one visited by the bus before returning to the school. If the bus capacity constraint is exceeded and if it is required that all students assigned to a stop be picked up by the same bus, the previous stop is the last one serviced by the bus before returning to the school. If the bus capacity constraint is exceeded but all the students assigned to the stop need not be picked up by the same bus, the stop currently under consideration is partially serviced by the bus before returning to the school. The next bus route starts at the school and proceeds directly to the stop of the traveling-salesman tour which immediately follows the last completely serviced stop. All routes are determined in a similar manner. The sequence of stops generated by Algorithms 1 and 2 is preserved at all times.

COMPUTATIONAL PROCEDURE

A. *Nomenclature*

N = total number of bus stops

$[M]$ = a square matrix of order $(N + 1)$ composed of an $N \times N$ interstop travel time matrix augmented by an additional row and an additional column

$M(I,J)$ = number of time units required to travel from bus stop I to bus stop J: $1 \leq I \leq N$, $1 \leq J \leq N$, $I \neq J$

Since the elements on the main diagonal are not needed, they are used to store the time required by the current solution. Therefore,

$M(I,I)$ = number of time units required to travel from bus stop I to the stop which immediately follows it in the current traveling-salesman route.

It is computationally convenient to augment the interstop time matrix with a row and column to provide for the maintenance of the identity of the current route. Therefore,

$M(I,N + 1)$ = identification number of the bus stop which immediately follows bus stop I in the traveling-salesman route, and

$M(N + 1,J)$ = identification number of the bus stop which immediately precedes bus stop J in the traveling-salesman route

$M(N + 1,N + 1)$ is not used.

In the flow charts of Figs. 3 and 4, the identification of the path segments is accomplished by,

O1 = origin of new path segment 1
$T1$ = terminus of new path segment 1
O2 = origin of new path segment 2
$T2$ = terminus of new path segment 2
O3 = origin of new path segment 3
$T3$ = terminus of new path segment 3.

B. *Algorithm* 1

A computational scheme for Algorithm 1 is described by the flow chart given in Fig. 3.

Blocks 1–4 provide for the initialization and incrementation of the I and J indexes.

Blocks 5–6 prevent the creation of an illegal set of changes. Block 5 prevents a change in which indexes I and J are identical. Block 6 ensures that stop J does not immediately follow stop I in the current tour and thereby eliminates the possibility of generating a new route which is identical to the old one.

Block 7 initializes the origin and terminus for each of the three new path segments associated with the current I and J.

Block 8 calculates the time required to traverse the three old path segments which are candidates for replacement.

Block 9 calculates the time required to traverse the proposed new path segments.

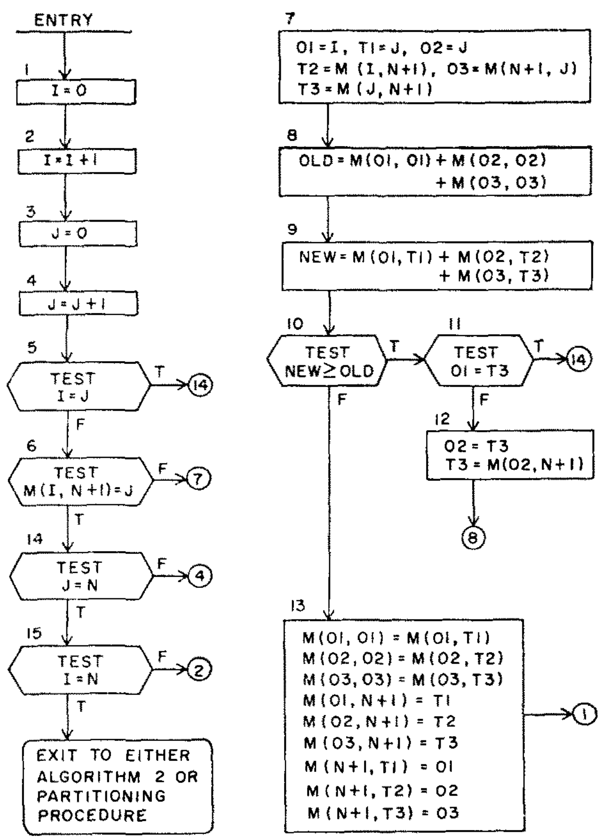
Block 10 determines whether the proposed set of new path segments reduces total transit time.

Block 11 determines whether another set of path segments can be calculated for the current I and J indexes.

Block 12 calculates the origin for new path segment 2 and the terminus for new path segment 3. These are the only modifications required to generate the next set of changes in the sequence for the current I and J indexes.

Block 13 modifies the current traveling-salesman route in order to incorporate the three new path segments, which have been found to reduce total transit time. This modification is accomplished by altering row $(N + 1)$ and column $(N + 1)$ which store the sequence of stops in the new tour and by inserting the new transit times into the main diagonal elements. Algorithm 1 is then restarted at block 1.

Blocks 14–15 determine whether J and I can be updated. If I cannot be updated, Algorithm 1 has been executed for all possible combinations of I and J.



ALGORITHM 1

FIG. 3.

C. *Algorithm* 2

A computational scheme for Algorithm 2 is described by the flow chart given in Fig. 4.

Blocks 1–4 provide for the initialization and incrementation of the I and J indexes.

Block 5 prevents a change in which indexes I and J are identical.

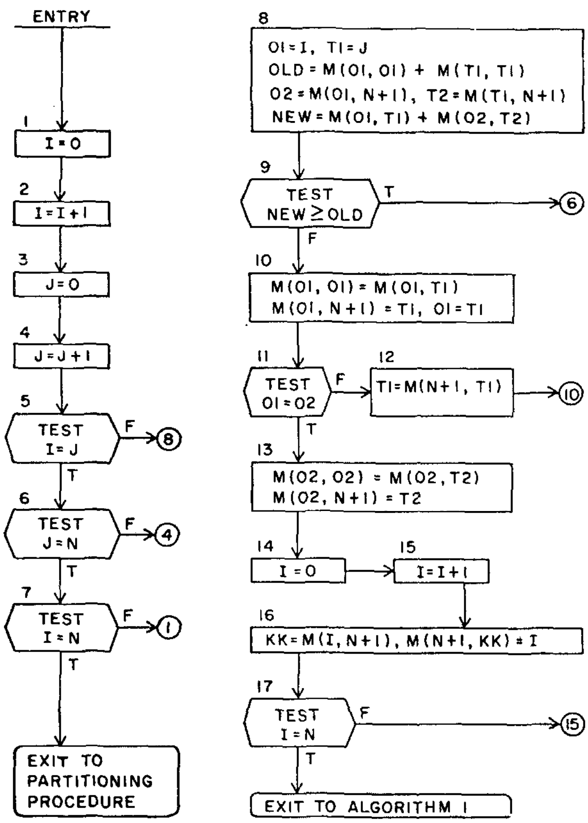
Blocks 6–7 determine whether J and I can be updated. If I cannot be updated, Algorithm 2 has been executed for all possible combinations of I and J. Since, no further improvement can be made by either Algorithm 1 or 2, the partitioning procedure is applied.

Block 8 calculates:

1. the origin and terminus for each of the two new path segments associated with the current I and J.
2. the time required to traverse the two old path segments which are candidates for replacement.
3. the time required to traverse the two proposed new path segments.

Block 9 determines whether the new path segments reduce total transit time.

Blocks 10–17 change the current traveling-salesman route in order to incorporate the two new path segments, which have been found to reduce total transit time. Moreover, in



ALGORITHM 2

FIG. 4.

order to preserve the continuity of the closed tour, the direction of the path segments which lie between the terminus of new path segment 1 and the origin of new path segment 2 are reversed. This modification is accomplished by altering row $(N+1)$ and column $(N+1)$ which store the sequence of stops in the new tour and by inserting the new transit times into the main diagonal elements. Since a change has been made, Algorithm 1 is reapplied.

D. *Other algorithms*

Since the nearest-city approach and the partitioning procedure can be readily programmed, a computational scheme for them is not described in this paper.

EXAMPLE

For illustrative purposes, the Initial Solution Algorithm, Algorithm 1, and Algorithm 2 are applied to the seven bus stop problems defined by the following symmetric interstop travel time matrix.

$$\begin{bmatrix} 0 & 21 & 21 & 12 & 10 & 39 & 7 \\ 21 & 0 & 22 & 21 & 19 & 30 & 16 \\ 21 & 22 & 0 & 9 & 23 & 52 & 14 \\ 12 & 21 & 9 & 0 & 22 & 51 & 5 \\ 10 & 19 & 23 & 22 & 0 & 29 & 17 \\ 39 & 30 & 52 & 51 & 29 & 0 & 46 \\ 7 & 16 & 14 & 5 & 17 & 46 & 0 \end{bmatrix}$$

A. *Initial solution algorithm*

After the Initial Solution Algorithm has been applied and the extra row and column added, matrix M becomes:

$$[M] = \left[\begin{array}{ccccccc|c} 7 & 21 & 21 & 12 & 10 & 39 & 7 & 7 \\ 21 & 19 & 22 & 21 & 19 & 30 & 16 & 5 \\ 21 & 22 & 22 & 9 & 23 & 52 & 14 & 2 \\ 12 & 21 & 9 & 9 & 22 & 51 & 5 & 3 \\ 10 & 19 & 23 & 22 & 29 & 29 & 17 & 6 \\ 39 & 30 & 52 & 51 & 29 & 39 & 46 & 1 \\ 7 & 16 & 14 & 5 & 17 & 46 & 5 & 4 \\ \hline 6 & 3 & 4 & 7 & 2 & 5 & 1 & 0 \end{array}\right]$$

The initial traveling-salesman route starting at stop 1 is 1–7–4–3–2–5–6–1. The total time required to traverse this route, the sum of the N main diagonal elements, is 130 time units.

B. *Algorithm* 1

Algorithm 1 is now applied to the initial traveling-salesman route. The first accepted route change occurs when $O1 = 2$, $T1 = 1$, $O2 = 3$, $T2 = 5$, $O3 = 6$, and $T3 = 2$. In this case, OLD $= 80$ and NEW $= 74$. Therefore, the traveling-salesman route 1–7–4–3–5–6–2–1, which has a total transit time of 124 time units, is accepted. Algorithm 1 is reapplied to this new route. When $O1 = 1$, $T1 = 5$, $O2 = 2$, $T2 = 7$, $O3 = 3$, and $T3 = 1$, OLD $= 51$

and NEW = 47. Therefore, the traveling-salesman route 1–5–6–2–7–4–3–1, whose total transit time is 120 time units, is accepted. Repeated application of Algorithm 1 results in no further reduction of the total transit time.

When Algorithm 1 is exited, matrix M is:

$$[M] = \left[\begin{array}{ccccccc|c} 10 & 21 & 21 & 12 & 10 & 39 & 7 & 5 \\ 21 & 16 & 22 & 21 & 19 & 30 & 16 & 7 \\ 21 & 22 & 21 & 9 & 23 & 52 & 14 & 1 \\ 12 & 21 & 9 & 9 & 22 & 51 & 5 & 3 \\ 10 & 19 & 23 & 22 & 29 & 29 & 17 & 6 \\ 39 & 30 & 52 & 51 & 29 & 30 & 46 & 2 \\ 7 & 16 & 14 & 5 & 17 & 46 & 5 & 4 \\ \hline 3 & 6 & 4 & 7 & 1 & 5 & 2 & 0 \end{array}\right]$$

The traveling-salesman route determined by Algorithm 1 is 1–5–6–2–7–4–3–1.

C. *Algorithm* 2

Algorithm 2 is applied to the traveling-salesman route determined by Algorithm 1. The first beneficial change occurs when $I = 2$, $J = 3$, $O2 = 7$, $T2 = 1$, OLD = 37 and NEW = 29. Therefore, the traveling-salesman route 1–5–6–2–3–4–7–1, whose total transit time is 112 time units, is accepted.

When Algorithm 2 is exited, matrix M is:

$$[M] = \left[\begin{array}{ccccccc|c} 10 & 21 & 21 & 12 & 10 & 39 & 7 & 5 \\ 21 & 22 & 22 & 21 & 19 & 30 & 16 & 3 \\ 21 & 22 & 9 & 9 & 23 & 52 & 14 & 4 \\ 12 & 21 & 9 & 5 & 22 & 51 & 5 & 7 \\ 10 & 19 & 23 & 22 & 29 & 29 & 17 & 6 \\ 39 & 30 & 52 & 51 & 29 & 30 & 46 & 2 \\ 7 & 16 & 14 & 5 & 17 & 46 & 7 & 1 \\ \hline 7 & 6 & 2 & 3 & 1 & 5 & 4 & 0 \end{array}\right]$$

Application of Algorithm 1 to the route determined by Algorithm 2 results in no further change. Therefore, the last traveling-salesman route accepted, 1–5–6–2–3–4–7–1, is the solution to the problem.

D. *Partitioning*

The approach of generating a single route and then, while preserving its sequence, partitioning it into several routes is a heuristic one which is reasonable for the large scale bus problem but which is not appropriate for problems which consist of such a small number of stops. Therefore, in order to emphasize this point, the partitioning procedure is not applied to the illustrative example.

COMPUTATIONAL EXPERIENCE

Problems involving 50, 60, 70, and 80 bus stops, each subject to four different sets of constraints, required 1.19, 2.11, 3.73, and 6.59 min respectively for solution on an IBM 7090 computer.

The 33-city, 42-city, and 57-city traveling-salesman problems described in [4] were solved in 0.71, 1.14 and 2.88 minutes respectively. Routes which covered 10944, 704, and 13160 distance units respectively were obtained by this method. Solving the same problems, Karg and Thompson [4] determined routes which covered 10861, 699, and 12955 distance units respectively. This demonstrates the fact that these algorithms can generate reasonable routes with minimal computer time.

Although this method was not designed to solve the classical delivery problem, application of it to the twelve-stop delivery problem described in [1] and [2] yielded a set of truck routes which covered 306 miles. For the same problem, Clarke and Wright [1] determined routes which covered 290 miles and Dantzig and Ramser [2] generated routes which covered 294 miles.

DISCUSSION

A. *Comparison with the delivery problem*

The absence of extensive literature dealing with the school bus problem is consistent with the fact that interest in this problem area is relatively recent. However, more information is available on a closely related problem, the delivery problem.

The delivery problem is concerned with the determination of routes for a vehicle, initially located at a depot, which visits a number of delivery or pickup points and returns to the depot. Since the capacity of the vehicle is less than the total quantity of goods which must be transported, several trips must be made. If the capacity of the vehicle is greater than the quantity of goods to be transported, the problem degenerates into the traveling-salesman problem.

It is evident that the school bus scheduling problem and the delivery problem are conceptually the same. Therefore, an analytical solution to one is applicable to the other. However, at the present time, no analytical solutions are available to solve practical problems of either type. Since methods which guarantee an optimum solution are non-existent, out of necessity heuristic methods must be used. Therefore, even though the school bus problem and the delivery problem are similar, differences which exist in them must be considered.

The school bus scheduling problem is characterized by a large number of stops which must be visited. Frequently, the bus fleet must service several hundred stops. Moreover, an interstop *time* matrix is more appropriate than an interstop *distance* matrix for the school bus problem inasmuch as student riding time is of greater concern than the total mileage covered by the buses. Since the bus stops are located in the same general area, the time matrix usually consists of elements of similar magnitude. The interstop time matrix is characteristically non-symmetric for reasons such as: restrictions imposed upon the crossing of busy streets by students, one-way streets, limited access highways, and restrictions imposed on vehicle turns at intersections. Another characteristic of the school bus problem is that the school at which all bus routes originate and terminate is usually centrally located.

The delivery problem on the other hand usually contains relatively few stops, interstop travel times of large variability and a depot which is often not centrally located. Frequently, a symmetric interstop time or distance matrix is appropriate.

B. *Evaluation of heuristic procedures*

Any heuristic procedure should be evaluated on two bases. First, in order to be acceptable, the method must be reasonable with respect to the size of the problem it can handle, the assumptions it imposes, and the logical processes it uses. Second, in order to be practical, the procedure must be able to yield answers to a problem at a cost commensurate with the value received. The differences between the delivery problem and the school bus scheduling problem are great enough to make the widespread interchange of heuristic procedures infeasible.

In the school bus scheduling problem, each bus route services a large number of stops and therefore the sequence in which these stops are visited is of primary concern. Moreover, since the return trip to the school introduces a relatively small additional time increment to the bus route and since this return trip occurs infrequently when compared to the total number of stops visited, it is reasonable to place emphasis on the initial route sequencing part of the problem. Therefore, this paper is primarily concerned with the determination of the traveling-salesman route associated with the given interstop time matrix. The non-optimalities introduced by the partitioning of this single route are present but relatively small when compared with the benefits derived from the near-optimal sequencing of all the stops. Any scheme more elegant than the partitioning process must be economically justified when dealing with problems of the magnitude encountered in practice. Although the partitioning process is appropriate for the school bus scheduling problem, care must be exercised in applying it to a delivery type problem.

None of the methods currently available [3] for solving the delivery problem are capable of handling problems of the size present in the school bus scheduling environment.

SUMMARY AND FUTURE WORK

This paper presents an efficient, automatic procedure for solving the school bus scheduling problem. It offers an initial feasible solution to an exceedingly complex problem which faces every community. Future areas of research include: the improvement of the individual bus routes determined by the partitioning process, the provision of a means of generating bus routes whose origin and terminus are different, and the development of a procedure, which not only designs the bus routes for all the schools which use the same fleet of buses, but also assigns each bus to a set of routes in an optimal fashion.

REFERENCES

1. G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, *Ops Res.*, **12**, 568–581 (1964).
2. G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem", *Mgmt. Sci.*, **6**, 80–91 (1959).
3. R. L. Hayes, "The Delivery Problem", paper presented at the Thirty-first National Meeting of ORSA, New York, June 1967.
4. R. L. Karg and G. L. Thompson, "A Heuristic Approach to Solving Traveling-salesman Problems", *Mgmt. Sci.* **10**, 225–248 (1964).
5. R. M. Newton, "A School Bus Scheduling Algorithm", M.S. Thesis, State University of New York at Buffalo, 1967.