A computer-based method was developed that can translate available data about schools, students, and bus facilities into a set of bus routes and schedules prior to the start of the school year. Each route can be so designed via the computer model that student riding time and bus capacity constraints are satisfied at the same time that total bus travel (including running empty) and number of routes required to service all the stops are minimized. The mathematical models developed were programmed in FORTRAN IV for use on a CDC 6400 computer and were applied to four schools. An efficient routing system involving six possible bus route origins and 96 stops was developed for one of these schools in 61 seconds using a CDC 6400 computer. A bibliography and program listing are appended. (Author/MF)
DEVELOPING A COMPUTER PROGRAM FOR BUS ROUTING

Rita M. Newton
Warren H. Thomas
State University of New York at Buffalo
Buffalo, New York 14214

The Research Foundation of the
State University of New York
P.O. Box 7126
Albany, New York 12224

July 1970

U.S. DEPARTMENT OF
HEALTH, EDUCATION AND WELFARE
Office of Education
Bureau of Research
FINAL REPORT
Project No. 9-B-124
Grant No. OEG-2-9-420128-1062(010)

DEVELOPING A COMPUTER PROGRAM FOR BUS ROUTING

Rita M. Newton
Warren H. Thomas
State University of New York at Buffalo
Buffalo, New York 14214

July 1970

The research reported herein was performed pursuant to a grant with the Office of Education, U.S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.
ABSTRACT

This report describes and evaluates a practical computer based method for translating data concerning:

1. the location of each school to be serviced by a bus fleet,
2. the locations and numbers of students to be transported to each school,
3. the time interval during which the students are to be transported, and
4. the available bus facilities

into a set of bus routes which specify school-to-school sequencing of each bus and the stop-to-stop route to be followed in traveling to every school. Each route is designed in such a way that the bus capacity and student riding time constraints are satisfied while attempting not only to minimize the total bus travel time (including running empty) for a school but also to minimize the number of routes required to service all the stops associated with the school. The mathematical models developed were programmed in FORTRAN IV for use on a CDC 6400 computer and were applied to four schools in the Williamsville New York Central School District. An efficient routing system involving 6 possible bus route origins and 96 stops was developed for one of these schools in 61 seconds on a CDC 6400 computer.
ACKNOWLEDGEMENTS

This report is based on the dissertation of Mrs. Rita M. Newton entitled "Bus Routing in a Multi-School System" completed as part of the requirements of the Ph.D. program in Operations Research at the State University of New York at Buffalo. This program is administered by the Department of Industrial Engineering.

The effectiveness of this project was greatly enhanced by the cooperation and enthusiastic support of John Allan, assistant to the superintendent of schools, and Charles Bastian, director of transportation, of the Williamsville, New York Central School District. Appreciation is also due to the Board of Cooperative Educational Services, First Supervisory District of Erie County which provided all the key-punching services for this project.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. LITERATURE REVIEW</td>
<td>4</td>
</tr>
<tr>
<td>A. School Bus Routing Problem</td>
<td>4</td>
</tr>
<tr>
<td>B. Delivery Problem</td>
<td>7</td>
</tr>
<tr>
<td>III. PROBLEM FORMULATION</td>
<td>12</td>
</tr>
<tr>
<td>A. Statement</td>
<td>12</td>
</tr>
<tr>
<td>B. Discussion of Problem Statement</td>
<td>14</td>
</tr>
<tr>
<td>C. Mathematical Model</td>
<td>19</td>
</tr>
<tr>
<td>IV. METHOD OF SOLUTION</td>
<td>25</td>
</tr>
<tr>
<td>A. General Procedure</td>
<td>25</td>
</tr>
<tr>
<td>B. Selection of Bus Route Origins</td>
<td>27</td>
</tr>
<tr>
<td>C. Determination of Trial Routes: Nearest-City Approach</td>
<td>31</td>
</tr>
<tr>
<td>D. Determination of Trial Routes: Algorithm A</td>
<td>32</td>
</tr>
<tr>
<td>E. Partitioning Procedure</td>
<td>36</td>
</tr>
<tr>
<td>F. Improvement Process</td>
<td>37</td>
</tr>
<tr>
<td>G. Acceptance Criteria</td>
<td>38</td>
</tr>
<tr>
<td>H. Final Allocation of Origins to Individual Routes</td>
<td>39</td>
</tr>
<tr>
<td>V. COMPUTATIONAL PROCEDURE</td>
<td>43</td>
</tr>
<tr>
<td>A. Algorithm A</td>
<td>43</td>
</tr>
<tr>
<td>B. Lower Bound</td>
<td>47</td>
</tr>
<tr>
<td>VI. EVALUATION</td>
<td>50</td>
</tr>
<tr>
<td>A. Model and Method</td>
<td>50</td>
</tr>
<tr>
<td>B. Computational Experience</td>
<td>54</td>
</tr>
<tr>
<td>VII. SUMMARY AND RECOMMENDATIONS</td>
<td>85</td>
</tr>
<tr>
<td>VIII. BIBLIOGRAPHY</td>
<td>87</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>91</td>
</tr>
</tbody>
</table>
# List of Figures and Tables

<table>
<thead>
<tr>
<th>Figure/Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Algorithm A</td>
<td>44</td>
</tr>
<tr>
<td>Table 1</td>
<td>Thompson Data</td>
<td>57</td>
</tr>
<tr>
<td>Table 2</td>
<td>Summary of Results for Problems in the Literature</td>
<td>73</td>
</tr>
<tr>
<td>Table 3</td>
<td>Summary of Sample School Bus Scheduling Problems</td>
<td>76</td>
</tr>
<tr>
<td>Table 4</td>
<td>Academy School</td>
<td>79</td>
</tr>
<tr>
<td>Table 5</td>
<td>Forest School</td>
<td>81</td>
</tr>
<tr>
<td>Table 6</td>
<td>South Senior High School</td>
<td>82</td>
</tr>
<tr>
<td>Table 7</td>
<td>Dodge School</td>
<td>83</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

School districts are aware of the power of the digital computer as a tool for both reducing cost and improving service in the management of their educational programs. The computer has already proven its worth in financial accounting, in personnel administration, in class scheduling, and in planning school construction.

One area of school administration which has not yet been adequately served by the capability of the computer is the management of the transportation system. Since the cost of procurement, maintenance, and operation of a bus fleet requires a large portion of a school district's budget, the director of transportation is expected to minimize these costs while simultaneously providing an acceptable level of service.

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. Since a single school may have as many as two hundred bus stops and a district as many as twenty-five schools, this procedure is not only time consuming but requires an excessive amount of administrative talent which could be better utilized in other endeavors. 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 being able to evaluate the sensitivity of current fleet operations to various changes. For example, it would be useful to be able to easily examine how population fluctuations, a change in bus fleet size, a change in school boundaries, or a change in policy concerning the maximum allowable walking distance would affect bus operations.
Therefore, a great need exists for some means for generating school bus routes and schedules efficiently.

This dissertation describes and evaluates a computer based methodology for translating data concerning:

1. the identification and location of each school which is to be serviced by the bus fleet,
2. the location and number of students to be transported to each school,
3. the time interval during which students are to be transported, and
4. the available bus facilities

into a set of bus routes which specify school-to-school sequencing of each bus and the stop-to-stop route to be followed in traveling to the school. Each route is designed in such a manner that the bus capacity and maximum allowable student riding time constraints are satisfied while attempting to not only minimize the total bus travel time (including running empty) for a school but also to minimize the number of routes required to service all the stops associated with the school.

The output of the computer program based upon this methodology is the detailed information needed to prepare specific bus schedules and bus passes for each student. In addition it provides a means for determining overall measures of schedule performance as total travel time, average riding time, average bus load, etc. These performance measures coupled with the inexpensive and rapid computer development of a complete set of bus routes for a school district make feasible a quantitative evaluation of the effect of any administrative policy changes on the transportation system.
The computer model makes operationally feasible the generation of bus routes and schedules immediately prior to the start of the school year when knowledge of transportation demands is most accurate. Moreover, it provides a tool for evaluating alternative new school sites in a district experiencing significant population growth.
II. LITERATURE REVIEW

A. School Bus Routing Problem

A review of the literature concerned with school transportation reveals that most of the studies conducted in this area deal with standards for school buses, methods of allocating funds to school districts for transportation purposes, surveys of the status and philosophy of school transportation in various geographic regions, and examination of legislation affecting school transportation. The absence of extensive literature dealing with the school bus routing problem indicates that either interest in this problem area is relatively recent or that a large portion of the attempts at developing a general method for school bus routing have proved to be fruitless and have therefore not been reported.

Dantzig, Fulkerson, and Johnson\textsuperscript{12} gave the allusion that the problem of using a digital computer to design school bus routes was simple and straightforward. Intrigued by this, Boyer\textsuperscript{5} attempted to solve a five stop school bus problem by applying the simplex algorithm. Because the application of a linear programming technique to the problem seemed to be highly impractical, he developed the Sequential Steps Method\textsuperscript{5}. This procedure is based upon the premise that all students should be transported the shortest possible distance to school. Each route starts at the bus stop which is the farthest distance from the school and proceeds along the shortest path to the school in such a manner that no isolated bus stops are created and that a bus stop is assigned to only one route.

Thompson\textsuperscript{34} tested the effectiveness of the Sequential Steps Method in determining school bus routes for a hypothetical school. He learned that although student riding time was lowered, this method tended to generate more routes for a school than other manual methods favored by
the transportation personnel participating in the study. Although this method was sequential, it was not amenable to computer programming. Moreover, it made no provision for imposing bus capacity and passenger riding time constraints.

Boyer\textsuperscript{6,7} described a procedure for designing school bus routes in which a set of bus routes or a route system is developed manually. A computer is then used to list all possible ways in which the stops of a bus route can be visited. By inspection, the best permutation for each individual route of the route system is determined. After several arbitrary route systems have been analyzed in this manner, the best route system is selected.

Since this method requires a great deal of manual work and personal judgement, it offers little improvement over the current manual methods of designing bus routes and schedules. Moreover, this method makes no provision for imposing a passenger riding time constraint.

Tillman\textsuperscript{35} applied the technique of dynamic programming to the school bus routing problem in which all routes start and end at the school. In this formulation stage \( j \) was the number of the bus being loaded or the number of the route being designed, the decision variable was the number of stops made by bus \( j \) or the number of stops assigned to route \( j \), the stage input was the number of stops not yet serviced at stage \( j \), and the return obtained at each stage was the minimum number of miles traveled by bus \( j \). The objective was to minimize the sum of the returns subject to a bus capacity constraint. Using this method, an optimum solution was obtained for a problem involving five bus stops, three buses, and forty students.

Although dynamic programming guarantees an optimal solution and
allows the interstop travel time matrix to be asymmetric, it is an
impractical method because of the extremely large number of calculations
which have to be performed for even relatively small school bus routing
problems.

Newton and Thomas\textsuperscript{29,30} described a practical method for generating
school bus routes and schedules by computer. Given the matrix of inter-
stop travel times, which may be asymmetric, 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 then partitioned into
individual bus routes which satisfy bus capacity, bus loading policy,
and passenger riding time constraints. The order of the route determined
in step one is preserved during the partitioning process and all routes
originate and terminate at the school. This heuristic procedure has
been used to solve an eighty stop problem in approximately six minutes
on a 7090 computer.

Davis\textsuperscript{15} described a branch and bound algorithm for solving the
school bus routing problem in which the interstop travel time matrix is
symmetric, bus capacity and passenger riding time constraints are imposed,
and all routes start and end at the school. This procedure partitioned
the set of all possible routes into mutually exclusive subsets of routes
by solving a series of transportation problems. The cost of each sub-
set was the total traveling time required by the optimum solution to the
transportation problem associated with the subset.

Although this method is amenable to computer programming and guaran-
tees an optimal solution, it proved to be an impractical one. For
example, an attempt to solve a thirteen stop bus problem, whose interstop travel time matrix was symmetric, subject only to a bus capacity constraint from was abandoned when no feasible solution was obtained after approximately eighteen minutes CDC 6500 time had been spent on the problem.

B. Delivery Problem

Considerably more information is available on the closely related 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. The delivery problem usually is not subject to a traveling time or distance constraint whereas the school bus routing problem is nearly always constrained by both bus capacity and maximum allowable student riding time.

Balinski and Quandt formulated the delivery problem as an integer programming problem. Although this method guarantees an optimal solution, it is undesirable because of the large number of variables and constraints required to express a delivery problem involving relatively few stops. Moreover, the available integer programming algorithms are often unable to achieve solutions to even moderately large problems even though theoretically they should always determine the optimum solution.

Hayes applied the branch and bound method of Little, Murty, Sweeney, and Karel to the delivery problem. This procedure partitioned the set of all possible routes into mutually exclusive subsets by either assigning or not assigning a particular link to a route. The cost of each
subset was the minimum possible total length of any set of routes containing the link assignment associated with the subset.

Although this method guarantees an optimal solution and is amenable to computer programming, it also is an impractical method because a large number of time consuming operations must be performed. For example, an attempt to solve the thirteen stop problem from 14 was abandoned when no solution was obtained after one hour was spent on the Control Data G-21 computer.

Dantzig and Ramser 14 described a heuristic procedure for solving the delivery problem when all trucks have the same capacity. After the delivery points are arranged in numerically ascending order with respect to demand and the number of stages required to achieve solution has been calculated, this algorithm synthesizes routes by a stage-wise aggregation of the delivery points, i.e. in stage one, pairs of points are joined, in stage \( K \), groups of \( K \) points are joined to other groups of \( K \) points. At each stage the points are combined in such a manner that truck capacity is not exceeded and the sum of the interstop distances is minimized.

Although this method is sequential and amenable to computer programming, personal judgement may be required in adjusting the final solution which may not be uniquely defined. Moreover, this procedure places more emphasis on insuring that the trucks are loaded to capacity than on minimizing the total distance traveled. It also appears that this algorithm would require considerable alteration to handle the asymmetric case because direction is not considered when the interstop distance matrix is searched to determine the minimum entry. This method would probably be quite time consuming in solving problems which involve more than 20-30 stops. Each entry of the distance table in stage \( K \),
where $K = 2, 3, \ldots, N$ and $N$ is the number of stages required, contains the shortest route which starts at the depot, visits $2^K$ points and returns to the depot. Therefore, many traveling-salesman problems must be solved at each stage. For example, if after 2 stages 100 stops are aggregated into 25 groups of 4 stops each, then the distance table for stage 3 requires the solution of 300 traveling-salesman problems of 9 stops.

Clarke and Wright\textsuperscript{9} described a heuristic procedure for solving the delivery problem when all vehicles do not have the same capacity. After the demand points are arranged in numerically ascending order with respect to distance from the origin, the algorithm assigns each stop to one vehicle or route. Then each pair of points is examined to determine the savings which would result if they were linked together on the same route instead of being assigned to different routes. A savings calculation is made only for those points, currently linked to the origin, which could be assigned to the same route without violating vehicle capacity constraints. The two points associated with the maximum savings are then assigned to the same route. After repeating the procedure until no further savings can be calculated, the resulting set of routes is the solution to the problem.

This method is amenable to computer programming and appears to be efficient for solving the delivery problems in the literature. However, it cannot be readily extended to handle the case of the asymmetric interstop distance matrix because direction is not considered in calculating the savings or the criterion for accepting or rejecting the linkage of two points on the same route. Moreover, the largest problem for which solution success with this method has been reported is one involving thirty-two stops\textsuperscript{18}.
Cochran\textsuperscript{10,36} made two modifications to the algorithm of Clarke and Wright\textsuperscript{9}. The first change permits the reassignment of vehicles to individual routes each time a vehicle becomes available as a result of the linkage of two points on the same route. The second change allows an upper bound to be placed upon the length of any route. Although the first modification insures that the vehicles will be more fully utilized and the second modification is a useful one, neither the probability of reaching optimality nor the efficiency of the algorithm is increased by their inclusion.

Braun\textsuperscript{8} described a simulation approach to the delivery problem. First, delivery points are randomly assigned to an individual route subject to the vehicle capacity constraint. Each route is then improved by applying a traveling-salesman algorithm. After a specified number of sets of routes are developed in this manner, the set which covers the least number of miles is selected as the solution.

Although this procedure is simple and amenable to computer programming, it proved to be less efficient than other available methods with respect to the quality of the routes produced and the amount of computer time required to achieve solution. Moreover, the quality of the sets of routes tends to become poorer as the number of delivery points increases.

After abandoning the branch and bound method for solving the delivery problem, Hayes\textsuperscript{21} developed a computerized version of a procedure which an experienced dispatcher might employ for routing trucks. Since this method assumes that the warehouse or depot is located near the center of the scatter of customers and performs poorly when this assumption is not satisfied, it does not appear to be superior to the available
heuristic methods.

Gaskell\textsuperscript{18} reported on his experiments with the method of Clarke and Wright\textsuperscript{9}. He devised five functions for calculating the savings which would result from linking two points on the same route. After applying each of them to a set of delivery problems, he concluded that none was uniformly better and that the function used by Clarke and Wright to calculate savings was a reasonable one.

Although more work has been done on the delivery problem, successful solution has been reported only for relatively small problems with no indication of apparent near future breakthroughs for larger problems involving asymmetric interstop time/distance matrices, routes whose origin and terminus do not coincide, and restrictions upon the length of a route, vehicle capacity, and the number of routes used to service all the stops. Indeed, solution success on the school bus routing problem may provide a means of solving the delivery problem.
III. PROBLEM FORMULATION

A. Statement

Thus far, the school bus routing problem has been solved for an environment which has been greatly simplified by the imposition of various assumptions. Most solutions not only exclude constraints on some of the route characteristics but also assume that the origin and terminus of a bus route coincide, whereas in reality they are not necessarily the same. In fact, there often may be several possible origins for the set of bus routes servicing a particular school. Therefore, it would be desirable to be able to judiciously select the proper combination of origins from the set of possible origins for the system of bus routes servicing a particular school.

With these thoughts in mind, this dissertation is concerned with the determination of all bus routes for a school district. Buses are routed from school to school picking up students as they travel. A heuristic procedure has been developed and programmed in FORTRAN IV to generate efficient bus routes when the school bus routing problem has the following definition:

Given:

1. the number of time periods used by the school district for bussing
2. for each time period
   a. the number and location of schools to be serviced
   b. the number of possible origins for the bus routes
   c. the identification of each origin
   d. the number of buses available at each origin at the beginning of the period
3. for each school to be serviced during the same time period
   a. the identification of the school,
   b. the identification of each stop to be visited,
   c. the number of students assigned to each stop,
   d. the matrix of interstop travel times between each possible origin and every stop assigned to the school,
   e. the matrix of interstop travel times for all pairs of stops except those links involving an origin,
   f. the bus capacity,
   g. the maximum allowable student riding time, and
   h. the criterion for accepting a set of feasible routes as the quasi-optimal solution.

Determine for each school:

1. the set of bus routes and schedules required to provide transportation for all students either to school or from school and

2. a lower bound on the total number of time units required to traverse all the bus routes

such that:

1. no more than the absolute minimum number of routes required to transport the students plus one route will be used for any school,

2. the bus capacity and maximum allowable student riding time constraints will be satisfied,

3. the criterion for accepting a set of feasible routes as
the quasi-optimal solution will be satisfied, and

4. optimization will be with respect to minimizing the total traveling time for a set of routes.

It is assumed that:

1. all buses assigned to a particular school have the same capacity,
2. all routes for a particular school will be subject to the same maximum allowable student riding time constraint,
3. the matrix of interstop travel times may be asymmetric,
4. a stop is assigned to only one route,
5. the traveling time from any bus stop to the terminus is less than the maximum allowable student riding time,
6. the origin and terminus of any bus route do not necessarily coincide,
7. a bus services only one route for a particular school, and
8. the number of buses available at the beginning of period i is adequate to meet the needs of all students to be serviced during period i, i.e. it is possible to service all the routes simultaneously.

B. Discussion of the Problem Statement

Most districts set the opening and closing times of the schools serviced by the same bus fleet so that all students can be transported during two or three non-overlapping bussing periods whose total elapsed time is approximately two and one-half hours. Schools maintaining common hours of operation are serviced during the same bussing period.

The data required to design the routes for each school can be
developed either manually or by computer from the map of the area and the associated census tract. Each stop serviced by the school is usually identified by both name and number in order to facilitate communication between the administrators and the bus drivers. The elements of the interstop travel time matrix are calculated from a map of the area so that they include the effects of both distance and expected driving conditions in traveling between every pair of points. Often in an effort to reduce the size of the interstop travel time matrix, bus stops adjacent to each other on the same side of the road are combined into a single stop. The information required to assign students to a bus stop can be extracted from the census tract. Usually upper limits are imposed upon the number of students assigned to a bus stop in order to reduce the noise level and the possibility of landscape damage in residential areas.

Optimization is with respect to not only minimizing the total time required to traverse a set of routes but also with respect to minimizing the number of routes required to transport all the students of a particular school. Both of these factors contribute heavily toward the operational costs of maintaining a bus fleet. Minimizing the number of routes insures that the buses are being utilized to full capacity and also tends to reduce the total traveling time associated with a set of routes by eliminating some of the links between an origin and the first stop of any route and the links between the last stop of any route and the terminus. Since schools usually place more emphasis on minimizing the number of routes required to transport all the students than on minimizing the total traveling time for a set of routes, the maximum allowable number of routes for a school will be considered to be a constraint.
The minimum number of routes required to service all the stops assigned to a school is the smallest integer which is greater than or equal to the quotient of the total number of students to be transported and the bus capacity. Since a riding time constraint is imposed on each route and since all the students assigned to a particular bus stop have to be picked up by the same bus in order to avoid confusion, it may be impossible to service all the bus stops by the minimum number of routes. Therefore, a set of routes is considered feasible if each route satisfies the bus capacity and student riding time constraints and if the set contains at most one more than the minimum number of routes required to transport all the students. Thus, a feasible set of routes contains either the minimum number of routes or one more than the minimum number of routes.

Because the set of bus routes for any school is designed by a heuristic procedure, some method must be devised to determine when a set of routes, acceptable to the school, has been developed. One possible criterion is to accept the best set of feasible routes available after the procedure has been executed a specified number of times. A second possible criterion is to accept the first set of feasible routes developed whose total travel time is less than the product of a given factor, greater than one, and the lower bound upon the total time required to traverse all the routes for a school. Another alternative is to accept as the quasi-optimal solution the best set of feasible routes available at the time at which either of the criteria is first satisfied.

The lower bound upon the total time required to traverse a set of routes for a school is the minimum length of time in which all the bus stops could be serviced by the number of routes included in the set.
In calculating the lower bound, it is assumed that all connections between the origin and the first stop of a route, all the connections between the bus stops, and all the connections between the last stop of a route and the terminus are made in the least time consuming manner. Since the lower bound does not include the effects of the bus capacity and passenger riding time constraints, the probability of developing a set of feasible routes whose total traveling time equals the lower bound is very low. However, even though the lower bound may be unattainable, it provides some measure for assessing the quality of the set of routes developed.

An examination of the set of assumptions under which the problem is to be solved reveals that the environment has not been oversimplified. The bus capacity is based upon the expected girth of children at various age levels, the size of the bus used, and the degree of bus utilization required by the school. The maximum student riding time is dependent upon the size of the area, the scatter of the bus stops and the local or state laws regulating student riding time. Usually, districts consider the bus capacity and the maximum riding time constraints to be fixed for a particular school.

The interstop travel time matrix is characteristically asymmetric for reasons such as: restrictions imposed upon the crossing of busy streets by children, one-way streets, limited access highways, and restrictions imposed on vehicle turns at intersections. Moreover, most schools maintain a bus loading policy which requires that all students assigned to a stop by picked up by the same bus in order to avoid confusion. This restriction implies that a stop is assigned to only one route.
The traveling time from any bus stop to the terminus has to be less than the maximum allowable student riding time in order to insure that no bus stop be isolated. The assumptions that a bus services only one route for a school and that it is possible to service all the routes simultaneously are imposed in order to facilitate the computational procedure used to develop all the bus routes for a district and do not simplify the environment. Thus, this formulation of the school bus routing problem is considered to be a realistic and a reasonable one.
C. Mathematical Model

The bus routing problem for any school, as previously defined, can be expressed as a zero-one integer programming problem. Let:

- \( i \) = origin of any link
- \( j \) = terminus of any link
- \( k \) = number of any possible bus origin
- \( m \) = route number
- \( K \) = number of possible different bus origins for a set of routes
- \( M \) = number of routes allowed for the school
- \( N \) = total number of bus stops (includes the origin and the terminus) assigned to the school
- \( \text{stop } l \ (k) \) = bus origin number \( k \)
- \( \text{stop } N \) = terminus of any route or the school
- \( t(i,j) \) = number of time units required to travel between bus stop \( i \) and bus stop \( j \)
- \( L(j) \) = number of students assigned to bus stop \( j \)
- \( B(k) \) = number of buses available at origin \( k \)
- \( C \) = bus capacity
- \( R \) = maximum allowable student riding time

\[ x(i,j,m) = \begin{cases} 
1, & \text{if link } (i,j) \text{ is assigned to route } m \\
0, & \text{otherwise}
\end{cases} \]

The subscript denoting the particular school has been omitted in order to simplify the nomenclature. The words bus and route are used interchangeably.

The problem can be stated as follows:

Find variables \( x(i,j,m) \) for all combinations of \( i, j, k \) and \( m \) where...
such that the objective function

$$
\sum_{j=2}^{N} \sum_{k=1}^{K} \sum_{m=1}^{M} [t(1(k),j)][x(1(k),j,m)]
$$

\[
+ \sum_{i=2}^{(N-1)} \sum_{j=2}^{N} \sum_{m=1}^{M} [t(i,j)][x(i,j,m)]
\]

is minimized while restrictions one through eight, stated below, are satisfied.

Optimization, as previously stated, is with respect to not only minimizing the total time required to traverse a set of routes but also with respect to minimizing the number of routes required to transport all the students of the school. Since it is difficult to work with two objective functions, and since most schools place greater emphasis on the minimization of the number of routes than on the minimization of the total traveling time, the number of routes required to transport all the students of a school will be considered a constraint and the objective function will involve the total traveling time only. The first term of the objective function is the total time spent in traveling from the origin to the first stop of any route; the second term is the total time spent in traveling between the other pairs of points.
The first constraint,
\[ \sum_{k=1}^{K} \sum_{m=1}^{M} x(1(k),j,m) + \sum_{i=2}^{(N-1)} \sum_{m=1}^{M} x(i,j,m) = 1, \text{ for } j = 2,3,\ldots,(N-1), \]

insures that any bus stop \( j \), where \( j \) is not the origin or terminus of any route, will be the terminus of exactly one link on one route. There will be \((N-2)\) constraints of this type.

The second constraint,
\[ \sum_{j=2}^{N} \sum_{m=1}^{M} x(i,j,m) = 1, \text{ for } i = 2,3,\ldots,(N-1), \]

insures that bus stop \( i \), where \( i \) is not the origin or terminus of any route, will be the origin of exactly one link on one route. There will be \((N-2)\) constraints of this type.

The third constraint,
\[ \sum_{j=2}^{(N-1)} \sum_{k=1}^{K} \sum_{m=1}^{M} x(1(k),j,m) = M, \]

insures that only \( M \) routes are used by counting the number of links between the origin of each route and the first stop serviced by the route.

The fourth constraint,
\[ \sum_{i=2}^{(N-1)} \sum_{m=1}^{M} x(i,N,m) = M, \]

also insures that only \( M \) routes are used by counting the number of
The fifth constraint,
\[ \sum_{j=2}^{(N-1)} \sum_{m=1}^{M} x(l(k),j,m) \leq B(k), \quad \text{for } k = 1,2,\ldots,K, \]
insures that the number of routes starting at a particular origin does not exceed the number of buses available at that origin for the school. There will be \( K \) constraints of this type.

The sixth constraint,
\[ \sum_{j=2}^{(N-1)} \sum_{k=1}^{K} [x(l(k),j,m)]L(j) + \sum_{i=2}^{(N-1)} \sum_{j=2}^{(N-1)} [x(i,j,m)]L(j) \leq C, \]
for \( m = 1,2,\ldots,M \), insures that the bus capacity will not be exceeded by any route. The first term of the constraint counts the number of students picked up at the first bus stop serviced by the route and the second term counts the students picked up at the other stops assigned to the route. There will be \( M \) constraints of this type.

The seventh constraint,
\[ \sum_{i=2}^{(N-1)} \sum_{j=2}^{N} [t(i,j)][x(i,j,m)] \leq R, \quad \text{for } m = 1,2,\ldots,M, \]
insures that no route will exceed the maximum allowable student riding time where the riding time is counted from the first pick-up point. There will be \( M \) constraints of this type.

The eighth constraint,
\[ \sum_{p=1}^{(r-1)} x(i_p,i_{p+1},m) + x(i_r,i_1,m) \leq (r-1), \quad \text{for } m = 1,2,\ldots,M, \]
where \(i_1, i_2, \ldots, i_r\) ranges over all permutations of subsets of the bus stops \(\{2, 3, \ldots, (N-1)\}\) of size \(r\), and \(2 \leq r \leq (N-1)\), prevents the formation of a loop, a route which starts and ends at the same point.

There will be \(\sum_{k=2}^{(N-2)} \binom{(N-2)}{k} \) constraints of this type where \(\binom{z}{x}\) denotes the permutation of \(z\) things taken \(x\) at a time. Since \(\sum_{k=2}^{(N-2)} \binom{(N-2)}{k}\) is greater than \(2(N-2)!\), there will be more than \(2M(N-2)!\) constraints of this type. None of the permutations involve the origin or terminus because no link is allowed to start at the terminus and no link is allowed to end at the origin in the case of the general route whose origin and terminus do not coincide. This is accomplished by setting the travel times associated with these links equal to infinity, i.e. the elements of column one and row \(N\) of the interstop travel time matrix are assigned large values. To illustrate the manner in which the eighth constraint prevents the formation of a loop, let \(r = 2\), \(m = 4\), \(i_1 = 3\), \(i_2 = i_r = 5\), \(x(3,5,4) = 1\), \(x(5,3,4) = 1\) and the rest of the variables, \(x(i,j,4)\) equal zero. In this example, route four starts at stop three, proceeds to stop five, and then returns to stop three. According to the eighth constraint, \(x(3,5,4) + x(5,3,4) = 2 = r\) and route four is a loop.

This zero-one integer programming model would have to be solved twice. First, it would be solved for \(M\) equal to the minimum number of routes required to transport all the students of the school and then for \(M\) equal to one route plus the minimum number of routes required by
the school.

Since i may assume any of \((K+N-2)\) values, j any of \((N-1)\) values and m any of \(M\) values, there are \((K+N-2)(N-1)M\) variables, \(x(i,j,m)\), in this problem. Moreover, there are more than \(2N(N-2)!\) constraints. A normal bus routing problem for one school involving ten origins, twenty routes and one hundred bus stops would require 213840 variables and more than \(40(98)!\) constraints. Since the available integer programming algorithms are clearly unable to handle problems of this magnitude, a heuristic procedure is the only recourse for developing all the bus routes for a school district.
IV. METHOD OF SOLUTION

A. General Procedure

Bus routing for a school district is accomplished by suboptimization from period to period. Bus routing for a period is accomplished by suboptimization over the sets of routes developed for each of the schools serviced during the period. Quasi-optimal bus routing for an individual school is accomplished by an algorithm based selective enumeration procedure. A detailed description of each step of this enumeration procedure is given in the following sections. It can be summarized as follows:

1. Assuming that the maximum allowable number of routes will be used by all schools, determine for every school the number of bus routes which should start from each origin supplying buses for the period during which the school is being serviced so that the total estimated traveling time required by all the routes for the period is minimized.

2. For each school, select the location which serves as the origin for the greatest number of routes as determined in step one. This point will be called the "super-origin" for the school.

3. Determine by either using the Nearest City Approach or Algorithm A, a trial route which starts at the super-origin, visits every stop once and terminates at the school. The method used to develop this trial route is dependent upon the number of times step three has been executed.

4. Partition the single route determined in step three into
individual routes which satisfy bus capacity and passenger riding time constraints. The order of the stops determined in step three is preserved during the partitioning process.

5. If the set of routes contains no more than the maximum allowable number of routes, then proceed to step six. Otherwise, return to step three to generate a different trial route.

6. Improve each of the individual routes by Algorithm A, a modified traveling-salesman type algorithm. This step attempts to reduce the traveling time required by the individual route while preserving the assignment of stops to the route made by the partitioning procedure.

7. Determine the current best set of routes developed by the procedure.

8. If the set of routes obtained in step seven satisfies the acceptance criterion specified by the school, then proceed to step nine. Otherwise, repeat steps three through eight until either the acceptance criterion is satisfied or the algorithms used to generate the trial route in step three are exhausted, i.e. they are unable to generate another different trial route.

9. If the results of step one specify that all the routes for the school should start at the super-origin, then this acceptable set of routes is considered to be the quasi-optimum solution. Otherwise proceed to step ten.

10. Allocate the remaining origins specified by step one in such a way that the additional number of time units traveled
will be minimum. Then, this modified set of acceptable
routes is considered to be the quasi-optimum set of routes
for the school.

After bus routing has been completed, the schedule or timetable
for each individual route is calculated. Each schedule gives the bus
load and the arrival time at each stop serviced by the route. The
time required to load the bus at each stop and to unload the bus at
the school is not included in calculating the timetable. In addition,
a lower bound upon the total number of time units required to traverse
all the routes required by the school is calculated.

B. Selection of Bus Route Origins

At the beginning of the first bussing period all buses are
located at the garage(s) maintained by the school district. For all
other bussing periods the available buses are initially located at the
schools serviced during the previous period. It is assumed that the
number of buses available at the beginning of any bussing period is
adequate to meet the needs of all students to be transported during
the period, i.e. it is possible to service all the routes for the
period simultaneously. For each school $j$ serviced during the same
period, the determination of the number of bus routes which should
start at each origin $i$ can be expressed as a transportation problem.

Let

\[ n_1 = \text{number of origins supplying buses for the period} \]
\[ n_2 = \text{number of schools serviced during the period} \]
\[ x(i,j) = \text{number of routes starting at origin } i \text{ and ending} \]
\[ \text{at school } j \]
\[ c(i,j) = \text{average estimated time required to traverse any} \]
route starting at origin i and ending at school j

\[ A(i) = \text{number of buses available at origin } i \text{ at the} \]
\[ \text{beginning of the period} \]

\[ R(j) = \text{number of routes required by school } j \]

\[ B(j) = \text{bus capacity specified by school } j \]

The problem can be stated as follows:

Find variables \( x(i,j) \) for all combinations of \( i \) and \( j \), where \( i = 1, 2, \ldots, n_1 \) and \( j = 1, 2, \ldots, n_2 \), such that the objective function

\[
\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} [c(i,j)] [x(i,j)]
\]

is minimized and restrictions one through four, stated below, are satisfied. Optimization is with respect to minimizing the total estimated traveling time required by the \( n_2 \) sets of routes associated with the period.

The first constraint,

\[
\sum_{i=1}^{n_1} x(i,j) = R(j), \quad \text{for } j = 1, 2, \ldots, n_2,
\]

insures that all the routes required by school \( j \) are assigned to an origin. There will be \( n_2 \) constraints of this type.

The second constraint,

\[
\sum_{j=1}^{n_2} x(i,j) = A(i), \quad \text{for } i = 1, 2, \ldots, n_1,
\]

insures that that number of bus routes starting at origin \( i \) equals the number of buses available at origin \( i \). It is assumed that a bus services only one route for a school. There will be \( n_1 \) constraints
of this type.

The third constraint,

\[ x(i,j) \geq 0, \text{ an integer, for all } i \text{ and } j, \]

is self-explanatory.

The fourth constraint,

\[ \sum_{i=1}^{n_1} A(i) = \sum_{j=1}^{n_2} R(j) \]

insures that a feasible solution exists. It can be verified by observation. Since the first constraint implies \( \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} x(i,j) = \sum_{j=1}^{n_2} R(j) \) and the second constraint implies \( \sum_{j=1}^{n_2} \sum_{i=1}^{n_1} x(i,j) = \sum_{i=1}^{n_1} A(i) \), then indeed the fourth constraint must be satisfied.

Because it is assumed that \( \sum_{i=1}^{n_1} A(i) \geq \sum_{j=1}^{n_2} R(j) \) at the beginning of any bussing period, a fictitious school may have to be introduced to use the extra available buses in order to satisfy the fourth constraint.

The parameters \( n_1, n_2, A(i), \) and \( B(j) \) are given data for the problem. However, the parameters \( R(j) \) and \( c(i,j) \) for \( i = 1, 2, \ldots, n_1 \) and \( j = 1, 2, \ldots, n_2 \) must be calculated. The \( R(j) \) are determined as follows:

Let \( L(j,k) = \) number of students assigned to stop \( k \) associated with school \( j \)

\( n(j) = \) number of stops assigned to school \( j \)

(including the origin and terminus)
m(j) = minimum number of routes required to transport all students of school j

The values of L(j,k) and n(j) where j = 1,2,...,n^2 and k = 1,2,...,n(j) are given data for the problem.

\[ m(j) = \left\lceil \sum_{k=1}^{n(j)} \frac{L(j,k)}{\text{R}(j)} \right\rceil + 0.999999 \text{ truncated to the nearest integer.} \]

\[ \text{R}(j) = m(j) + 1, \text{ for } j = 1,2,...,n^2 \]

The parameter \( \text{R}(j) \) is used in the determination of the number of routes which should start at origin i for school j instead of m(j) because the bus capacity and student riding time constraints often make the generation of a set of feasible routes containing only m(j) routes unattainable.

The \( c(i,j) \) are calculated as follows:

Let \( \text{LB}(i,j) = \text{lower bound on the total traveling time required by } \text{R}(j) \text{ routes, all of which start at origin } i \text{ and end at school } j \)

\[ c(i,j) = \frac{\text{LB}(i,j)}{\text{R}(j)} \text{ for } i = 1,2,...,n_1 \text{ and } j = 1,2,...,n^2 \]

The \( \text{LB}(i,j) \) are calculated under the assumption that all the connections between origin i and the first stop of each of the \( \text{R}(j) \) routes, all the connections between the pairs of bus stops, and all the connections between the last stop of each of the \( \text{R}(j) \) routes and the terminus are made in the least time consuming manner. A computational procedure for determining the lower bound upon the total traveling time of a set of routes is described in section V-B.
The c(i,j) are based upon a lower bound instead of an upper bound because an attempt is being made to develop a set of routes whose total traveling time approaches this lower bound.

\[ c(i,j) = 0 \] when school j is a fictitious school.

The origin associated with the maximum \( x(i,j) \) for school j is defined to be the super-origin for school j. In case of a tie, the super-origin is selected arbitrarily.

The allocation of a number of routes for school j to origin i and the selection of a super-origin for each school serviced during the same period is determined once.

C. Determination of Trial Routes: Nearest-City Approach

A trial route which starts at the super-origin, visits every stop once, and terminates at the school, the route that an infinite capacity bus would traverse, is determined either by the Nearest City Approach or by Algorithm A discussed in the next section. The Nearest City Approach can be described as follows:

Let

- \( n \) = number of stops assigned to the school (including origin and terminus)
- \( \text{stop } 1 = \text{origin of route} \)
- \( \text{stop } n = \text{terminus or school} \)
- \( r = \text{number of the trial route being generated} \)

where \( r = 1, 2, \ldots, (n-2) \).

The subscript denoting the particular school has been omitted in order to simplify the nomenclature.

This trial route generated is one in which an "infinite capacity" bus starts at the super-origin, proceeds to stop \((r+1)\) and then
repeatedly selects as its next stop that point which is nearest to its present stop and which has not yet been serviced until it reaches the terminus or school. The current best trial route, the one requiring the least traveling time, is determined and saved each time that this process is executed. Since this procedure specifies the first bus stop visited, a different trial route is generated each time. If the acceptance criterion has not been satisfied after \((n-2)\) trial routes have been generated, then Algorithm A is used to develop succeeding trial routes from the best trial route previously determined.

D. Determination of Trial Routes: Algorithm A

Algorithm A, a systematic procedure for decreasing the total time required to traverse the infinite capacity bus route, is an extension of Algorithm 1 developed by the author. After the Nearest City Approach has been exhausted, the trial routes are generated by applying Algorithm A to the best infinite capacity bus route available. Initially Algorithm A is applied to the best route determined by the Nearest City Approach. Thereafter, Algorithm A is applied to the most recent trial route it generated.

This algorithm determines sets of three links which can be changed simultaneously without destroying the continuity of the tour, the non-coincidence of the origin and the terminus of the route, and the direction of the unchanged portions 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 the three new links is less than the time required to traverse the links which they replace, then the new route becomes the
next trial route and the best infinite capacity bus route available. If the proposed change results in no improvement, then another set of changes is determined and examined. Algorithm A is repeated until it is unable to improve the best trial route available, i.e. Algorithm A is exhausted.

The sets of three link changes in the current tour through a network of n points or stops in which the origin and terminus do not coincide are generated as follows for all combinations of i and j where 1 ≤ i ≤ (n-1), 2 ≤ j ≤ (n-1), stop 1 is the super-origin and stop n is the terminus or school:

1. New link 1 starts at point i and ends at point j where \( i \neq j \) and point \( j \neq \) the point which follows point i in the current tour.

2. New link 2 starts at point k and ends at the point which follows point i in the current route where \( k \neq n \). k is cycled as follows:

- point \( k_1 \) = point j
- point \( k_2 \) = point which follows point j in the current route
- point \( k_L \) = point which lies \( (L-1) \) consecutive positions after point j in the current route in a clockwise direction
- point \( k_m \) = point which precedes point i in the current route

In order to determine the complete range of values for index k associated with a particular i,j combination, it
is necessary to assume the existence of a fictitious link between the terminus and origin of the current route while counting the consecutive positions after point \( j \) in a clockwise direction.

3. New link 3 starts at the point which precedes point \( j \) in the current route and ends at the point which follows point \( k \) in the current route.

For illustrative purposes, Algorithm A will be applied to a route containing six points for one combination of \( i \) and \( j \) and the entire range of index \( k \) associated with it. Point one is the super-origin and point six is the terminus or school. The number associated with each node is the permanent identification number of the stop and corresponds to its position in the interstop travel time matrix, e.g., stop four data would form row four of the interstop travel time matrix.

Let \( i = 2 \) and \( j = 3 \) be the \( i,j \) combination

\[ R_0 = 1-4-2-5-3-6 \] be the current route
The first value assumed by k is the number of point j or three. The proposed route is $R_1 = 1-4-2-3-5-6$. Links 2-5, 5-3, and 3-6 are replaced by new links 2-3, 3-5, and 5-6, indicated by the dashed arcs.

The second value assumed by k is the number of the point which follows point j or six. However, no new route can be proposed by Algorithm A for this combination of i, j, and k because the value assigned to index k is the number associated with the terminus of the route. Since k has not yet assumed the value of the identification number of the point preceding point i in the current route, four, the cycle for index k is incomplete.

The third value assumed by k is the number of the point which lies two consecutive positions after point j, in a clockwise direction, or one. The proposed route is $R_2 = 1-5-4-2-3-6$. Links 1-4, 2-5, and 5-3 are replaced by new links 1-5, 5-4, and 2-3.
The fourth value of k is the number of the point which lies three consecutive positions, in a clockwise direction, after point j in the current route or four. This is the last k which can be generated for the combination i = 2 and j = 3 because the point numbered four precedes point i in the current route. The proposed route is \( R_3 = 1-4-5-2-3-6 \). Links 4-2, 2-5, and 5-3 are replaced by new links 4-5, 5-2, and 2-3.

When Algorithm A cannot improve the best trial route available, it is said to be exhausted. This best trial route which Algorithm A cannot improve is considered to be the quasi-optimal modified traveling-salesman route and the last possible trial route. A computational procedure for Algorithm A is described in section V-A.

E. Partitioning Procedure

The partitioning procedure is applied to every trial infinite capacity bus route determined by either the Nearest City Approach or Algorithm A. This procedure requires the following additional information:

1. a student load vector specifying the number of students
assigned to each stop,

2. the bus capacity, and

3. the maximum allowable riding time of the students picked up at the first stop of any route.

The partitioning procedure generates a set of bus routes each of which starts at the super-origin, visits the stops of the trial infinite capacity bus route in the order previously determined until the bus is loaded properly and proceeds to the terminus or 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 the traveling time from the previous stop. When either the bus capacity or the riding time constraint is about to be exceeded, the previous stop becomes the last one serviced by the bus before proceeding to the school. The next route starts at the super-origin and proceeds directly to that stop of the trial infinite capacity bus route which immediately follows the last stop serviced. All individual bus routes are determined in the same manner. The sequence of stops generated by either the Nearest City Approach or by Algorithm A is preserved throughout this procedure. If the set of routes determined by the partitioning procedure contains no more than the maximum allowable number of routes, then the improvement process, described in the next section, is applied to each route of the set. Otherwise, another trial infinite capacity bus route is generated by either the Nearest City Approach or Algorithm A.

F. Improvement Process

Each individual route belonging to a feasible set of routes
developed by the partitioning procedure is then improved by application of Algorithm A, previously described in section IV-D.

First, form a submatrix of the given interstop traveling time matrix consisting of those elements associated with the super-origin, the stops serviced by the individual route, and the terminus or school. The infinite capacity bus route corresponding to this submatrix is the individual route being improved. Then Algorithm A is applied to this route until it can make no further improvement. This two step procedure is repeated until all the individual bus routes of the feasible set have been improved. The given matrix of interstop travel times between each possible origin and every stop assigned to the school and the given matrix of interstop travel times for all pairs of stops except those links involving an origin are preserved at all times.

G. Acceptance Criteria

The best available set of routes, all of which start at the super-origin, for a particular school j will be considered to form the basis of the quasi-optimal solution if one of the following criteria is satisfied:

1. The total time required to traverse this set of routes is less than or equal to the product of a factor specified by the transportation director, greater than one, and the lower bound on the total time required to traverse the number of routes contained in the best available set of routes, assuming that all of them start at the super-origin.

2. The total number of trial routes generated is about to
exceed some number specified by the transportation director. Thus, the school administrator has an opportunity to specify the degree of optimality required for the set of routes accepted as a final solution. In the event that neither of the criteria are satisfied and no further trial infinite capacity bus routes can be developed because both the Nearest City Approach and Algorithm A have been exhausted, then the best available set of routes will become the basis of the quasi-optimal solution by default. The latter situation is one in which the capability of the method is unable to satisfy the requirements of the school.

H. Final Allocation of Origins to Individual Routes

The feasible set of individual bus routes which satisfy the acceptance criterion specified by school j all start at the super-origin. If all the routes for school j should start at the super-origin, as previously determined, then a final allocation of origins to individual routes is unnecessary. However, if all the routes for school j do not start at the super-origin, then the remaining origins are allocated in such a way that the additional number of time units traveled will be minimum. This final allocation problem can also be expressed as a transportation problem.

Let

- \( n_1 \) = number of origins supplying buses for the period
- \( n_3(j) \) = number of routes belonging to the feasible set of routes satisfying the acceptance criterion specified by school j
- \( x(i,j) \) = number of routes which should start at origin i and end at school j
d(i,j,k) = total traveling time of route k assigned to school j when it starts at origin i

\[ y(i,j,k) = \begin{cases} 
1 & \text{when route } k \text{ assigned to school } j \text{ starts at origin } i \\
0 & \text{otherwise}
\end{cases} \]

The problem can be stated as follows:

Find variables \( y(i,j,k) \) for a fixed \( j \) and all combinations of \( i \) and \( k \), where \( i = 1,2,...,n_1 \) and \( k = 1,2,...,n_3(j) \), such that the objective function

\[ \sum_{i=1}^{n_1} \sum_{k=1}^{n_3(j)} [d(i,j,k)] [y(i,j,k)] \]

is minimized and restrictions one through four, stated below, are satisfied. Optimization is with respect to minimizing the total traveling time required by the \( n_3(j) \) routes accepted by school \( j \).

The first constraint,

\[ \sum_{i=1}^{n_1} y(i,j,k) = 1, \text{ for } k = 1,2,...,n_3(j), \]

insures that a route is assigned to only one origin. There will be \( n_3(j) \) constraints of this type.

The second constraint,

\[ \sum_{k=1}^{n_3(j)} y(i,j,k) = x(i,j), \text{ for } i = 1,2,...,n_1 \]

insures that the number of bus routes starting at origin \( i \) equals...
the number of buses previously assigned to school \( j \) from origin \( i \).

There will be \( n_l \) constraints of this type.

The third constraint,

\[
y(i,j,k) = 0 \text{ or } 1, \quad \text{for all } i \text{ and } k,
\]
is self-explanatory.

The fourth constraint,

\[
\sum_{i=1}^{n_l} x(i,j) = n_3(j)
\]

insures that a feasible solution exists. Since the first constraint

\[
\sum_{i=1}^{n_l} \sum_{k=1}^{n_l} y(i,j,k) = \sum_{k=1}^{n_l} l = n_3(j) \quad \text{and the second}
\]

\[
\sum_{i=1}^{n_l} \sum_{k=1}^{n_l} y(i,j,k) = \sum_{i=1}^{n_l} x(i,j), \quad \text{then indeed}
\]

the fourth constraint must be satisfied. Since \( \sum_{i=1}^{n_l} x(i,j) \) equals the

maximum allowable number of routes and \( n_3(j) \leq \sum_{i=1}^{n_l} x(i,j) \), a

fictitious route may have to be introduced to use the extra available

bus in order to satisfy the fourth constraint.

The parameter \( n_l \) was given; the parameter \( n_3(j) \) was determined

by the partitioning procedure described in section IV-E. The values
of \(x(i,j)\), the number of bus routes which should start at origin \(i\) and end at school \(j\), were determined by solving the transportation problem described in section IV-B. The \(d(i,j,k)\) are calculated as follows:

Let

\[
t(s,j,k) = \text{total time required to traverse route } k \text{ when it starts at the super-origin and ends at school } j
\]

\[
l(s,j,k) = \text{traveling time between the super-origin and the first stop serviced by route } k \text{ associated with school } j
\]

\[
l(i,j,k) = \text{traveling time between origin } i \text{ and the first stop serviced by route } k \text{ associated with school } j
\]

\[
d(i,j,k) = t(s,j,k) - l(s,j,k) + l(i,j,k)
\]

\[d(i,j,k) = 0 \text{ when } k \text{ is a fictitious route.}\]

After the final allocation of origins to the individual bus routes for school \(j\) has been completed, a lower bound upon the total traveling time of the number of routes required by school \(j\) is calculated. This procedure is described in section V-B.
V. COMPUTATIONAL PROCEDURE

A. Algorithm A

A computational scheme for Algorithm A is described by the flow chart given in Figure 1. The nomenclature used in the flow chart is:

\[ N = \text{total number of bus stops (including the origin and the terminus)} \]

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

\[ M(I,J) = \text{number of time units required to travel from bus stop I to bus stop J: } 1 \leq I \leq N, 1 \leq J \leq N, I \neq J \]

\[ M(I,I) = \text{number of time units required to travel from bus stop I to the stop which immediately follows it in the current modified traveling-salesman route} \]

\[ M(I,N+1) = \text{identification number of the bus stop which immediately follows bus stop I in the current modified traveling-salesman route} \]

\[ M(N+1,J) = \text{identification number of the bus stop which immediately precedes bus stop J in the current modified traveling-salesman route} \]

\[ M(N+1,N+1) \text{ is not used} \]

\[ 01 = \text{origin of new link 1} \]

\[ T1 = \text{terminus of new link 1} \]

\[ 02 = \text{origin of new link 2} \]

\[ T2 = \text{terminus of new link 2} \]

\[ 03 = \text{origin of new link 3} \]
Fig. 1 ALGORITHM A

Entry:
1. I = 0
2. I = I + 1
3. J = 1
4. J = J + 1
5. TEST I = J
   T → 15
   F → 6
6. TEST M(I, N+1) = J
   T → 15
   F → 4
7. TEST J = N - 1
   T → 16
   F → 2
8. OLD = M(01, 01) + M(02, 02) + M(03, 03)
   NEW = M(01, T1) + M(02, T2) + M(03, T3)
9. TEST NEW < OLD
   T → 17
   F → 10
10. TEST O1 = T3
    T → 15
    F → 11
11. TEST T3 = N
    T → 8
    F → 12
12. TEST O1 = 1
    T → 8
    F → 15
13. O2 = T3
    T3 = M(02, N+1)
14. O2 = 1
    T3 = M(1, N+1)
15. EXIT ALGORITHM A HAS BEEN EXHAUSTED
16. EXIT EITHER TO COLUMNS OR PARTITIONING

Notes:
- 01 = I, T1 = J
- 02 = J, T2 = M(I, N+1)
- 03 = M(N+1, J), T3 = M(J, N+1)
T3 = terminus of new link 3

The first row of [M] is related to the origin and row N is related to the terminus of the modified traveling-salesman route.

Blocks 1-4 provide for the initialization and incrementation of index I and index J.

Blocks 5-6 prevent the creation of an illegal combination of I and J. Block 5 prevents a change in which indexes I and J are identical. Block 6 prevents a change in which stop J immediately follows stop I in the current tour and thereby eliminates the possibility of generating a new route that is identical to the old one.

Block 7 initializes the origin and terminus of each of the three new links associated with the current I, J combination.

Block 8 calculates the time required to traverse the three old links which are candidates for replacement and the time required to traverse the proposed new links.

Block 9 determines whether the proposed set of new links reduces total transit time.

Block 10 tests whether index k, the origin of new link 2, has assumed all possible values for the current combination of indexes I and J.

Block 11 tests whether the next value of index k would be the identification number assigned to the terminus of the current route.

Block 12 determines whether it would be possible to assign another value to index k by introducing a fictitious link between the terminus and origin of the current route.

Block 13 calculates the new value of index k, the origin of new
link 2 as described in Section IV-D, and the terminus of new link 3
when index k does not assume the value of the identification number of
the terminus of the current route.

Block 14 calculates the new value of index k, the origin of new
link 2, and the terminus of new link 3 after a fictitious link was
assumed to exist between the origin and terminus of the current route.

Blocks 15-16 determine whether indexes I and J can be further
updated. If I cannot be updated, then Algorithm A has been executed
for all possible combinations of I and J. If Algorithm A is exhausted
while it is being used to generate a trial infinite bus capacity route
prior to execution of the partitioning procedure, then the best avail-
able set of routes will have to be accepted as the quasi-optimal
solution for the school. Exit will be to the final allocation of
origins procedure. If Algorithm A is exhausted while it is being used
to improve an individual bus route, then exit is to the improvement of
the next route of the set.

Block 17 incorporates the three new links, that have been found to
reduce total transit time, into the current modified traveling-salesman
route 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. If Algorithm A produces an advantageous
change while it is being used to create a new trial infinite capacity
bus route, then exit is to the partitioning procedure. If Algorithm A
is being used to improve an individual bus route when an advantageous
change occurs, then Algorithm A is restarted and exit is to Block 1.
B. Lower Bound

The lower bound on the total time required to traverse the set of routes accepted as the quasi-optimal solution by the school is calculated under the assumption that all the connections between an origin and the first stop of any route, all the connections between the pairs of bus stops, and all the connections between the last stop of any route and the terminus are made in the least time consuming manner.

Let

\[ N = \text{total number of bus stops (including origin and terminus)} \]

\[ M = \text{rectangular matrix with (N-1) rows and N columns} \]

\[ M(I,J) = \text{number of time units required to travel from bus stop I to bus stop J: } 2 \leq I \leq N, 1 \leq J \leq N, I \neq J \]

\[ N_1 = \text{number of origins supplying buses for the school} \]

\[ M_1 = \text{rectangular matrix with } N_1 \text{ rows and } N \text{ columns} \]

\[ M_1(K,J) = \text{number of time units required to travel from origin } K \text{ to bus stop } J: 1 \leq K \leq N_1, 1 \leq J \leq N \]

\[ N_R(K) = \text{number of routes starting at origin } K: 1 \leq K \leq N_1 \]

\[ Q = \sum_{K=1}^{N_1} N_R(K) \]

\([M]\) is the given matrix of interstop travel times for all pairs of stops except those links involving an origin. Since the origin of a route is not allowed to be the terminus of a link, \(M(I,1) = \infty\) for \(2 \leq I \leq N\). Moreover, \(M(N,J) = \infty\) for \(1 \leq J \leq N\) because the terminus of a route is not allowed to be the origin of a link. These two restrictions are necessary because the method of solution is applicable to the general routing problem in which the origin and terminus do not coincide.
[M1] is the given matrix of interstop travel times between each origin servicing the period and every stop assigned to the school. Since no link is allowed between the origin and terminus of a route, 
\[ M_{1}(K,N) = \infty \text{ for } 1 \leq K \leq N1. \] Moreover, 
\[ M(I,I) = \infty \text{ for } 2 \leq I \leq N \] and 
\[ M_{1}(K,1) = \infty \text{ for } 1 \leq K \leq N1 \] because no loop is permitted at any bus stop.

Parameters N1 and NR(K) where 1 \leq K \leq N1 and NR(K) \neq 0 are determined by the final allocation of origins to routes procedure described in Section IV-H.

The lower bound is calculated as follows:

1. Form \([M2]\), a square matrix of order (Q+N-1), by the procedure described below.
2. Reduce \([M2]\) until there is at least one zero in every row and column. This is accomplished by subtracting the smallest element in each row from every element in the row, and then subtracting the smallest element in each column of the remaining matrix from every element in the column. The lower bound on the final set of routes is the total reduction or the sum of the elements subtracted from the rows and columns.

Starting at row one, \([M2]\) is formed as follows:

1. For each K, 1 \leq K \leq N1, row K of \([M1]\) is stored NR(K) times in \([M2]\), i.e. each route starting at origin K contributes one row to \([M2]\). This rectangular submatrix consisting of Q rows and N columns occupies the upper left portion of 
\([M2]\). During the reduction process, this submatrix provides
that portion of the lower bound contributed by traveling between any origin and the first stop of a route.

2. $[M]$ is stored in rows $(Q+1)$ through $(Q+N-1)$ of $[M2]$.

During the reduction process the first $(N-1)$ columns of this submatrix provide that part of the lower bound contributed by traveling between any pair of points.

3. Steps one and two create columns one through $N$ of $[M2]$. Column $N$ is then stored $(Q-1)$ more times in $[M2]$, i.e. columns $N$ through $(Q+N-1)$ of $[M2]$ are the same vector whose elements represent the time required to travel between any bus stop and the terminus. During the reduction process, the submatrix occupying rows one through $(Q+N-1)$ and columns $N$ through $(Q+N-1)$ of $[M2]$ provides that portion of the lower bound contributed by traveling between the last stop of any route and the terminus.

When the lower bound has to be calculated for a set of routes, all of which start at the same origin, then $N1 = 1$ and $NR(1) = Q$ equals the number of routes contained in the set for this procedure.
VI. EVALUATION

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 the problem being considered at a cost commensurate with the value received.

A. Model and Method

The model developed in this dissertation is a general one which utilizes variables that are applicable to all school systems. Moreover, it requires few assumptions that simplify the real world environment. The method used to obtain a solution for the model reduces the routing of buses for an entire school district to a set of sequential steps that can be readily programmed for a digital computer. Furthermore, this set of sequential steps is arranged into groups called iterations or passes such that a "reasonably good" feasible routing system can be obtained after a few iterations have been completed. Thus, the user of the model and method can specify the degree of optimality desired for the routing system being developed.

The procedure used to develop a set of bus routes for each school is logical. At the beginning of a period, the available buses at the origins are allocated to every school serviced during the period so that the total estimated traveling time required by all the routes developed for the period is minimized. This is accomplished by solving a transportation problem in which the elements of the requirements vector are the maximum allowable number of buses required by a school, the
elements of the availability vector are the number of buses located at each origin at the beginning of the period, and the elements of the cost matrix are based upon the lower bound on the total traveling time required by each combination of origin and school. Since the bus capacity and riding time constraints often make the generation of a feasible set of routes containing the absolute minimum number of routes unattainable, the selection of this requirement vector is considered reasonable. Moreover, because an attempt is being made to develop a set of routes whose total traveling time approaches the lower bound, the estimation of the elements of the cost matrix used by the method is a logical choice.

The determination of a trial route which starts at the "super-origin", visits every stop once and terminates at the school by either the Nearest City Approach or Algorithm A tends to group stops that are located in the same neighborhood. Since a school usually services an area of less than twenty square miles, the variability associated with the magnitude of the elements of the interstop travel time matrix is low. Thus, arranging the bus stops to be serviced into groups located in the same neighborhood is logical. The first \((n-2)\) different trial routes, where \(n\) is the total number of stops assigned to the school including the origin and the terminus, are developed by the Nearest City Approach because it is a rapid and relatively efficient process. After all trial routes are generated by the Nearest City Approach, the route requiring the least traveling time is saved in order to reduce the number of trial routes which Algorithm A will generate before it is exhausted and to insure that Algorithm A does not create one of the routes previously developed.

51
Algorithm A is applied to the best available trial route because a good "infinite capacity" bus route usually partitions into a good routing system even though the best "infinite capacity" bus route does not necessarily generate the best set of individual routes. Although Algorithm A is a slightly slower process than the Nearest City Approach, it guarantees that the next trial route developed will require less traveling time. Both of the algorithms were selected because they are rapid, efficient, and exhausted in a finite number of steps.

The partitioning procedure quickly generates routes which not only satisfy the bus capacity and passenger riding time constraints but also service stops in the same general area. If the set of routes developed contains at most one more than the absolute minimum number of routes required by the school, then each route of the set is improved by application of Algorithm A until it is exhausted while preserving the assignment of stops to the route made by the partitioning process. This improvement procedure may reduce the traveling time required by the individual route and thus tends to reduce the total transit time of the routing system.

Minimizing the number of routes included in a routing system insures that the buses are being utilized to full capacity and also tends to reduce the total traveling time required by the set of routes by eliminating some of the links between an origin and the first stop of any route and the links between the last stop of any route and the terminus. Since at most ten students are usually assigned to a bus stop in order to reduce the noise level and the possibility of landscape damage in residential areas, satisfying the maximum allowable number of routes in a routing system constraint presents no problem. However, if
a transportation director decides to lump the bus stops extensively so as to reduce the size of the interstop travel time matrix or if the population density is such that many students are assigned to one stop, then determining a feasible routing system containing at most one more than the absolute minimum number of routes may be unattainable. The user of this method must then adjust the set of constraints by either increasing the maximum allowable number of routes in the routing system or by reducing the bus capacity. Another alternative is the addition of bus stops to reduce the number of students assigned to individual stops.

The specification of a criterion for accepting a set of routes gives the transportation director the opportunity to select the degree of optimality under which the routing system will be developed. This acceptance criterion is applied to a routing system when all the routes start at the "super-origin" because allocating an origin to each route every time a feasible routing system is generated would require more computer time than is warranted by the improvement which would be realized.

The final allocation of origins to individual routes, if necessary, is accomplished by solving a transportation problem in which the elements of the availability vector are the number of routes which start at each origin as determined by step one of the solution procedure and the elements of the requirements vector are all equal to one. Each element of the cost matrix is the total traveling time required by a route when it starts at a particular origin. Optimization is with respect to minimizing the total traveling time required by the final set of routes for the school. Initially, the solution procedure develops routing
systems in which all the individual routes start at the "super-origin" in order to minimize the effects of this final allocation of origins to routes. After an origin has been assigned to each route of the set, no attempt is made to further reduce the transit time required by each route because the improvement process may create a violation of the student riding time constraint.

Although the final lower bound on the set of routes considered to be the quasi-optimal routing system does not include the effects of the bus capacity and riding time constraints, it does include a partial effect of the number of routes included in the system. However, it provides some measure for assessing the quality of the set of routes developed even though it may be unattainable.

Thus, this heuristic procedure is considered to have a sound basis.

B. Computational Experience

Since there is no known bus routing method suitable for use on a computer which will guarantee an optimum solution, any heuristic procedure must be judged not only with respect to its degree of success relative to the best set of routes available for known problems but also with respect to its consistency in the level of success.

A computer program based upon the procedure described in this dissertation was written in FORTRAN IV for use on the CDC 6400 computer. It can handle bus routing for a school district involving any number of periods, nine route origins and nine schools per period; one hundred twenty stops per school, thirty-four routes per school, and thirty stops per route. All computations are done in integer arithmetic.

Computational experience was gained in three phases. First,
Newton's procedure was applied to school bus routing and delivery problems reported in the literature. Then, the relationship between the amount of computer time required to develop a routing system and the number of stops serviced by the set of routes was examined. Finally, this method was used to develop a set of bus routes for four schools in the Williamsville Central School District, a suburban area in Western New York.

Thompson drew a map of a hypothetical school system showing the location of the school, the location of 31 bus stops specified by the school, and the number of students assigned to each stop. Moreover, all roads were marked off in units of one-half mile. In order to simulate the area of a real school system as closely as possible, the map included features such as: isolated areas, contour roads, and varying population densities. The elements of the symmetric interstop distance matrix calculated from this map ranged between 1 and 13.5 miles; the elements of the student load vector varied between 3 and 22 students. Although it was stated that 3 minutes were required to travel one mile and that loading at each bus stop required a minute, no restriction was imposed on the student riding time. Buses of 30, 36, 42, 48, 54, 60, 66, and 72 passenger capacities were available. The problem was to design the set of bus routes, all of which started and ended at the school, required to transport the 252 students assigned to the 31 bus stops.

The group of 49 school superintendents with experience in school bus routing and the group of 37 transportation directors who participated in Thompson's study were allowed to design the routing system for the hypothetical school by any manual method they favored.
The best set of routes developed by a member of the group of transportation directors involved 5 routes whose loads varied between 28 and 67 students and whose total traveling distance was 100 miles. The best set of routes designed by a member of the group of school superintendents consisted of 5 routes whose loads ranged between 33 and 67 students and whose total traveling distance was 101 miles.

Newton's method was applied to the same hypothetical school for all combinations of 4 bus capacities and 4 maximum allowable riding distances or a total of 16 cases. Each case was run until Algorithm A was exhausted at iteration or pass 49, i.e., the Nearest City Approach generated 31 trial routes and Algorithm A generated 17 "infinite capacity" bus routes. 177.133 seconds of computer time were used to develop the routing systems for the 16 cases or approximately 11 seconds per case. Most of the solutions were accepted from one of the last 3 trial routes generated by Algorithm A as was expected. A summary of the results appears in Table 1.

Case 13 defined by a bus capacity of 54 students and a maximum riding distance of 20 miles yielded the best routing system whose total traveling distance was 94.5 miles. This represents a savings of 5.8% miles with respect to the best set of routes developed by a transportation director participating in Thompson's study. The best set of routes for Case 13 is:
THOMPSON DATA

31 STOPS SERVICED
252 STUDENTS TRANSPORTED
APPROXIMATELY 11 SECONDS CDC 6400 COMPUTER TIME PER CASE

<table>
<thead>
<tr>
<th>CASE</th>
<th>CAPACITY OF BUS</th>
<th>MAXIMUM RIDING DISTANCE</th>
<th>MINIMUM NUMBER ROUTES</th>
<th>NUMBER ROUTES USED</th>
<th>DISTANCE ROUTE SYSTEM</th>
<th>RATIO</th>
<th>NUMBER ANSWER PASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>72</td>
<td>20.0 mi.</td>
<td>4</td>
<td>5</td>
<td>97.0 mi.</td>
<td>1.48</td>
<td>44</td>
</tr>
<tr>
<td>2</td>
<td>72</td>
<td>18.0</td>
<td>4</td>
<td>5</td>
<td>97.0</td>
<td>1.48</td>
<td>47</td>
</tr>
<tr>
<td>3</td>
<td>72</td>
<td>17.5</td>
<td>4</td>
<td>5</td>
<td>95.5</td>
<td>1.46</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>72</td>
<td>17.0</td>
<td>4</td>
<td>5</td>
<td>95.5</td>
<td>1.46</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>66</td>
<td>20.0</td>
<td>4</td>
<td>5</td>
<td>97.0</td>
<td>1.48</td>
<td>44</td>
</tr>
<tr>
<td>6</td>
<td>66</td>
<td>18.0</td>
<td>4</td>
<td>5</td>
<td>95.5</td>
<td>1.46</td>
<td>47</td>
</tr>
<tr>
<td>7</td>
<td>66</td>
<td>17.5</td>
<td>4</td>
<td>5</td>
<td>95.5</td>
<td>1.46</td>
<td>48</td>
</tr>
<tr>
<td>8</td>
<td>66</td>
<td>17.0</td>
<td>4</td>
<td>5</td>
<td>95.5</td>
<td>1.46</td>
<td>47</td>
</tr>
<tr>
<td>9</td>
<td>60</td>
<td>20.0</td>
<td>5</td>
<td>5</td>
<td>95.5</td>
<td>1.46</td>
<td>47</td>
</tr>
<tr>
<td>10</td>
<td>60</td>
<td>18.0</td>
<td>5</td>
<td>5</td>
<td>95.5</td>
<td>1.46</td>
<td>47</td>
</tr>
<tr>
<td>11</td>
<td>60</td>
<td>17.5</td>
<td>5</td>
<td>5</td>
<td>103.5</td>
<td>1.54</td>
<td>47</td>
</tr>
<tr>
<td>12</td>
<td>60</td>
<td>17.0</td>
<td>5</td>
<td>6</td>
<td>106.0</td>
<td>1.58</td>
<td>46</td>
</tr>
<tr>
<td>13</td>
<td>54</td>
<td>20.0</td>
<td>5</td>
<td>5</td>
<td>94.5</td>
<td>1.44</td>
<td>44</td>
</tr>
<tr>
<td>14</td>
<td>54</td>
<td>18.0</td>
<td>5</td>
<td>6</td>
<td>102.0</td>
<td>1.52</td>
<td>47</td>
</tr>
<tr>
<td>15</td>
<td>54</td>
<td>17.5</td>
<td>5</td>
<td>6</td>
<td>102.0</td>
<td>1.52</td>
<td>47</td>
</tr>
<tr>
<td>16</td>
<td>54</td>
<td>17.0</td>
<td>5</td>
<td>6</td>
<td>105.5</td>
<td>1.57</td>
<td>47</td>
</tr>
</tbody>
</table>

Table 1
Cases 3, 4, 7, 8, 9 and 10 produced routing systems whose total traveling distance was 95.5 miles. However, all the routing systems were not identical. Cases 3, 4, and 7 resulted in the same routing system; case 8 yielded a second set of routes; cases 9 and 10 produced a third routing system. Thus, by developing routing systems for various combinations of bus capacity and passenger riding distance constraints and a fixed interstop distance matrix it may be possible to produce alternative routing systems whose total traveling distance is identical. A routing system can then be selected from these sets upon the basis of either bus load or individual route length variability or some other statistic considered important by the school transportation personnel. Studying alternate optimal solutions can be easily accomplished by using this computational procedure because of its speed and efficiency.

Boyer designed a set of bus routes, all of which start and end at the school, to transport 575 students assigned to 45 bus stops for a school in Hennepin County, Minnesota. The problem involved a symmetric interstop travel time matrix, a bus capacity constraint of 65 students, and a riding time constraint of infinity. Since Boyer's method does not require knowledge of all the elements of the interstop travel time matrix, only 171 elements were listed in (6). The elements of the
student load vector ranged between 1 and 33 students; the known elements of the interstop travel time matrix varied between 1 and 23 minutes. Although the absolute minimum number of buses needed to provide transportation for the school was 9, Boyer developed a routing system using 11 buses and requiring 413 minutes traveling time.

Newton's procedure was applied to the same problems until Algorithm A was exhausted. At iteration 51, i.e. the Nearest City Approach generated 45 trial routes and Algorithm A generated 5 trial routes. A routing system using 10 buses whose total traveling time was 394 minutes with a lower bound of 221 minutes was developed in 13.392 seconds. The quasi-optimal routing system, obtained from iteration 49 of this procedure, requires 4.8% fewer minutes than Boyer's solution. The author feels that a better solution would have been obtained with the complete interstop travel time matrix. The best set of routes is:

<table>
<thead>
<tr>
<th>Number</th>
<th>Route</th>
<th>Load</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S-22-23-S</td>
<td>60</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>S-24-25-26-31-21-20-19-S</td>
<td>65</td>
<td>38</td>
</tr>
<tr>
<td>3</td>
<td>S-37-41-36-35-S</td>
<td>65</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>S-40-39-34-38-S</td>
<td>60</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>S-33-32-29-30-27-28-12-S</td>
<td>57</td>
<td>62</td>
</tr>
<tr>
<td>6</td>
<td>S-6-13-11-5-1-1-S</td>
<td>63</td>
<td>49</td>
</tr>
<tr>
<td>7</td>
<td>S-2-3-4-7-9-8-S</td>
<td>56</td>
<td>48</td>
</tr>
<tr>
<td>8</td>
<td>S-10-43-42-S</td>
<td>62</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>S-44-45-18-17-S</td>
<td>61</td>
<td>36</td>
</tr>
<tr>
<td>10</td>
<td>S-16-15-14-S</td>
<td>26</td>
<td>36</td>
</tr>
</tbody>
</table>

Thus, Newton's method was able to obtain better routing systems for the only two school bus routing problems which have appeared in the
literature. The results for Thompson's data were especially encouraging because experienced transportation directors can usually develop almost optimum routing systems for problems involving 20-30 stops by visual trial-and-error adjustment.

The rest of the problems from the literature which were solved by this computational procedure are delivery problems. Although, the school bus scheduling problem and the delivery problem are conceptually the same, differences which exist in them must be considered when evaluating a heuristic procedure.

The school bus scheduling problem is characterized by a large number of stops which must be serviced and a non-symmetric interstop travel time matrix for reasons such as: restrictions imposed upon the crossing of busy streets by students, one-way streets and limited access highways. The elements of the interstop time matrix usually have a narrow range and a low variability because of the relatively small area serviced by a school and the restrictions imposed upon student walking time to the bus stop. The elements of the student load vector also have a narrow range and a low variability because of upper bounds usually placed upon the number of students assigned to a bus stop to reduce the noise level and the possibility of landscape damage in residential areas. Moreover, the bus stops are usually arranged in groups because of housing developments and the placement of bus stops along main thoroughfares in sparsely populated areas. Even in sparsely populated areas, no bus stop is really isolated, i.e. each stop is a relatively short distance from either the school or any other stop designated by the school. Moreover, any computer based procedure for the school bus scheduling problem must require little computer time in order to be of
practical value to a school district because of the frequent updating of routing systems necessitated by population fluctuations, changes in school boundaries, and the building of new schools and lack of funds.

The delivery problem is characterized by relatively few stops and a symmetric interstop distance matrix whose elements usually have a wide range and high variability because of the large area serviced by a warehouse or depot. Since no restrictions are placed upon customer demands, the elements of the customer demand vector also have a wide range and high variability. Moreover, the delivery stops are not necessarily arranged in groups because customer demand is not area dependent. Then too, any computer based procedure for the delivery problem may require a great deal of computer time and still be acceptable to a corporation which normally allocates ample funds for the development of routing systems.

These differences between the delivery problem and the school bus scheduling problem are great enough to make the widespread interchange of heuristic solution procedures infeasible. Therefore, an efficient computational procedure which was designed primarily to handle routing problems possessing the characteristics of the school bus scheduling problem is not expected to be consistently superior with respect to the quality of the routing systems developed when applied to delivery problems.

Dantzig and Ramser\textsuperscript{14} developed a set of truck routes, all of which start and end at the depot, to deliver 18200 gallons of material to 12 customers in 6000 gallon capacity trucks. The elements of the symmetric interstop distance matrix varied between 5 and 52 units; the elements of the customer demand vector ranged between 1100 and 1900 gallons.
The Dantzig and Ramser method produced a routing system using 4 routes whose total traveling distance was 294 units.

Clarke and Wright\(^9\) developed a routing system for the same problem requiring 4 routes covering 290 units, the conjectured optimum.

Newton's procedure was applied to the same delivery problem until Algorithm A was exhausted at iteration 13. A routing system using 4 routes whose total traveling distance was 304 distance units was developed in 1.117 seconds. The solution, obtained from iteration 5 of this procedure, requires 3.4% more distance units than the Dantzig and Ramser solution and 4.8% more distance units than the conjectured optimum.

The set of routes developed by this procedure is:

<table>
<thead>
<tr>
<th>Number</th>
<th>Route</th>
<th>Load</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-6-7-5-0</td>
<td>4300</td>
<td>64</td>
</tr>
<tr>
<td>2</td>
<td>0-9-8-10-0</td>
<td>5300</td>
<td>92</td>
</tr>
<tr>
<td>3</td>
<td>0-11-12-4-3-0</td>
<td>5700</td>
<td>120</td>
</tr>
<tr>
<td>4</td>
<td>0-2-1-0</td>
<td>2900</td>
<td>28</td>
</tr>
</tbody>
</table>

Clarke and Wright\(^9\) designed a set of truck routes, all of which start and end at the depot, to deliver 104,300 pounds of goods to 30 customers in 14,000 pound capacity trucks. The elements of the symmetric interstop distance matrix ranged between 3 and 98 miles; the elements of the customer demand vector varied between 100 and 12,300 pounds. The Clarke and Wright solution\(^9\) used 8 trucks, the absolute minimum number of trucks possible, and required 1427 miles. For the same problem, the Dantzig and Ramser method\(^14\) produced a routing system using 10 routes whose total traveling distance was 1766 miles. By
visual trial-and-error adjustment, Gaskell\textsuperscript{18}, developed a routing system for this problem using 8 routes and covering 1416 miles.

Newton's procedure was applied to the same problem until Algorithm A was exhausted at iteration 44. A routing system using 9 routes whose total traveling time was 1544 miles was developed in 7.948 seconds. The solution, obtained from iteration 5, required 9% more miles than the Gaskell solution, 8.2% more miles than the Clarke and Wright method, and 14.3% fewer miles than the Dantzig and Ramser solution.

The set of routes developed by this method is:

<table>
<thead>
<tr>
<th>Number</th>
<th>Route</th>
<th>Load</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-6-5-11-16-15-9-7-13-29-0</td>
<td>13700</td>
<td>215</td>
</tr>
<tr>
<td>2</td>
<td>0-12-14-4-3-24-23-22-0</td>
<td>13500</td>
<td>207</td>
</tr>
<tr>
<td>3</td>
<td>0-27-26-0</td>
<td>11900</td>
<td>199</td>
</tr>
<tr>
<td>4</td>
<td>0-8-10-19-0</td>
<td>8700</td>
<td>176</td>
</tr>
<tr>
<td>5</td>
<td>0-18-25-20-0</td>
<td>12200</td>
<td>150</td>
</tr>
<tr>
<td>6</td>
<td>0-2-1-21-17-0</td>
<td>8500</td>
<td>111</td>
</tr>
<tr>
<td>7</td>
<td>0-30-0</td>
<td>12300</td>
<td>136</td>
</tr>
<tr>
<td>8</td>
<td>0-28-0</td>
<td>9500</td>
<td>186</td>
</tr>
<tr>
<td>9</td>
<td>0-19-0</td>
<td>14000</td>
<td>164</td>
</tr>
</tbody>
</table>

Cochran\textsuperscript{10} designed the routing systems for two delivery problems using a modified Clarke and Wright method. Problem 1 involved 14,461 units to be delivered to 12 customers in trucks of 4500 unit capacity. The elements of the symmetric interstop distance matrix varied between 8 and 315 units; the elements of the customer demand vector ranged between 100 and 3726 units. The Cochran solution\textsuperscript{10} used 4 routes whose total traveling distance was 1433 units.

Newton's method was applied to Problem 1 until Algorithm A was
exhausted at iteration 19. A routing system containing 4 routes whose total traveling distance was 1383 units was developed in 1.717 seconds. The solution, obtained from iteration 18 of this procedure, required 3.6% fewer distance units than the solution produced by the modified Clarke and Wright method. Although the relative locations of the delivery stops are unavailable, the author feels that this computational procedure produced a superior solution because the delivery stops were arranged in groups and thus Problem 1 resembled a school bus scheduling problem.

The set of routes developed by this method is:

<table>
<thead>
<tr>
<th>Number</th>
<th>Route</th>
<th>Load</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-7-8-5-0</td>
<td>3290</td>
<td>185</td>
</tr>
<tr>
<td>2</td>
<td>0-14-0</td>
<td>3726</td>
<td>444</td>
</tr>
<tr>
<td>3</td>
<td>0-6-12-13-11-10-0</td>
<td>3745</td>
<td>478</td>
</tr>
<tr>
<td>4</td>
<td>0-4-3-2-9-0</td>
<td>3700</td>
<td>276</td>
</tr>
</tbody>
</table>

The first two routes appeared in both routing systems.

Cochran's Problem 2 involved 1405 units of goods to be delivered to 25 customers in 120 unit capacity trucks. The elements of the symmetric interstop distance matrix varied between 2 and 221 distance units; the elements of the customer demand vector ranged between 15 and 100 units. Using a modified Clarke and Wright method, Cochran designed a routing system using 14 trucks whose total traveling distance was 1468 distance units.

This computational procedure was applied to Problem 2 until Algorithm A was exhausted at iteration 31. A routing system using 14 trucks whose total traveling distance was 1486 distance units was
developed in 3.637 seconds. The solution, obtained from iteration 12 of this procedure, travels 1.2% more distance units than the solution obtained by the modified Clarke and Wright method. Both solutions used 2 more trucks than the absolute minimum number of trucks possible. This was due to the combination of customer demand loads, most of which were greater than 50 demand units, and the location of the customers, i.e. customers in the same general area had total demands which exceeded the truck capacity and thus one route could service only one or two stops.

The set of routes developed by this method is:

<table>
<thead>
<tr>
<th>Number</th>
<th>Route</th>
<th>Load</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-13-11-0</td>
<td>90</td>
<td>82</td>
</tr>
<tr>
<td>2</td>
<td>0-7-15-0</td>
<td>120</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>0-16-14-0</td>
<td>120</td>
<td>102</td>
</tr>
<tr>
<td>4</td>
<td>0-2-0</td>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0-3-0</td>
<td>80</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>0-4-9-0</td>
<td>110</td>
<td>56</td>
</tr>
<tr>
<td>7</td>
<td>0-5-0</td>
<td>90</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>0-8-6-0</td>
<td>115</td>
<td>76</td>
</tr>
<tr>
<td>9</td>
<td>0-10-0</td>
<td>60</td>
<td>48</td>
</tr>
<tr>
<td>10</td>
<td>0-12-0</td>
<td>90</td>
<td>54</td>
</tr>
<tr>
<td>11</td>
<td>0-17-18-0</td>
<td>120</td>
<td>185</td>
</tr>
<tr>
<td>12</td>
<td>0-19-20-21-0</td>
<td>110</td>
<td>210</td>
</tr>
<tr>
<td>13</td>
<td>0-24-25-0</td>
<td>120</td>
<td>276</td>
</tr>
<tr>
<td>14</td>
<td>0-23-22-26-0</td>
<td>120</td>
<td>285</td>
</tr>
</tbody>
</table>

Routes 1, 5, 6, 7, 10, 11, 12, 13, and 14 appeared in both routing systems. If routes 4 and 9 are merged into one route 0-2-10-0 which requires 52 distance units, then the routing system will use one less truck. However, the total traveling distance of the routing system with the merged routes in this case happens to remain unchanged. Although
the purpose of a computer based routing procedure is to avoid visual trial-and-error adjustments which become less efficient and highly impractical as problem size increases, this merging of two routes is noted as a point of interest.

Gaskell created four new delivery problems which were constrained by both truck capacity and route length. Total route distance included not only the number of units covered in traveling between delivery stops but also an allowance of 10 miles for each customer serviced. These problems were designed to compare the efficiency of variations of the Clarke and Wright method with respect to groupings of stops and isolated customers.

Routing systems for each of these delivery problems were developed by the Visual Method, a combination of trial-and-error adjustment and manual permutation of groups of delivery stops. The routing systems produced by the Visual Method were the best available and therefore were used to evaluate the routing systems developed by the computer based variations of the Clarke and Wright method. Since these problems involved at most 32 delivery stops, the time consuming Visual Method produced optimal routing systems. However, for larger problems it would be unable to compete with computer based procedures. Thus, the routing systems developed by the computational procedure of this dissertation will be compared primarily with the routing systems produced by variations of the Clarke and Wright method, a computer based procedure.

The first problem, Case Study Number 3, involved 29,370 units to be delivered to 32 customers in 8000 unit trucks. Each route was not to exceed 240 miles including the mileage allowance of 10 miles per customer. The elements of the interstop distance matrix varied between

66
1 and 118 miles; the elements of the customer demand vector ranged between 40 and 4000 units. This case is characterized by a close grouping of some of the customers, a centrally located depot with respect to 30 of the delivery stops and 2 customers located a great distance from the depot.

The Visual Method yielded a routing system using 4 routes whose total traveling distance was 813 miles. The best solution produced by a variation of the Clarke and Wright method used 5 routes and covered 821 miles. However, the poorest solution produced by a variation of the Clarke and Wright method used 5 routes and required 850 miles.

Newton's procedure was applied to Case Study Number 3 until Algorithm A was exhausted at iteration 39. It produced a routing system using 5 routes whose total traveling distance was 886 miles. The solution, obtained from iteration 23 of the procedure, required 7.9% more miles than the best solution obtained by a variation of the Clarke and Wright method. The reason that this computational procedure designed a less desirable routing system was the presence of the isolated delivery stops which are generally not present in the school bus scheduling problem.

The set of routes developed by this method is:

<table>
<thead>
<tr>
<th>Number</th>
<th>Route</th>
<th>Load</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-17-24-23-22-20-21-18-19-15-14-0</td>
<td>6900</td>
<td>228</td>
</tr>
<tr>
<td>2</td>
<td>0-1-11-5-6-7-8-9-10-32-13-0</td>
<td>7920</td>
<td>177</td>
</tr>
<tr>
<td>3</td>
<td>0-31-30-3-4-2-12-0</td>
<td>6350</td>
<td>222</td>
</tr>
<tr>
<td>4</td>
<td>0-29-28-27-26-25-0</td>
<td>7500</td>
<td>147</td>
</tr>
<tr>
<td>5</td>
<td>0-16-0</td>
<td>700</td>
<td>112</td>
</tr>
</tbody>
</table>
The second problem, Case Study Number 4, involved 22,500 units to be delivered to 21 customers in 6000 unit trucks. Each route was to be less than 200 miles in length including the mileage allowance of 10 miles per customer. The elements of the interstop distance matrix varied between 3 and 83 miles; the elements of the customer demand vector ranged between 100 and 2500 units. None of the customers are isolated from either the depot or from other customers. The variability of the elements of the interstop distance matrix is relatively low.

The Visual Method produced a routing system using 4 routes whose total traveling distance was 585 miles. The best solution produced by a variation of the Clarke and Wright method used 4 routes and covered 598 miles. However, the poorest solution produced by a variation of the Clarke and Wright method used 4 routes and required 648 miles.

Newton's procedure was applied to Case Study Number 4 until Algorithm A was exhausted at iteration 25. A routing system using 4 routes whose total traveling distance was 593 miles was developed. The solution, obtained from iteration 22 of this procedure, required 1% fewer miles than the best and 9% fewer miles than the poorest routing systems designed by variations of the Clarke and Wright method.

The set of routes developed by this method is:

<table>
<thead>
<tr>
<th>Route</th>
<th>Load</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-6-1-2-5-7-9-0</td>
<td>5600</td>
<td>173</td>
</tr>
<tr>
<td>0-10-8-3-4-11-13-0</td>
<td>5400</td>
<td>162</td>
</tr>
<tr>
<td>0-12-15-18-16-14-0</td>
<td>5500</td>
<td>126</td>
</tr>
<tr>
<td>0-17-20-21-19-0</td>
<td>6000</td>
<td>132</td>
</tr>
</tbody>
</table>

Routes 1 and 2 also appear in the routing system developed by the Visual
Method. Since none of the customers were isolated and the variability of the elements of the interstop distance was low, Case Study Number 4 possessed two of the characteristics of the school bus scheduling problem and this computational procedure developed a superior routing system.

The third problem, Case Study Number 5, involved 12,750 units of goods to be delivered to 29 customers in 4500 unit capacity trucks. Each route was not to exceed 240 miles including the mileage allowance of 10 miles per delivery stop. The elements of the interstop distance matrix ranged between 1 and 121 miles; the elements of the customer demand vector varied between 100 and 3100 units. This problem was characterized by a loose grouping of customers located at various distances from the depot. Gaskell considered this to be a difficult problem.

The Visual Method designed a routing system containing 4 routes whose total traveling distance was 876 miles. The best solution produced by a variation of the Clarke and Wright method used 5 routes and required 943 miles. However, the poorest routing system developed by a variation of the Clarke and Wright method involved 5 routes covering 1017 miles.

Newton's method was applied to Case Study Number 5 until Algorithm AVAS exhausted at iteration 33. A routing system using 5 routes whose total traveling distance was 913 miles was developed. The solution, obtained from iteration 30 of this procedure, required 3.3% fewer miles than the best and 11.3% fewer miles than the poorest routing systems designed by variations of the Clarke and Wright method.
The set of routes developed by this method is:

<table>
<thead>
<tr>
<th>Number</th>
<th>Route</th>
<th>Load</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-21-14-8-9-17-12-11-10-28-18-0</td>
<td>4125</td>
<td>220</td>
</tr>
<tr>
<td>2</td>
<td>0-15-16-7-13-0</td>
<td>1000</td>
<td>156</td>
</tr>
<tr>
<td>3</td>
<td>0-26-28-27-25-24-29-0</td>
<td>2850</td>
<td>234</td>
</tr>
<tr>
<td>4</td>
<td>0-3-6-1-4-5-2-0</td>
<td>3975</td>
<td>216</td>
</tr>
<tr>
<td>5</td>
<td>0-22-20-19-0</td>
<td>800</td>
<td>87</td>
</tr>
</tbody>
</table>

Again, this computational method produced a superior routing system because the customers were arranged loosely in groups, a characteristic of the school bus scheduling problem.

The fourth problem, Case Study Number 6, involved 10,189 units of goods to be delivered to 22 customers in 4500 unit capacity trucks. Each route was not to exceed a length of 240 miles including the mileage allowance of 10 miles per delivery. The elements of the interstop distance matrix ranged between 4 and 145 miles; the elements of the customer demand vector varied between 60 and 4100 units. Some of the customers are loosely arranged in groups with the distance between customers greater than the distance between neighboring stops in other problems. One customer is isolated at a relatively great distance from the depot.

The Visual Method produced a routing system involving 5 routes whose total traveling distance was 949 miles. The best solution produced by a variation of the Clarke and Wright method used 5 routes and required 955 miles. However, the poorest solution produced by a variation of the Clarke and Wright method required 6 routes covering 1015 miles.

Newton's procedure was applied to Case Study Number 6 until Algorithm A was exhausted at iteration 25. A routing system involving 6
routes whose total traveling time was 1009 miles was developed. The solution, obtained from iteration 14 of this procedure, requires 5.7% more miles than the best solution produced by a variation of the Clarke and Wright method.

The set of routes developed by this method is:

<table>
<thead>
<tr>
<th>Number</th>
<th>Route</th>
<th>Load</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-14-17-15-16-3-2-0</td>
<td>1144</td>
<td>227</td>
</tr>
<tr>
<td>2</td>
<td>0-11-13-6-1-0</td>
<td>775</td>
<td>156</td>
</tr>
<tr>
<td>3</td>
<td>0-10-0</td>
<td>4100</td>
<td>78</td>
</tr>
<tr>
<td>4</td>
<td>0-12-9-5-4-8-7-0</td>
<td>2700</td>
<td>200</td>
</tr>
<tr>
<td>5</td>
<td>0-18-19-22-20-0</td>
<td>1295</td>
<td>216</td>
</tr>
<tr>
<td>6</td>
<td>0-21-0</td>
<td>175</td>
<td>132</td>
</tr>
</tbody>
</table>

Routes 2, 3, and 6 appeared in both this routing system and the one developed by the Visual Method. 17,406 seconds of computer time were used to solve the last four problems. Again, as a point of interest, if routes 2 and 6 are merged to form the route 0-21-11-13-6-1-0 which requires 275 miles, then the routing system will contain one less route and the total traveling distance will be reduced to 996 miles.

Newton's procedure, developed primarily to handle the school bus scheduling problem, behaved as was to be expected when applied to eight delivery problems from the literature. When the delivery problem possessed some of the characteristics of the school bus scheduling problem, this computational method produced a superior routing system. On the other hand, when the delivery problem involved isolated customers and many customers who were not even loosely grouped in areas, then this computational method produced a less desirable routing system than the other computer based methods. However, the number of extra miles
required by the routing systems developed by this computational procedure never exceeded the best solution generated by a variation of the Clarke and Wright method by more than 8.2% which occurred in the Clarke and Wright 30 stop problem. Thus, from a practical standpoint, this computational procedure performed at an acceptable level even when applied to delivery problems possessing characteristics which the method was not designed to handle.

Since the ratio of the total traveling time or distance required by the set of routes developed divided by the lower bound on the total traveling time or distance is used not only to assess the efficiency of the routing system but to form a criterion for the acceptance or rejection of a set of routes as the quasi-optimal solution for the problem being considered, an examination of this ratio with respect to the problems from the literature is warranted.

A summary of the results obtained by applying this computational procedure to two school bus scheduling and eight delivery problems from the literature is given in Table 2. The column labeled % gives the percentage by which the method of this dissertation either exceeded or improved the total number of time/distance units required by the best routing system developed by a computer based procedure for each problem being considered. In the case of the Thompson-31 stop problem, routing systems developed by manual methods were the only ones available.

By observation, the ratio of the total traveling time/distance for a routing system divided by the lower bound on the total traveling time/distance yields little information about the relative efficiency of a routing system developed by this computational procedure in
### SUMMARY OF RESULTS FOR PROBLEMS IN THE LITERATURE

<table>
<thead>
<tr>
<th>PROBLEM NAME</th>
<th>TOTAL TIME/DISTANCE</th>
<th>ROUTE SYSTEM</th>
<th>LOWER BOUND</th>
<th>RATIO</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>THOMPSON-31 STOPS</td>
<td>94.5</td>
<td>65.5</td>
<td>1.44</td>
<td>-5.8</td>
<td></td>
</tr>
<tr>
<td>BOYER-45 STOPS</td>
<td>394.0</td>
<td>221.0</td>
<td>1.78</td>
<td>-4.8</td>
<td></td>
</tr>
<tr>
<td>DANTZIG &amp; RAMSER-12 STOPS</td>
<td>304.0</td>
<td>145.0</td>
<td>2.10</td>
<td>+4.8</td>
<td></td>
</tr>
<tr>
<td>CLARKE &amp; WRIGHT-30 STOPS</td>
<td>1544.0</td>
<td>868.0</td>
<td>1.78</td>
<td>+8.2</td>
<td></td>
</tr>
<tr>
<td>COCHRAN-12 STOPS</td>
<td>1383.0</td>
<td>816.0</td>
<td>1.69</td>
<td>-3.6</td>
<td></td>
</tr>
<tr>
<td>COCHRAN-25 STOPS</td>
<td>1486.0</td>
<td>383.0</td>
<td>3.88</td>
<td>+1.2</td>
<td></td>
</tr>
<tr>
<td>GASKELL #3-32 STOPS</td>
<td>886.0</td>
<td>620.0</td>
<td>1.43</td>
<td>+7.9</td>
<td></td>
</tr>
<tr>
<td>GASKELL #4-21 STOPS</td>
<td>593.0</td>
<td>432.0</td>
<td>1.38</td>
<td>-1.0</td>
<td></td>
</tr>
<tr>
<td>GASKELL #5-29 STOPS</td>
<td>913.0</td>
<td>590.0</td>
<td>1.55</td>
<td>-3.3</td>
<td></td>
</tr>
<tr>
<td>GASKELL #6-22 STOPS</td>
<td>1009.0</td>
<td>626.0</td>
<td>1.61</td>
<td>+5.7</td>
<td></td>
</tr>
</tbody>
</table>

Table 2
comparison with one generated for the same problem by another computer
based method or the conjectured optimum routing system attainable.

The ratio seems to be sensitive to the range and variability of
the elements of the interstop travel time/distance matrix and the
grouping of stops as evidenced by the Thompson-31 stop problem \(^{34}\) and
the Gaskell \#4-21 stop problem \(^{18}\). Both of these problems had inter-
stop travel time/distance matrices whose elements had a narrow range
and low variability and the stops were arranged in groups. Moreover,
both of these problems also had low ratios.

This ratio appears to reflect the effect of using more than the
absolute minimum number of routes required as evidenced by the Cochran-
25 stop problem \(^{10}\). The routing system developed for the problem con-
tained two more than the absolute minimum number of routes required and
the ratio was 3.88.

The routing system developed by Newton's procedure for the Dantzig
and Ramser-12 stop problem \(^{14}\) traveled 4.8% more distance units than
the conjectured optimal routing system whose total traveling distance
was 290 distance units and the ratio was 2.10. However, the ratio of
the total traveling distance for the conjectured optimal routing system
divided by the lower bound on the total distance equals 2.0.

Although the ratio is a relatively poor predictor of the degree of
optimality attained by a routing system developed by any method, it can
still be used as a criterion for terminating this computational pro-
cedure. Knowledge of reasonable ratios which can be expected for a
routing system developed for a school will be acquired after experience
with this procedure has been gained.

Although this computational procedure has proved to be efficient
with respect to the quality of routing systems developed and with respect to the amount of computer time used to solve ten problems from the literature, no problem involved more than 45 stops, the Boyer data. A thorough evaluation of a heuristic procedure requires that its efficiency be also examined for problems of the size which it would be expected to handle in the real world.

Thus, 18 problems involving up to 120 bus stops were created from tables of random numbers to examine the computer time required to develop routing systems for problems involving various numbers of stops. The elements of the interstop travel time matrices ranged between 2 and 20 minutes; the elements of the student load vector varied between 1 and 9. These ranges of values were selected because they satisfied the characteristics associated with the school bus scheduling problem.

All problems were constrained by a student riding time of 45 minutes and all problems were run until Algorithm A was exhausted, i.e. no further trial routes could be generated. A summary of the results appears in Table 3.

By observation, it appears that the lower bus capacity for a problem involving the same number of bus stops requires slightly less computer time. This is due to the fact that the individual routes contain fewer bus stops than the routes for buses of larger capacity and the amount of time required to improve each individual route is decreased.

Moreover, it seems that the quasi-optimal routing system is usually selected from an iteration whose trial route was generated by Algorithm A. This evidence also appeared when routing systems were developed for ten problems from the literature and thus bears out the
<table>
<thead>
<tr>
<th>NUMBER OF STOPS</th>
<th>CAPACITY OF BUS</th>
<th>NUMBER OF PUPILS</th>
<th>MINIMUM NUMBER ROUTES</th>
<th>NUMBER ROUTES USED</th>
<th>TIME ROUTE SYSTEM</th>
<th>TIME LOWER BOUND</th>
<th>RATIO</th>
<th>NUMBER LAST PASS</th>
<th>NUMBER SOLUTION PASS</th>
<th>COMPUTER TIME USED (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>60</td>
<td>208</td>
<td>4</td>
<td>4</td>
<td>143</td>
<td>98</td>
<td>1.46</td>
<td>47</td>
<td>40</td>
<td>24.455</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
<td>208</td>
<td>5</td>
<td>5</td>
<td>156</td>
<td>104</td>
<td>1.50</td>
<td>47</td>
<td>40</td>
<td>21.132</td>
</tr>
<tr>
<td>50</td>
<td>60</td>
<td>268</td>
<td>5</td>
<td>5</td>
<td>159</td>
<td>114</td>
<td>1.39</td>
<td>57</td>
<td>56</td>
<td>36.140</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>268</td>
<td>6</td>
<td>6</td>
<td>184</td>
<td>121</td>
<td>1.52</td>
<td>57</td>
<td>50</td>
<td>31.607</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>324</td>
<td>6</td>
<td>6</td>
<td>196</td>
<td>132</td>
<td>1.48</td>
<td>69</td>
<td>61</td>
<td>50.787</td>
</tr>
<tr>
<td>60</td>
<td>50</td>
<td>324</td>
<td>7</td>
<td>7</td>
<td>212</td>
<td>138</td>
<td>1.54</td>
<td>69</td>
<td>62</td>
<td>43.488</td>
</tr>
<tr>
<td>70</td>
<td>60</td>
<td>391</td>
<td>7</td>
<td>7</td>
<td>233</td>
<td>157</td>
<td>1.48</td>
<td>78</td>
<td>72</td>
<td>71.811</td>
</tr>
<tr>
<td>70</td>
<td>50</td>
<td>391</td>
<td>8</td>
<td>9</td>
<td>265</td>
<td>169</td>
<td>1.57</td>
<td>78</td>
<td>43</td>
<td>64.760</td>
</tr>
<tr>
<td>80</td>
<td>60</td>
<td>457</td>
<td>8</td>
<td>8</td>
<td>254</td>
<td>178</td>
<td>1.43</td>
<td>89</td>
<td>87</td>
<td>138.999</td>
</tr>
<tr>
<td>80</td>
<td>50</td>
<td>457</td>
<td>10</td>
<td>10</td>
<td>292</td>
<td>190</td>
<td>1.54</td>
<td>89</td>
<td>84</td>
<td>128.463</td>
</tr>
<tr>
<td>90</td>
<td>60</td>
<td>521</td>
<td>9</td>
<td>9</td>
<td>323</td>
<td>211</td>
<td>1.53</td>
<td>108</td>
<td>106</td>
<td>225.716</td>
</tr>
<tr>
<td>90</td>
<td>50</td>
<td>521</td>
<td>11</td>
<td>11</td>
<td>347</td>
<td>223</td>
<td>1.56</td>
<td>108</td>
<td>106</td>
<td>213.029</td>
</tr>
<tr>
<td>100</td>
<td>60</td>
<td>587</td>
<td>11</td>
<td>11</td>
<td>330</td>
<td>210</td>
<td>1.57</td>
<td>118</td>
<td>110</td>
<td>218.701</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>587</td>
<td>12</td>
<td>13</td>
<td>369</td>
<td>216</td>
<td>1.71</td>
<td>118</td>
<td>117</td>
<td>199.408</td>
</tr>
<tr>
<td>110</td>
<td>60</td>
<td>657</td>
<td>12</td>
<td>12</td>
<td>396</td>
<td>278</td>
<td>1.42</td>
<td>126</td>
<td>118</td>
<td>300.670</td>
</tr>
<tr>
<td>110</td>
<td>50</td>
<td>657</td>
<td>14</td>
<td>14</td>
<td>426</td>
<td>290</td>
<td>1.47</td>
<td>126</td>
<td>123</td>
<td>289.035</td>
</tr>
<tr>
<td>120</td>
<td>60</td>
<td>709</td>
<td>12</td>
<td>13</td>
<td>414</td>
<td>290</td>
<td>1.43</td>
<td>136</td>
<td>133</td>
<td>433.106</td>
</tr>
<tr>
<td>120</td>
<td>50</td>
<td>709</td>
<td>15</td>
<td>15</td>
<td>470</td>
<td>300</td>
<td>1.57</td>
<td>136</td>
<td>131</td>
<td>416.499</td>
</tr>
</tbody>
</table>

Table 3
conjecture that better trial routes tend to partition into better sets of routes.

Since, this computational procedure was able to develop routing systems for problems involving 120 stops in approximately seven minutes, it is considered to be efficient with respect to computer time usage and therefore would be of practical value to a school district.

Finally, to examine the worth of Newton's procedure with respect to a real world situation, it was applied to four schools in the Williamsville Central School District, a rapidly growing suburban area in Western New York.

This district maintains a fleet of 78 buses which service ten elementary schools, two middle schools and three secondary schools located within the 42 square mile area of the district and twenty private and special schools outside the district. Furthermore, it is anticipated that ten new schools will be added to the school system within the next five years. Therefore, the Williamsville Central School District expects to be continually faced with a complex school bus routing problem and expressed a great interest in using a practical computer based method for designing its school bus routes.

The transportation director and the assistant to the superintendent of schools requested that Newton's procedure be applied to Academy Elementary School, Forest Elementary School, South Senior High School, and Dodge Elementary School. The first three schools service the established, densely populated part of Williamsville. Dodge Elementary School is located in the new, sparsely populated section of the area.

For each school, the school administrators designated the bus stops on a large map of the area and assigned students to the stops
from census tracts. Each bus stop was assigned two labels, a road code number and a map number.

The interstop travel distance matrices were then developed manually by using a map reader. Two people developed the matrices for the four schools in approximately forty hours. The elements of the interstop distance matrix were then multiplied by a time factor, specified by the school administrators, to convert the distance matrix into an interstop travel time matrix. This time factor was large enough to include the time spent in servicing a stop. Since the school administrators felt that variability in travel time was negligible when considering all the routing systems collectively, no attempt was made to include the effects of this variability.

The elements of the interstop travel time matrix for the Academy School varied between 0.5 and 14 minutes or 0.25 and 7 distance units; the elements of the student load vector ranged between 2 and 44. The large elements of the student load vector were due to the high density of the population in the area serviced by the Academy School and the lumping of points by the school administrators. The results of applying this computational procedure to the Academy School data is summarized in Table 4. The routing system developed by this heuristic procedure requires 40% fewer distance units and 1 bus less than the current routing system used by the school.

The elements of the interstop travel time matrix for the Forest School ranged between 0.5 and 14 minutes or 0.25 and 7 distance units; the elements of the student load vector varied between 1 and 30 students. The Forest School is located in a high population density area. The results of applying Newton's method to the Forest School
ACADEMY SCHOOL

TYPE OF AREA  - densely populated
NUMBER OF STOPS  - 45 (including origin and terminus)
NUMBER OF STUDENTS  - 532
NUMBER OF ROUTE ORIGINS  - 1 (South Senior High School)
MAXIMUM RIDING TIME  - 45 minutes

<table>
<thead>
<tr>
<th>CAPACITY OF BUS</th>
<th>MINIMUM NUMBER ROUTES</th>
<th>NUMBER OF ROUTES USED NEWTON SCHOOL</th>
<th>DISTANCE UNITS ROUTE SYSTEM NEWTON SCHOOL</th>
<th>DISTANCE LOWER BOUND</th>
<th>RATIO</th>
<th>NUMBER LAST PASS</th>
<th>NUMBER ANSWER PASS</th>
<th>COMPUTER TIME USED (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>8</td>
<td>9</td>
<td>61.00</td>
<td>28.50</td>
<td>2.14</td>
<td>64</td>
<td>63</td>
<td>39.990</td>
</tr>
<tr>
<td>67</td>
<td>8</td>
<td>9</td>
<td>63.75</td>
<td>28.50</td>
<td>2.24</td>
<td>64</td>
<td>61</td>
<td>36.482</td>
</tr>
<tr>
<td>62</td>
<td>9</td>
<td>10</td>
<td>67.00</td>
<td>28.75</td>
<td>2.33</td>
<td>64</td>
<td>44</td>
<td>38.496</td>
</tr>
<tr>
<td>57</td>
<td>10</td>
<td>11</td>
<td>80.75</td>
<td>29.00</td>
<td>2.78</td>
<td>64</td>
<td>3</td>
<td>31.740</td>
</tr>
<tr>
<td>52</td>
<td>11</td>
<td>12</td>
<td>77.50</td>
<td>29.25</td>
<td>2.65</td>
<td>64</td>
<td>60</td>
<td>31.815</td>
</tr>
</tbody>
</table>

1 distance unit = 0.4444 mile

Algorithm A was exhausted

Table 4
data is summarized in Table 5. The routing system developed by this computational procedure requires 38% fewer distance units than the current routing system used by the school. However, both routing systems use 10 buses when the bus capacity is 67 students. Because of the lumping of bus stops, the routing systems required more than one extra bus over the absolute minimum number of buses required when the bus capacity was reduced to 57 and 52.

The elements of the interstop travel time matrix for the South Senior High School varied between 0.5 and 22 minutes or 0.25 and 11 distance units; the elements of the student load vector ranged between 1 and 41 students. These large student load elements were due to lumping of bus stops by the school administrators. The results of applying this computational procedure to the South Senior High School data is summarized in Table 6. The routing system developed by Newton's procedure requires 25% fewer distance units and one less bus than the current routing system used by the school.

The elements of the interstop travel time matrix for the Dodge School varied between 0.5 minute and 35 minutes or 0.25 and 17.5 distance units; the elements of the student load vector ranged between 1 and 27 students. All the routes for the Academy School started from the South Senior High School, all the routes for the Forest School started from the Academy School and all the routes for the South Senior High School started from the garage. However, the bus routes for the Dodge School can start at any of six origins. The results of applying Newton's procedure to the Dodge School data is summarized in Table 7. The routing system used by this heuristic procedure requires 17% fewer distance units and two less buses than the current routing.
**FOREST SCHOOL**

**TYPE OF AREA**  - densely populated  
**NUMBER OF STOPS**  - 37 (including origin and terminus)  
**NUMBER OF STUDENTS**  - 596  
**NUMBER OF ROUTE ORIGINS**  - 1 (Academy School)  
**MAXIMUM RIDING TIME**  - 45 minutes

<table>
<thead>
<tr>
<th>CAPACITY OF BUS</th>
<th>MINIMUM NUMBER ROUTES</th>
<th>NUMBER OF ROUTES USED</th>
<th>DISTANCE UNITS ROUTE SYSTEM NEWTON SCHOOL</th>
<th>DISTANCE LOWER BOUND</th>
<th>RATIO</th>
<th>NUMBER LAST PASS</th>
<th>NUMBER ANSWER PASS</th>
<th>COMPUTER TIME USED (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>9</td>
<td>9</td>
<td>-</td>
<td>57.00</td>
<td>-</td>
<td>26.75</td>
<td>2.13</td>
<td>45</td>
</tr>
<tr>
<td>67</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>63.25</td>
<td>87.0</td>
<td>27.75</td>
<td>2.28</td>
<td>45</td>
</tr>
<tr>
<td>62</td>
<td>10</td>
<td>11</td>
<td>-</td>
<td>66.00</td>
<td>-</td>
<td>28.75</td>
<td>2.30</td>
<td>45</td>
</tr>
<tr>
<td>57</td>
<td>11</td>
<td>13</td>
<td>-</td>
<td>72.75</td>
<td>-</td>
<td>30.75</td>
<td>2.37</td>
<td>45</td>
</tr>
<tr>
<td>52</td>
<td>12</td>
<td>15</td>
<td>-</td>
<td>84.00</td>
<td>-</td>
<td>32.75</td>
<td>2.56</td>
<td>45</td>
</tr>
</tbody>
</table>

1 distance unit = 0.4444 mile

Algorithm A was exhausted

Table 5
SOUTH SENIOR HIGH SCHOOL

TYPE OF AREA - densely populated
NUMBER OF STOPS - 76 (including origin and terminus)
NUMBER OF STUDENTS - 1097
NUMBER OF ROUTE ORIGINS - 1 (garage - Mill Middle School)
MAXIMUM RIDING TIME - 45 minutes

<table>
<thead>
<tr>
<th>CAPACITY OF BUS</th>
<th>MINIMUM NUMBER ROUTES</th>
<th>NUMBER OF ROUTES USED NEWTON SCHOOL</th>
<th>DISTANCE UNITS ROUTE SYSTEM NEWTON SCHOOL</th>
<th>DISTANCE LOWER BOUND</th>
<th>RATIO</th>
<th>NUMBER LAST PASS</th>
<th>NUMBER ANSWER PASS</th>
<th>COMPUTER TIME USED (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>17</td>
<td>19</td>
<td>20</td>
<td>175.0</td>
<td>219.5</td>
<td>72.25</td>
<td>2.42</td>
<td>97</td>
</tr>
<tr>
<td>60</td>
<td>19</td>
<td>21</td>
<td>-</td>
<td>189.0</td>
<td>-</td>
<td>75.25</td>
<td>2.51</td>
<td>97</td>
</tr>
<tr>
<td>55</td>
<td>20</td>
<td>24</td>
<td>-</td>
<td>211.5</td>
<td>-</td>
<td>79.75</td>
<td>2.65</td>
<td>97</td>
</tr>
<tr>
<td>50</td>
<td>22</td>
<td>28</td>
<td>-</td>
<td>233.5</td>
<td>-</td>
<td>85.75</td>
<td>2.72</td>
<td>97</td>
</tr>
<tr>
<td>45</td>
<td>25</td>
<td>30</td>
<td>-</td>
<td>251.0</td>
<td>-</td>
<td>88.75</td>
<td>2.83</td>
<td>97</td>
</tr>
</tbody>
</table>

1 distance unit = 0.4444 mile

Algorithm A was exhausted

Table 6
### Table 7

<table>
<thead>
<tr>
<th>CAPACITY OF BUS</th>
<th>MINIMUM NUMBER ROUTES</th>
<th>NUMBER OF ROUTES USED NEWTON SCHOOL</th>
<th>DISTANCE UNITS ROUTE SYSTEM NEWTON SCHOOL</th>
<th>DISTANCE LOWER BOUND</th>
<th>RATIO</th>
<th>NUMBER LAST PASS</th>
<th>NUMBER ANSWER PASS</th>
<th>COMPUTER TIME USED (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>10</td>
<td>10</td>
<td>196.50</td>
<td>107.5</td>
<td>1.81</td>
<td>100</td>
<td>100</td>
<td>90.999</td>
</tr>
<tr>
<td>72</td>
<td>10</td>
<td>10</td>
<td>196.50</td>
<td>107.5</td>
<td>1.81</td>
<td>105</td>
<td>102</td>
<td>148.213</td>
</tr>
<tr>
<td>72</td>
<td>10</td>
<td>10</td>
<td>196.50</td>
<td>107.5</td>
<td>1.81</td>
<td>110</td>
<td>102</td>
<td>243.488</td>
</tr>
<tr>
<td>67</td>
<td>10</td>
<td>11</td>
<td>204.00</td>
<td>116.5</td>
<td>1.78</td>
<td>100</td>
<td>82</td>
<td>60.989</td>
</tr>
<tr>
<td>67</td>
<td>10</td>
<td>11</td>
<td>204.00</td>
<td>116.5</td>
<td>1.78</td>
<td>105</td>
<td>82</td>
<td>116.821</td>
</tr>
<tr>
<td>66</td>
<td>10</td>
<td>11</td>
<td>204.00</td>
<td>116.5</td>
<td>1.78</td>
<td>110</td>
<td>82</td>
<td>211.202</td>
</tr>
<tr>
<td>62</td>
<td>11</td>
<td>12</td>
<td>221.50</td>
<td>121.5</td>
<td>1.82</td>
<td>100</td>
<td>31</td>
<td>70.240</td>
</tr>
<tr>
<td>62</td>
<td>11</td>
<td>12</td>
<td>221.00</td>
<td>121.5</td>
<td>1.32</td>
<td>105</td>
<td>102</td>
<td>126.550</td>
</tr>
<tr>
<td>62</td>
<td>11</td>
<td>12</td>
<td>221.00</td>
<td>121.5</td>
<td>1.82</td>
<td>110</td>
<td>102</td>
<td>218.009</td>
</tr>
<tr>
<td>57</td>
<td>12</td>
<td>13</td>
<td>239.25</td>
<td>131.0</td>
<td>1.83</td>
<td>100</td>
<td>86</td>
<td>63.323</td>
</tr>
<tr>
<td>57</td>
<td>12</td>
<td>13</td>
<td>238.00</td>
<td>131.0</td>
<td>1.82</td>
<td>105</td>
<td>101</td>
<td>119.392</td>
</tr>
<tr>
<td>57</td>
<td>12</td>
<td>13</td>
<td>234.25</td>
<td>131.0</td>
<td>1.79</td>
<td>110</td>
<td>109</td>
<td>213.476</td>
</tr>
</tbody>
</table>

1 distance unit = 0.444 mile

Algorithm A was not exhausted
system used by the school. Since this problem involved 96 stops, a study was made to examine the effect of limitations upon the number of trial routes generated. By observation, it can be noted in Table 7 that increasing the number of iterations allowed for each case used considerably more computer time but did not usually reduce the total number of distance units required by the routing system.

The application of Newton's computational procedure to four actual school bussing problems demonstrated its effectiveness. It was able to develop routing systems which were superior to the routing systems currently used by these schools with respect to the distance traveled and the number of buses used. Moreover, it was able to produce these high quality routing systems using an operationally acceptable amount of computer time.
VII. SUMMARY AND RECOMMENDATIONS

The model developed in this dissertation is a general one which utilizes variables that are applicable to all school systems. It represents a significant improvement over the manual and other computer based methods available in that it generates routing systems which are efficient with respect to the total mileage traveled and the number of routes required using a minimal amount of computer time. Moreover, it routes buses from school-to-school in addition to developing the individual routes for a school.

A review of the work accomplished during this study leads to the following recommendations for future efforts in the automatic design of school bus routes:

1. The manual calculation of the elements of the interstop distance matrix is tedious, error prone, and time consuming. Therefore, a great need exists for a computer based procedure for developing an accurate, non-symmetric interstop distance matrix involving 50-120 bus stops located in either densely or sparsely populated areas.

2. A computer based procedure for determining a student load vector whose elements could be constrained by student walking distance and size of load at a stop would be useful.

3. In order to avoid the development of a routing system involving a combination of full bus loads and half filled buses and a combination of maximum length and very short routes, a need exists for a method which would balance the
sizes of the bus loads and the lengths of the individual routes.

4. Since government aid is usually given to school districts for every bus load of a certain size, it would be useful to be able to specify both upper and lower limits on bus capacity.

5. School districts are interested in the cost associated with a transportation system. Therefore, it would be useful to be able to develop routing systems with respect to minimizing a cost function which would include the effects of the total mileage traveled and the size of the buses used.

6. A great need exists for a computer based procedure which would optimally assign all the routes which a particular bus would service between the time it left the garage and returned.
VIII. BIBLIOGRAPHY


87


APPENDIX

The Appendix contains a listing of Program BUS2 which generates bus routes and schedules for a multi-school system by means of the method described in this dissertation. Program BUS2 is written in FORTRAN IV for the CDC 6400 computer.
C PROGRAM BUS2 - RITA NEWTON

C SELECT THE ORIGIN AND THE STOPS TO BE VISITED FOR EACH BUS ROUTE

DIMENSION IALPH(30), IARRT(30), ILOAD(30), INUM(30), IRI(120)

DIMENSION IREFQ(35), IRTE(120), IRT(30), IRTF2(30), IRTF3(30), IRTP(240)

DIMENSION IVAIL(10), KDF(120), KL(35), KONRT(35), KORA(35)

DIMENSION KORN(35), KOST(10,35), KRT(35), KT(35), LAB(120)

DIMENSION LBA(10,35), LB01(10), LB02(10)

DIMENSION LIST(30), LOAD(120), LOR(10), LSCH(6)

C*** VECTOR LSCH CONTAINS 6 TEN-CHARACTER WORDS
C*** ALTRM DIMENSION OF VECTOR LSCH AND ALL READ/WRITE
C*** STATEMENTS INVOLVING VECTOR LSCH ACCORDING TO THE
C*** WORD LENGTH OF THE COMPUTER BEING USED
C***
DIMENSION M(121,121), M12(3,31), MAXBUS(35), MINRT(155)

DIMENSION MOR(10,120), NRAV(10), NIBUS(10,35), NUS(10,35)

DIMENSION NUS2(10), NSTOP1(35), NSUPER(10)

C ARRAYS MUST BE DIMENSIONED AS FOLLOWS

C LET A = NUMBER OF STOPS ON INDIVIDUAL ROUTE (INCLUDES OR. + TER.)

C R = NUMBER OF ROUTES FOR SCHOOL + 1

C C = NUMBER OF ORIGINS FOR PERIOD + 1

C N = NUMBER OF SCHOOLS SERVICED DURING PERIOD

C F = TOTAL NUMBER OF STOPS FOR SCHOOL (INCLUDES OR. + TER.)

C IALPH(A), IARRT(A), ILOAD(A), INUM(A), IRI(F), IREFQ(B), IRTE(F)

C IRTE2(A), IRTE3(A), IRTP(F+F), IVAIL(C), KDF(E), KL(B), KONRT(B)

C KORN(A), KOST(C,R), KRT(A), KT(R), LAB(E), LBA(C,R)

C LAB(1), LB02(C), LIST(A), LOAD(E), LOR(C), LRA(E)

C M2(A+1), A+1, MAXBUS(A), MINRT(E+F), MOR(C,E), NUS(C,B)

C PREPARE TAPE OR DISC A TO STORE SCHOOL DATA

1 REWIND 2

READ LABEL FOR TIME PERIOD IDENTIFICATION

READ (5, 999)

999 FORMAT (72H)

1
C KSS = 1 MEANS THE TIME MATRIX IS NOT SYMMETRIC
C KPER = ALLOWABLE PERCENTAGE ABOVE LOWER BOUND FOR A SET OF ROUTES
C KPASS = MAXIMUM NUMBER OF TRIAL ROUTES ALLOWED
21 READ ( 5, 1002 ) ( LSCH(I), I = 1, 6 )
1002 FORMAT ( 6X, 6A10 )
READ ( 5, 1000 ) N, KS, KSS, KPER, KPASS
NP1 = N+1
NM1 = N-1
C CLEAR MATRIX MOR
C MOR(I,J) = NUMBER OF MINUTES REQUIRED TO TRAVEL FROM ORIGIN I
C TO BUS STOP J
DO 24 I = 1, NOR
DO 23 J = 1, N
MOR(I,J) = 0
23 CONTINUE
24 CONTINUE
DO 25 I = 1, NOR
READ ( 5, 1000 ) ( MOR(I,J), J = 1, N )
25 CONTINUE
C CLEAR MATRIX M
C M(I,J) = NUMBER OF MINUTES REQUIRED TO TRAVEL FROM
C BUS STOP I TO BUS STOP J
DO 28 I = 1, N
DO 27 J = 1, N
M(I,J) = 0
27 CONTINUE
28 CONTINUE
TEST = MATRIX M IS SYMMETRIC
IF ( KSS ) 39, 39, 49
C READ LOWER HALF OF SYMMETRIC TIME MATRIX M
C ROWS 2 - N BECAUSE ROW 1 WILL BE SELECTED FROM MATRIX MOR
C ROW N IS REALLY COLUMN N BECAUSE NO TRAVEL IS ALLOWED
C BETWEEN THE TERMINUS OR SCHOOL AND ANY BUS STOP
30 DO 40 I = 2, N
READ (5, 1000) (M(I,J), J = 1, I)
40 CONTINUE
C
C  GENERATE THE UPPER HALF OF SYMMETRIC MATRIX M
C  DO 44 I = 1, N, 1
C  K = I + 1
C  DO 43 J = K, N
C  M(I,J) = M(J,I)
43 CONTINUE
C
C  CONTINUE
C  CONTINUE
C  CONTINUE
C  CONTINUE
C  GO TO 52
C
C  READ NON-SYMMETRIC TIME MATRIX M
C  READ ROWS 2 - N BECAUSE ROW 1 IS SELECTED FROM MATRIX MOR
C  GO TO 54 I = 2, N
C  READ (5, 1000) (M(I,J), J = 1, N)
50 CONTINUE
C
C  LAB(I) = 6 CHARACTER LABEL FOR BUS STOP I
C  LOAD(I) = STUDENT LOAD ASSIGNED TO BUS STOP I
C  READ (5, 1001) (LAB(I), LOAD(I), I = 1, N)
57 CONTINUE
C
C  MAXL = MAXIMUM BUS LOAD
C  C
C  MAXRT = MAXIMUM ALLOWABLE RIDING TIME FOR STUDENT LOAD
C  ASSIGNED TO BUS STOP I OF ANY ROUTE
C  READ (5, 1000) MAXL, MAXRT
C
C  STORE INFINITY ELEMENTS IN COL. 1, ROW N, DIAGONAL OF MATRIX M
C  DO 54 I = 1, N
C  M(I,1) = 9999
C  M(I,N) = 9999
C  M(N,I) = 9999
54 CONTINUE
C
C  WRITE DATA FOR SCHOOL K1 ON TAPE OR DISC A
C  WRITE (2) K1, (LSCH(I), I=1,6), N, KS, KSS, KPER, KPASS,
C  1 (MOR(I,J), I=1,NOR.), J=1:N) (M(I,J), I=1:N), J=1:N)
C  2 (LAB(I), I=1:N), (LOAD(I), I=1:N), MAXL, MAXRT
C  C
C  MINRUS = MINIMUM NUMBER OF RUSES REQUIRED BY SCHOOL K1

POOR ORIGINAL COPY. BEST
AVAILABLE AT TIME FILMED.
**MAXRUS(K1) = MAXIMUM NUMBER OF BUSES ALLOWED BY SCHOOL K1**

153

DO 63 J = 2, NM1
LSUM = LSUM + LOAD(J)
63 CONTINUE

A USEFUL CHANGE WHEN ADJUSTING CONSTRAINTS

IN ORDER TO OBTAIN A SOLUTION UNDER TIGHT RESTRICTIONS

WHERE KRUS IS AN INTEGER GREATER THAN ONE

LPs = PORTION OF THE LOWER ROUND WHICH IS INDEPENDENT

OF THE ORIGIN AND THE NUMBER OF ROUTES USED

MATRIX M(I,J) IS ALTERED DURING THE PROCESS OF FINDING LPS

KDE(J) = 0 MEANS COLUMN J DOES NOT HAVE A ZERO

KDE(J) = 1 MEANS COLUMN J HAS A ZERO

CONTINUE

UPDATE LBS AND KDE(NCOL)
```
KOE( NCOL ) = 1
L3 = LBS + MTN
C REDUCE ROW I BY THE MINIMUM ELEMENT
Do 73 J = 2, N
   M(I, J) = M(I, J) - MIN
73 CONTINUE
74 CONTINUE
C LROI(I) = LOWER ROUND WHEN ALL ROUTES START AT ORIGIN I
C AND MINUS ROUTES ARE USED
C LROP(I) = LOWER ROUND WHEN ALL ROUTES START AT ORIGIN I
C AND MAXUS(K1) ROUTES ARE USED
Do 117 I = 1, N
   MTN = 77777
   Do 89 J = 2, N
      IF ( M(I, J) - MIN ) .GE. 95, 89, 89
         R5 NCOL = J
         MTN = M(I, J)
         R9 CONTINUE
      END IF
      KOE( NCOL ) = 1
      LROI(I) = LBS + ( MINUS( MIN ) )
      LROP(I) = LBS + ( MAXUS(K1) * MIN )
      Do 100 J = 2, N
         M(I, J) = M(I, J) - MIN
      100 CONTINUE
      Do 113 J = 2, N
         IF ( KOE(J) ) .GE. 93, 93, 113
            MTN = 77777
65 CONTINUE
```
DO 1 IF (M(K,J) = MIN) 97, 10 1
97 MIN = M(K,J)
101 CONTINUE
IF (J = N) 105, 109, 109
105 LA(J) = LB01(I) + MIN
LA(J) = LB02(I) + MIN
GO TO 113
109 LA(J) = LB01(I) + (MINUS * MIN)
LA(J) = LB02(I) + (MAXUS(KI) * MIN)
113 CONTINUE
C RESTORE VECTOR KDE TO STATUS OF STATEMENT 75
KDE( NCOL ) = 0
117 CONTINUE
C WRITE LOWER BOUND DATA FOR SCHOOL K1 ON TAPE OR DISC A
WRITE (2) (LB01(I), I = 1, N), (LB02(I), I = 1, N)
WRITE (2) (LB02(I), I = 1, N), (M(I,J), I = 1, N), (J = 1, N)
C TEST - NOR = 1
120 IF (NOR - 1) 17, 17, 121
121 LA(I,KI) = AVERAGE LOWER BOUND OR EXPECTED MINIMUM COST OF ONE
C ROUTE STARTING AT ORIGIN I FOR SCHOOL K1 WHEN
C MAXBUS(K1) BUSES ARE USED AND ALL ROUTES START
C AT ORIGIN I
122 DO 123 I = 1, NOR
LA(I,KI) = (LB02(I) * 100) / MAXUS(KI)
123 CONTINUE
GO TO 17
C AT 125 ALL LOWER BOUNDS F.T.C. HAVE BEEN CALCULATED
C COMPARE BUS AVAILABILITY AND BUS REQUIREMENTS
ISUM = 0
DO 126 I = 1, NSCH
ISUM = ISUM * MAXUS(I)
126 CONTINUE
GO TO 17
C PREPARE TAPE OR DISC A FOR READING SCHOOL DATA PREVIOUSLY STORED
129 REWIND 2
C
126 CONTINUE
IF ( ISUM = NRSUM ) 128, 128, 127
C ERROR RETURN - BUS SHORTAGE
C
127 WRITE ( 6, 1003 )
WRITE ( 6, 999 )
WRITE ( 6, 2000 ) ISUM, NRSUM
2000 FORMAT ( 1HI0, 5X, 5SHERROR, 5X, 14, 14HUSES REQUIRED,
1 6X, 14, 15HUSES AVAILABLE )
GO TO 1
C TEST - NOR = 1
128 IF ( NOR = 1 ) 139, 139, 129
C EXECUTE TRANSPORTATION ALGORITHM
129 CALL TRAN (NOR, NSCH, NROW, MAXBUS, LRA, NBUS, KONE )
C NBUS(I,J) = NUMBER OF BUSES WHICH START FROM ORIGIN I AND
C TRAVEL TO TERMINUS OR SCHOOL J
C NSUPER(J) = NUMBER OF THE SUPER-ORIGIN OR THE ORIGIN WHICH
C SERVES THE GREATEST NUMBER OF ROUTES FOR SCHOOL J
D0 137 J = 1 NSCH
MAX = 0
D0 135 I = 1, NOR
IF ( NBUS(I,J) = MAX ) 135, 135, 133
133 MAX = NBUS(I,J)
NSROW = I
135 CONTINUE
NSUPER(J) = NSROW
137 CONTINUE
GO TO 143
C AT 139 ALL SCHOOLS ARE SERVICED BY ORIGIN I
139 D0 141 J = 1 NSCH
NSUPER(J) = 1
141 CONTINUE
C START PROCESSING EACH SCHOOL INDIVIDUALLY
C KI = COUNTER FOR NUMBER OF SCHOOLS PROCESSED
143 KI = 0
C  INTERMEDIATE PRINT-OUT OF PERIOD DATA
IF ( KS ) 145, 145, 152
152 WRITE ( 6, 1003 )
WRITE ( 6, 999 )
WRITE ( 6, 1029 )
1029 FORMAT ( 1H0, 5X, 10HMATR1X LBA )
DO 153 I = 1, NOR
WRITE ( 6, 1020 ) I
WRITE ( 6, 1021 ) ( LRA(I,J), J = 1, NSCH )
153 CONTINUE
WRITE ( 6, 1030 )
1030 FORMAT ( 1H0, 5X, 4HNBUS )
DO 154 I = 1, NOR
WRITE ( 6, 1020 ) I
WRITE ( 6, 1021 ) ( NBUS(I,J), J = 1, NSCH )
154 CONTINUE
WRITE ( 6, 1031 )
1031 FORMAT ( 1H0, 5X, 6HNUPER )
WRITE ( 6, 1021 ) ( NUPER(I,J), J = 1, NSCH )
WRITE ( 6, 1032 )
1032 FORMAT ( 1H0, 5X, 4HNRAV )
WRITE ( 6, 1021 ) ( NRAV(I,J), I = 1, NOR )
WRITE ( 6, 1033 )
1033 FORMAT ( 1H0, 5X, 6HMAXBUS )
WRITE ( 6, 1021 ) ( MAXBUS(I,J), J = 1, NSCH )
C  UPDATE K1
145 K1 = K1 + 1
C  TEST - K1 = NSCH
IF ( K1 = NSCH ) 149, 149, 1
C  READ DATA FROM TAPE OR DISC A FOR SCHOOL K1
149 READ (2) KK1, (LSCH(I), I=1,6), N, KS, KSS, KPER, KPAS,
1 ( M(I,J), I=1*NOR ), J=1*N ), ( M(I,J), I=1*N ), J=1*N ),
2 ( LRA(I), I=1*N ), ( LRA(I), I=1*N ), MAXBL, MAXRT
READ (2) ( LB01(I), I = 1, NOR ), ( LB02(I), I = 1, NOR )
C L1 = LOWER ROUND ON A SET OF ROUTES FOR SCHOOL K1 WHEN MINBUS ROUTES ARE USED
C ROUTES ARE USED AND ALL ROUTES START AT NSUPER(K1)
C L2 = SAME AS L1 WHEN MAXBUS(K1) ROUTES ARE USED
I = NSUPER(K1)
L1 = LB01(I)
L2 = LB02(I)
163 NP1 = N + 1
NP2 = N - 1
NP2 = N - 2
C MAXT1 = MAXIMUM TOTAL TIME ALLOWED FOR THE SET OF ROUTES
C FOR SCHOOL K1 WHEN MINBUS ROUTES ARE USED
C MAXT2 = SAME AS MAXT1 WHEN MAXBUS(K1) ROUTES ARE USED
C PER = KPER
C PER = .01 * PER
XL1 = LB1
XL2 = LB2
XTT1 = ((1.0 + PER) * XLR1) + 0.999999
XTT2 = ((1.0 + PER) * XLR2) + 0.999999
MAXT1 = XTT1
MAXT2 = XTT2
C NPASS = NUMBER OF CURRENT INFINITE BUS ROUTE BEING GENERATED
C NOTE - EVERY INFINITE BUS ROUTE CANNOT BE GENERATED INTO A SET
C OF ROUTES WHICH SATISFY BUS CAPACITY, PASSENGER RIDING
C TIME AND BUS AVAILABILITY CONSTRAINTS
C NPASS = 0
C KODE2 = CODE FOR METHOD OF GENERATING THE INFINITE BUS ROUTE
C KODE2 = 1 MEANS USE THE NEAREST-CITY METHOD
C KODE2 = 2 MEANS USE ALGORITHMS
C KODE2 = 1
C IT1 = TOTAL TRAVELING TIME REQUIRED BY THE BEST INFINITE BUS ROUTE
C IT1 = 77777
C IR1 = VECTOR CONTAINING THE BEST INFINITE BUS ROUTE
DO 169 I = 1, N
IR1(I) = 0
160 CONTINUE
C MINTT = TOTAL TRAVELING TIME REQUIRED BY THE BEST SET OF ROUTES
MINTT = 77777
C MPASS = NUMBER OF PASS GENERATING THE BEST SET OF ROUTES
MPASS = 0
C MINTT = VECTOR CONTAINING THE BEST SET OF ROUTES FROM NSUPER(K1)
C THE FORMAT OF MINTT IS
C WORD 1 = NUMBER OF ROUTES IN THE SET
C FOR EACH ROUTE
C WORD 1 = NUMBER OF STOPS SERVICED EXCLUDING ORIGIN AND TERMINUS
C WORDS 2 ... LIST OF STOPS VISITED
C K = N * MAXBUS(K1) - 1
DO 173 I = 1, K
MINTT(I) = 0
173 CONTINUE
C SHIFT ROW NSUPER(K1) FROM MATRIX MOR TO ROW 1 OF MATRIX M
K = NSUPER(K1)
DO 177 J = 1, N
M(I,J) = MOR(K,J)
177 CONTINUE
C INTERMEDIATE PRINT-OUT - ALL DATA FOR SCHOOL
IF ( KS ) 182, 182, 56
56 WRITE ( 6, 1003 )
WRITE ( 6, 1004 ) ( LSCH(I), I = 1, 6 )
WRITE ( 6, 1005 ) MAXBL, MAXRT
WRITE ( 6, 1019 )
1019 FORMAT ( 1H0, 5X, 10MHMATRIX MOR )
DO 57 I = 1, N
WRITE ( 6, 1020 ) I
WRITE ( 6, 1021 ) ( MOR(I,J), J = 1, N )
57 CONTINUE
1020 FORMAT ( 1H0, 5X, 3HROW, 2X, I4 )
1021 FORMAT ( 1H0, 5X, 20I6 )
WRITE ( 6, 1022 )
1022 FORMAT ( 1H6, 5X, BMATRIX M )
   DO 58 I = 1, N
   WRITE ( 6, 1020 ) I
   WRITE ( 6, 1021 ) ( M(I,J), J = 1, N )
58 CONTINUE
   WRITE ( 6, 1023 )
1023 FORMAT ( 1H6, 5X, 6HARELS )
   WRITE ( 6, 1024 ) ( LARI(I), I = 1, N )
1024 FORMAT ( 1H6, 5X, 20A1 )
   WRITE ( 6, 1025 )
1025 FORMAT ( 1H6, 5X, 5HLOADS )
   WRITE ( 6, 1021 ) ( LOAD(T), T = 1, N )
   WRITE ( 6, 1027 )
1027 FORMAT ( 1H6, 5X, 4HLR01 )
   WRITE ( 6, 1021 ) ( LRO1(I), I = 1, N )
   WRITE ( 6, 1028 )
1028 FORMAT ( 1H6, 5X, 4HLR02 )
   WRITE ( 6, 1021 ) ( LRO2(T), T = 1, N )
   WRITE ( 6, 1034 ) L31, L82, MXTT1, MXTT2
1034 FORMAT ( 1H6, 5X, 3HLRI, 2X, I6, 4X, 3HLB2, 2X, I6, 4X,
   1 AHB2, 2X, I6, 4X, AHMAXTT1, 2X, I6 )
   M(I,I+ = 0
   M(I,NP1) = 0
   M(NP1,I) = 0
180 CONTINUE
C IRTP = VECTOR CONTAINING THE SET OF ROUTES OBTAINED FROM THE
C CURRENT INFINITE BUS ROUTE
C THE FORMAT OF IRTP IS THE SAME AS THE FORMAT OF MINRT
C
C UPDATE NPASS
191 NPASS = NPASS + 1
IF ( NPASS = KPASS ) 192, 192, 610
192 K = N + N + 1
DO 193 I = 1, K
IRTP(I) = 0
193 CONTINUE
C IRTF = VECTOR CONTAINING THE CURRENT INFINITE BUS ROUTE
DO 197 I = 1, N
IRTF(I) = 0
197 CONTINUE
C GENERATE THE NEXT INFINITE BUS ROUTE
GO TO 191
C THE NEAREST-CITY APPROACH - IRN01 IS THE FIRST STOP VISITED
201 IRNO = IRN01
M(1,NP1) = IRNO
M(NP1,IRNO) = 1
M(1,1) = M(1,IRNO)
M(NP1,1) = N
M(N,NP1) = 1
DO 213 I = 2, NM2
MTN = 77777
IMINT = 0
DO 209 J = 2, NM1
IF ( M(J,NP1) ) 209, 209, 209
209 IF ( IRNO = J ) 205, 209, 209
205 IF ( M(IRNO, J) - MTN ) 207, 209, 209
207 MTN = M(IRNO, J)
IMINT = J
209 CONTINUE
M(IRNO, NP1) = IMINT
\[ M(\text{NP1, IDMIN}) = \text{IRNO} \]
\[ M(\text{IRNO, IRNO}) = \text{MIN} \]
\[ \text{IRNO} = \text{IDMIN} \]

219 CONTINUE
\[ M(\text{IRNO, NP1}) = N \]
\[ M(\text{NP1, N}) = \text{IRNO} \]
\[ M(\text{IRNO, IRNO}) = M(\text{IRNO, N}) \]

C ITT = TOTAL TRAVELING TIME FOR THE CURRENT INFINITE BUS ROUTE

C DETERMINE ITT AND VECTOR IRTE

219 CONTINUE
\[ \text{ITT} = 0 \]
\[ \text{IRT}(1) = 1 \]
\[ I = 1 \]
\[ \text{DO 221} \quad K = 2, N \]
\[ J = M(I, \text{NP1}) \]
\[ \text{ITT} = \text{ITT} + M(I, J) \]
\[ \text{IRT}(K) = J \]
\[ I = I + 1 \]

221 CONTINUE
C TEST - ITT = ITT1
\[ \text{IF} \quad (\text{ITT} = \text{ITT1}) \text{ 223, 230, 230} \]

C THE CURRENT INFINITE BUS ROUTE IS BEST

223 IT1 = ITT
\[ \text{DO 225} \quad I = 1, N \]
\[ \text{IRT}(I) = \text{IRTE}(I) \]

225 CONTINUE

C PARTITIONING INFINITE BUS ROUTE INTO INDIVIDUAL CONSTRAINT ROUTES

C NR = NUMBER OF CURRENT ROUTE
C NAS = NUMBER OF STOPS COMPLETELY SERVICED
C NAS1 = IDENTIFICATION NUMBER OF STOP 1 ON CURRENT ROUTE
C NASRT = NUMBER OF BUS STOPS ASSIGNED TO CURRENT ROUTE
C ITT1 = TOTAL RIDING TIME FROM BUS STOP 1 ON CURRENT ROUTE
C NASRTD = NUMBER OF PREVIOUS BUS STOP VISITED ON CURRENT ROUTE
C NASNEW = NUMBER OF PRESENT BUS STOP VISITED ON CURRENT ROUTE
C IAROLD = ARRIVAL TIME AT PREVIOUS BUS STOP
C IARNEW = ARRIVAL TIME AT PRESENT BUS STOP
C IRLLOD = BUS LOAD AT PREVIOUS STOP
C IRLNEW = BUS LOAD AT PRESENT STOP
C KODF3 = 0 MEANS THE CURRENT ROUTE IS INCOMPLETE
C KODF3 = 1 MEANS THE CURRENT ROUTE IS COMPLETE
C KODF3 = 2 MEANS ALL THE BUS STOPS HAVE BEEN SERVICED

236 NR = 0
    NRS = 0
    NPS = 0
C INITIALIZE FOR A NEW ROUTE
234 NR = NR + 1
    KODF3 = 0
    NRSOLD = 0
    NRSNEW = 1
    IAROLD = 0
    IARN = 0
    IRLOLD = 0
    IRLNEW = 0
    ITBS1 = 0
    NRSRT = 0
    NRS1 = INT( NBS + 2 )
242 NRSOLD = NBSNEW
    IAROLD = IARN
    IRLOLD = IRLNEW
    NRSNEW = INT( NBS + 2 )
    IARN = IAROLD + M( NRSOLD, NBSNEW )
    IRLNEW = IRLOLD + LOAD( NRSNEW )
    ITBS1 = IARN + M( NRSNEW, N ) - M( 1, NRS1 )
C TEST - BUS CAPACITY EXCEEDED
254 IF ( IRLNEW - MAXQL ) 256, 266, 270
C AT 256 THE BUS IS NOT FULL
C TEST - RIDING TIME EXCEEDED
254 IF ( ITBS1 - MAXRT ) 273, 258, 270
C AT 258 THE BUS IS NOT FULL BUT MAXIMUM RIDING TIME REACHED
C CURRENT ROUTE IS COMPLETE
25A: KODE3 = 1
475
GO TO 273
476
C AT 266 THE BUS IS FULL
477
C TRESH - RIDING TIME EXCEEDED
478
266 IF ( ITBS1 - MAXRT ) 26A, 26A, 27A
479
C AT 26A THE BUS IS FULL BUT THE RIDING TIME HAS NOT BEEN EXCEEDED
480
C CURRENT ROUTE IS COMPLETE
481
26A KODE3 = 1
482
GO TO 273
483
C AT 27A SERVICING THE CURRENT STOP WOULD VIOLATE A CONSTRAINT
484
C THE PREVIOUS STOP COMPLETED THE ROUTE
485
27A KODE3 = 1
486
GO TO 275
487
C AT 275 THE CURRENT STOP WILL BE ASSIGNED TO THE CURRENT ROUTE
488
275 NRS = NBS + 1
489
NRSRT = NBSRT + 1
490
K = NR + NBS + 1
491
IRTP( K ) = NRSRT
492
C TEST - ALL STOPS SERVICED
493
277 IF ( NRS = N + 2 ) 279, 277, 277
494
C AT 277 ALL STOPS SERVICED
495
277 KODE3 = 2
496
C TEST - PARTITIONING COMPLETED
497
279 IF ( KODE3 = 1 ) 248, 281, 281
498
C AT 281 THE CURRENT ROUTE IS COMPLETE BECAUSE EITHER THE CONSTRAINT LIMITS WERE REACHED OR ALL THE STOPS WERE SERVICED
499
C TEST - AT LEAST ONE STOP ON ROUTE
500
281 IF ( NBSRT ) 285, 285, 287
501
C AT 285 THE CURRENT ROUTE IS LEGAL, STORE NBSRT IN VECTOR IRTP
502
287 K = NR + NBS - NBSRT + 1
503
IRTP( K ) = NBSRT
504
IF ( KODE3 = 1 ) 234, 234, 287
505
C AT 234 THE CURRENT ROUTE IS NOT LEGAL, GENERATE NEXT INFINITE RTE.
506
234 GO TO ( 183, 191 ), KODE2
507
AT 287 PARTITIONING IS COMPLETE

IRTP(I) = NR
INTERMEDIATE PRINT-OUT
IF ( KS ) 293, 294, 289
WRITE ( 6, 1035 ) NPASS, KPASS, KODE2, IT1, ITT, MINTT, MPASS
1035 FORMAT ( 1H0, 5X, 5HNPASS, I6, 5X, 5HKPASS, I6, 5X, 5HKODE2,
1 I6, 5X, 3HIT1, I6, 5X, 3HITT, I6, 5X, 5HMINN, I6, 5X,
2 5HMPASS, I6 )
WRITE ( 6, 1036 )
1036 FORMAT ( 1H0, 5X, 5HVECTOR IR1 )
WRITE ( 6, 1021 ) ( IR1(I), I = 1, N )
WRITE ( 6, 1037 )
1037 FORMAT ( 1H0, 5X, 10HVECTOR IRTE )
WRITE ( 6, 1021 ) ( IRTE(I), I = 1, N )
WRITE ( 6, 1038 )
1038 FORMAT ( 1H0, 5X, 10HVECTOR IRTP )
K = NR + N - 1
WRITE ( 6, 1021 ) ( IRTP(I), I = 1, K )
WRITE ( 6, 1045 )
1045 FORMAT ( 1H0, 5X, 5HMINR )
K = N + MAXB(U)(K1) - 1
WRITE ( 6, 1021 ) ( MINR(I), I = 1, K )
TEST - NR SATISFY BUS AVAILABILITY CONSTRAINT
297 IF ( NR - MAXB(U)(K1) ) 305, 306, 295
295 GO TO ( 183, 191 ), KODE2
C AT 305 NR DOES NOT EXCEED MAXB(U)(K1)
C IMPROVE EACH OF THE INDIVIDUAL ROUTES
C ITIMP = TOTAL TRAVELING TIME REQUIRED BY THE SET OF IMPROVED ROUTE
C INITIALIZE FOR IMPROVING CURRENT ROUTE
C TEST - ALL ROUTES OF VECTOR IRTP IMPROVED
C
IF (NR = IRTP(I) ) 313, 313, 575
C AT 313 AT LEAST ONE ROUTE HAS TO BE IMPROVED
C NASRT = NUMBER OF STOPS ON THE CURRENT ROUTE EXCLUDING THE
C ORIGIN AND THE TERMINUS
317 K = NP * NBS + 1
   NASRT = IRTP(K)
C TEST = NASRT + 1
   IF (NASRT = 1 ) 317, 317, 321
C AT 317 THE CURRENT ROUTE HAS ONE STOP. NO IMPROVEMENT POSSIBLE
317 NPS = NBS + NASRT
   J = IRTP(K+1)
   ITIMP = ITIMP + M(I,J) + M(J,N)
GO TO 309
C KDE(I) = 0 MEANS ROW I AND COLUMN I OF MATRIX M WILL NOT
C BE USED FOR THE IMPROVEMENT PROCESS FOR ROUTE K>
C KDE(I) = 1 MEANS ROW I AND COLUMN I OF MATRIX M WILL
C BE USED FOR THE IMPROVEMENT PROCESS FOR ROUTE K>
321 DO 325 I = 2, NM1
   KDE(I) = 0
325 CONTINUE
C ROW 1, ROW N, COL.1, COL.N OF MATRIX M ARE ALWAYS USED
C KDE(I) = 1
C KDE(N) = 1
C IRTF2 = VECTOR CONTAINING THE CURRENT ROUTE EXPRESSED AS AN
C INFINITE BUS ROUTE INCLUES ORIGIN AND TERMINUS
C N2 = NUMBER OF STOPS ON THE ROUTE BEING IMPROVED
C N2 INCLUDES THE ORIGIN AND THE TERMINUS
329 N2 = NASRT + 2
   DO 333 I = 1, N2
      IRTF2(I) = 0
333 CONTINUE
   J = NR * NBS + 1
   DO 333 I = 1, NASRT
      K = J + I
IRF2(I+1) = IRS

KOE(IRS) = 1

CONTINUE

IRF2(1) = 1

IRF2(N2) = N

NOTE: ALL NUMBERING OF STOPS IS WITH RESPECT TO MATRIX M IN IRTE2

MATRIX M2 IS THE TIME MATRIX FOR THE INDIVIDUAL ROUTE

TO BE IMPROVED, IT IS OF ORDER N2

Ii = ROW NUMBER OF MATRIX M TO BE SHIFTED TO MATRIX M2

J1 = COLUMN NUMBER OF MATRIX M TO BE SHIFTED TO MATRIX M2

LIST = VECTOR OF ROW-COLUMN NUMBERS DESIGNATING THE
CONSTRUCTION OF MATRIX M2 FROM MATRIX M

K = COUNTER FOR POSITION IN VECTOR LIST

K = 0

DO 345 I = 1, N

IF (KOE(I)) \ 341, 345, 341

AT 341 ROW I OF MATRIX M IS TO BE SHIFTED TO ROW K OF MATRIX M2

K = K + 1

LIST(K) = I

CONTINUE

M = 341

DO 345 I = 1, N2

J = LIST(I)

M2(I,J) = M(I,J)

CONTINUE

RELATE EACH ROW OF MATRIX M2 TO VECTOR IRF2 SO THAT THE ROUTE
BEING IMPROVED IS THE INITIAL ROUTE FOR ALGORITHMS A AND B

N2P1 = N2 + 1

N2M1 = N2 - 1

M2(N2, N2P1) = 1

M2(N2P1, 1) = N2
M2( N2P1, N2P1 ) = 0
I1 = COUNTER FOR PICKING OFF THE STOPS FROM IRTE?
J1 AND KK1 ARE THE STOP NUMBERS WITH RESPECT TO MATRIX M
J AND K ARE THE STOP NUMBERS WITH RESPECT TO MATRIX M2
DO 379 I1 = 1, N2M1
J1 = TERMINUS OF CURRENT LINK, NUMBERING W.R.T. MATRIX M
J1 = IRTE2( I1 + 1 )
DO 365 I2 = 1, N2
I: ( LIST(I2) = J1 ) 365, 361, 365
AT 361 STOP J1 OF MATRIX M IS STOP I2 OF MATRIX M2
361 J = I2
GO TO 367
365 CONTINUE
367 DO 371 I2 = 1, N2
IF ( LIST(I2) = KK1 ) 371, 369, 371
C AT 369 STOP KK1 OF MATRIX M IS STOP I2 OF MATRIX M2
369 K = I2
GO TO 375
371 CONTINUE
375 M2( K, N2P1 ) = J
M2( N2P1, J ) = K
M2(K,K) = M2(K,J)
379 CONTINUE
C MATRIX M2(I,J) IS READY FOR ALGORITHM A
C ALGORITHM A
426 DO 460 I = 1, N2M1
450 J = 2, N2M1
IF ( I = J ) 422, 450, 422
422 IF ( M2( I, N2P1 ) = J ) 424, 450, 424
424 K01 = I
K11 = J
K02 = J

POOR ORIGINAL COPY-BEST
AVAILABLE AT TIME FILMED
\[ KT_2 = M_2(I, N_2P_1) \]
\[ K_03 = M_2(N_2P_1, J) \]
\[ KT_3 = M_2(J, N_2P_1) \]

474 \[ K_{0,1} = M_2(K_{0,1}, K_{0,1}) + M_2(K_{0,2}, K_{0,2}) + M_2(K_{0,3}, K_{0,3}) \]
\[ NEW = M_2(K_{0,1}, KT_1) + M_2(K_{0,2}, KT_2) + M_2(K_{0,3}, KT_3) \]
\[ IF (NEW = K_{0,1}) 470, 427, 427 \]
\[ IF (K_{0,1} = KT_3) 429, 450, 428 \]
478 \[ IF (KT_3 = N_2) 430, 429, 430 \]
479 \[ IF (K_{0,1} = 1) 431, 450, 431 \]

471 \[ K_{0,2} = 1 \]
472 \[ KT_3 = M_2(1, N_2P_1) \]
473 \[ Go TO 426 \]

476 \[ Go TO 426 \]

C AT THIS POINT ALGORITHM A IS EXHAUSTED

467 \[ Go TO 550 \]

C AT THIS POINT THE CURRENT ROUTE HAS BEEN IMPROVED

C ALL ROUTE DATA IS STORED IN MATRIX M2

C IRTF = TOTAL TIME REQUIRED TO TRAVERSE THE IMPROVED ROUTE

C IRTF3 = VECTOR CONTAINING THE CURRENT IMPROVED ROUTE EXPRESSED AS AN INFINITE BUS ROUTE INCLUDES ORIGIN AND TERMINUS
IF "ALL NUMBERING OF STOPS IS WITH RESPECT TO MATRIX M2 IN IRTE3"

IPTT = 0

CONTINUE

IRTF3(I) = 0

CONTINUE

I = 1

CONTINUE

IRTF3(I) = 1

CONTINUE

IRTF3(I) = 1

CONTINUE

UPDATF ITIMP

ITIMP = ITIMP + IPTT

CONTINUE

RELATE THE BUS STOP NUMBERS OF VECTOR IRTE3 IN TERMS OF THE

CONTINUE

STORE THE IMPROVED ROUTE RACK PROPERLY IN VECTOR IRTP

CONTINUE

THE ORIGIN AND TERMINUS IS NOT STORED IN VECTOR IRTP

CONTINUE

UPDATF NBS

NBS = NBS + NBS

CONTINUE

IMPROVE NEXT ROUTE OF STI

GO TO 309

CONTINUE

AT 5744 ALL THE ROUTES OF VECTOR IRTP HAVE BEEN IMPROVED
C  INTERMEDIATE PRINT-OUT
579 IF ( KS ) 578, 579, 576
579 WRITE ( 6, 1046 ) ITIMP
1046 FORMAT ( 1HO, 5X, 5HITIMP, 16 )
    K = IRTP(I) + N - 1
    WRITE ( 6, 1038 )
    WRITE ( 6, 1021 ) ( IRTP(I), I = 1, K )
C  TEST = ITIMP LESS THAN MINTT
579 IF ( ITIMP = MINTT ) 583, 579, 574
C  AT 579 GENERATE ANOTHER INFINITE BUS ROUTE
579 GO TO ( 183, 191 ), KODE2
C  AT 583 THE CURRENT SET OF ROUTES IS THE BEST AVAILABLE
583 MINTT = ITIMP
    MPASS = NPASS
    NR = IRTP(I)
    K = NR + N - 1
    DO 585 I = 1, K
    MINRT(I) = IRTP(I)
585 CONTINUE
C  TEST = NR = MAXBUS(K1)
   IF ( NR = MAXBUS(K1) ) 600, 603, 603
C  TEST = CURRENT SET OF ROUTES ACCEPTABLE AS FINAL SOLUTION
600 IF ( MINIT = MAXIT1 ) 610, 610, 579
603 IF ( MINIT = MAXIT2 ) 610, 610, 579
C  AT 610 A FINAL ACCEPTABLE SOLUTION MAY EXIST
610 IF ( MINRT(1) ) 604, 604, 606
604 WRITE ( 6, 1003 )
    WRITE ( 6, 999 )
    WRITE ( 6, 1004 ) ( LSCH(I), I = 1,6 )
    WRITE ( 6, 1047 )
1047 FORMAT ( 1HO, 5X, 42HNO ACCEPTABLE SOLUTION, ADJUST CONSTRAINTS )
    GO TO 145
C  AT 606 THE BEST SET OF ROUTES IS ACCEPTABLE AS A FINAL SOLUTION
C  LIST(J) = IDENTIFICATION NUMBER OF ORIGIN J SUPPLYING BUSF
FOR SCHOOL K1 - DETERMINED AT STATEMENT 129

CLEAR VECTOR LIST(J)

DO 611 J = 1, NOR

LIST(I) = 0

CONTINUE

KOF4 = CODE FOR CALCULATING FINAL LOWER BOUND

KOF4 = 1

TEST = MORE THAN ONE ORIGIN

IF ( NOR = 1 ) 775, 775, 612

TEST = ORIGIN NSUPER(K1) SUPPLIES ALL THE REQUIRED HOSES

I = NSUPER(K1)

IF ( NAUS(I,K1) = MAXUS(K1) ) 613, 875, 875

DEVFLOP COST MATRIX FOR ALLOCATION OF ORIGINS TO BUS ROUTES

KONRT(I) = CONSTANT PORTION OF ROUTE COST FOR ROUTE I

KONRT(I) = SUM OF ALL LINKS EXCEPT THE ONE FROM ORIGIN TO STOP 1

NSTOP(I) = IDENTIFICATION NUMBER OF STOP 1 ON ROUTE I

NR = n

NRS = 0

CONTINUE

TEST = ALL ROUTES UNPACKED

IF ( NR = MINRT(I) ) 621, 621, 641

L1 = NR + NBS + 1

L2 = I1 + 1

NASRT = MINRT(L1)

KONRT(NR) = 0

K = NASRT - 1

IF ( K = 633; 633, 625

DO 629 I = 1, K

I1 = I1 + 1

J1 = J1 + 1

IBSI = MINRT(I1)

IBS2 = MINRT(J1)

KONRT(NR) = KONRT(NR) + M(IBSI, IBS2)

CONTINUE
L3 = (1 + NASRT)
IRS1 = MINRT(L3)
IRS2 = N
KONRT(NR) = KONRT(NR) + M(IRS1, IRS2)
NSTOP l(NR) = MINRT(L2)

C
UPDATE NHS
NR = IRS + NASRT
GO TO 617

C
KOST(I,J) = TOTAL TRAVELING TIME FOR ROUTE J STARTING AT ORIGIN I
NCR2 = NUMBER OF ORIGINS WHICH SUPPLY BUSES FOR SCHOOL K1
C
IREQ(J) = NUMBER OF BUSES REQUIRED BY ROUTE J

-641 NR = MINRT(I)
NCR2 = 0
DO 649 I = 1, NR
IF (NBUS(I,K1)) 649, 649, 643
AT 649 ORIGIN I SUPPLIES AT LEAST ONE BUS FOR SCHOOL K1
C
UPDATE NCR2, VECTORS LIST AND IVAIL
C
IVAIL(I) = NUMBER OF BUSES AVAILABLE AT ORIGIN I FOR SCHOOL K1

643 NCR2 = NCR2 + 1
LIST(NCR2) = I
IVAIL(NCR2) = NBUS(I,K1)
DO 645 J = 1, NR
K = NSTOP(I(J))
KOST(NCR2,J) = KONRT(J) + MOR(I,K)

645 CONTINUE

644 CONTINUE
DO 653 J = 1, NR
IREQ(J) = 1

653 CONTINUE

C EXECUTE THE TRANSPORTATION ALGORITHM
CALL TRAN(NCR2, NR, IVAIL, IREQ, KOST, NBUS1, KODE)
IF (KS) 713, 713, 650
WRITE (6, 1051)
1051 FORMAT (1HO, 5X, 4HKOST)
ON 651 I = 1, NOR2
WRITE ( 6, 1021 ) ( KOST(I,J), J = 1, NR )
651 CONTINUE
WRITE ( 6, 1052 )
1052 FORMAT ( 1HO, 5X, SHNAUS1 )
ON 652 I = 1, NOR2
WRITE ( 6, 1021 ) ( NAUS1(I,J), J = 1, NR )
652 CONTINUE
WRITE ( 6, 1053 )
1053 FORMAT ( 1HO, 5X, SHIAUS1 )
WRITE ( 6, 1021 ) ( IVAIL(I), I = 1, NOR2 )
WRITE ( 6, 1054 )
1054 FORMAT ( 1HO, 5X, 4HREQ )
WRITE ( 6, 1021 ) ( IREQ(I), I = 1, NR )
WRITE ( 6, 1055 )
1055 FORMAT ( 1HO, 5X, 4HLIST )
WRITE ( 6, 1021 ) ( LIST(I), I = 1, NOR2 )
GO TO 713
C CALCULATE THE FINAL LOWER BOUND FOR THE ROUTES OF SCHOOL K1
C NAUS1(I,J) = 0 MEANS ORIGIN I DOES NOT SERVICE ROUTE J
C NAUS1(I,J) = 1 MEANS ORIGIN I SERVICES ROUTE J
C NAUS2(I) = NUMBER OF ROUTES STARTING FROM ORIGIN I
C NRTOT = TOTAL NUMBER OF BUSES USED BY SCHOOL K1
661 NR = MINRT(1)
ON 665 I = 1, NOR2
NAUS2(I) = 0
ON 661 J = 1, NR
NAUS2(I) = NAUS2(I) + NAUS1(I,J)
661 CONTINUE
665 CONTINUE
C LP = FINAL LOWER BOUND FOR SCHOOL K1
LP = IRS
C FIND PORTION OF LOWER BOUND CONTRIBUTED BY TRAVELING BETWEEN
C ORIGIN I AND THE FIRST STOP OF ANY ROUTE
DO 691 I = 1, N0R2
  K = LIST(I)
  MIN = 77777
  DO 675 J = 2, N0M1
     IF ( MOR(K, J) - MIN ) 669, 673, 673
  669 NCOL = J
  MIN = MOR(K, J)
  677 CONTINUE
  K0E( NCOL ) = 1
  LA = LA + ( N8US2(I) * MIN )
C REDUCE ROW K OF MATRIX MOR BY MIN
  DO 677 J = 2, N0M1
     MOR(K, J) = MOR(K, J) - MIN
  677 CONTINUE
C FIND PORTION OF LOWER BOUND CONTRIBUTED BY COLUMNS
  DO 705 J = 2, N
     IF ( K0E(J) ) 685, 685, 7A5
  6A5 MIN = 77777
  DO 693 I = 1, N0R2
     K = LIST(I)
     IF ( MOR(K, J) - MIN ) 689, 693, 693
  699 MIN = MOR(K, J)
  693 CONTINUE
  DO 701 I = 2, N0M1
     IF ( M(I, J) - MIN ) 697, 701, 701
  697 MIN = M(I, J)
  701 CONTINUE
     IF ( J = N ) 702, 703, 703
  702 LA = LA + MIN
     GO TO 705
  705 CONTINUE
     IF ( N8US2(I) - MIN )
  705 CONTINUE
     GO TO 773
WRITE THE CURRENT ROUTE
NR = NUMBER OF ROUTE BEING PRINTED
NRS = NUMBER OF BUS STOPS OF VECTOR MINRT PRINTED
NR = 0
NRS = 0
INITIALIZE FOR CURRENT ROUTE
NR = NR + 1
TEST- ALL ROUTES PRINTED.
IF (NR = MINRT(1)) 719, 719, 767
K3 = NR + NRS
NRSRT = MINRT(K3 + 1)
NR = NUMBER OF STOPS ON CURRENT ROUTE - EXCLUDING ORG. AND TER.
K = NRSRT + 1
ILOAD(I) = TOTAL BUS LOAD AT STOP I
IARRT(I) = ARRIVAL TIME AT STOP I
INUM(I) = NUMERIC LABEL FOR STOP I
IALPH(I) = ALPHABATIC LABEL FOR STOP I
DO 731 I = 1, NR2
IF (NBUS(I, NR)) 731, 731, 727
AT 727 ORIGIN I SERVS ROUTE NR
K4 = LIST(I)
GO TO 735
CONTINUE
DO 735 I = 1, K
IF (I = 2) 739, 743, 747
INUM(I) = I
IALPH(I) = LR(K4)
IARRT(I) = 0
ILOAD(I) = 0
GO TO 751
K7 = K3 + I
K6 = MINRT(K7)
INUM(I) = K6
IALPH(I) = LAB(K6)
IARRT(I) = IARRT(I-1) + MOR(K4, K6)  
ILOAD(I) = ILOAD(I-1) + LOAD(K6)  
GO TO 751  

747 K7 = K3 + I  
K6 = MINRT(K7)  
K5 = MINRT(K7 - 1)  
INUM(I) = K6  
IALPH(I) = LAB(K6)  
IARRT(I) = IARRT(I-1) + M(K5, K6)  
ILOAD(I) = ILOAD(I-1) + LOAD(K6)  
CONTINUE  

K7 = K3 + K  
K4 = N  
K5 = MINRT(K7)  
I = K + 1  
INUM(I) = K6  
IALPH(I) = LAB(K6)  
IARRT(I) = IARRT(I-1) + M(K5, K6)  
ILOAD(I) = ILOAD(I-1)  
WRITE (6, 1003)  
1003 FORMAT (1H1)  
WRITE (6, 999)  
WRITE (6, 1004) (LSCH(J), J=1,6)  
1004 FORMAT (1H0, 5X, 6A10)  
WRITE (6, 1005) MAXRT, MAXRT  
1005 FORMAT (1H0, 5X, 16HMAXIMUM BUS LOAD, I6, 7X, 19HMAXIMUM RIDING TIME, I6)  
WRITE (6, 1050) MPASS  
1050 FORMAT (1H0, 5X, 16HACCERTEO SET OF ROUTES FROM ITERATION, I6)  
WRITE (6, 1006) NR, K4, LOR(K4)  
1006 FORMAT (1H0, 5X, 16HROUTE, I5, 22X, 6HORIGIN, I5, 2X, A6)  
WRITE (6, 1007)  
1007 FORMAT (1H0, 5X, 19HSTOP IDENTIFICATION, 6X, 4HTIME, 6X, 4HLOAD)  
WRITE (6, 1008) (INUM(J), IALPH(J), IARRT(J), ILOAD(J), J=1,I)
1009 FORMAT ( 1H0, 5X, I6, 7X, A6, 6X, I4, 6X, I4 )
C PREPARE DATA FOR FINAL STATISTICS
C KRT(I) = NUMBER OF ROUTE I
C KORN(I) = NUMERIC LABEL FOR ORIGIN OF ROUTE I
C KORA(I) = ALPHANERIC LABEL FOR ORIGIN OF ROUTE I
C KT(I) = TOTAL TIME REQUIRFO BY ROUTE I
C KL(I) = TOTAL BUS LOAD FOR ROUTE I
C KRT(NR) = NR
KORN(NR) = K4
KORA(NR) = LOR(K4)
KT(NR) = IARRT(I)
KL(NR) = ILOAD(I)
C UPDATE NBS
NBS = NBS + NBSRT
GO TO 715
C AT 767 WRITE FINAL_STATISTICS FOR SCHOOL K1
767 WRITE ( 6, 1003 )
WRITE ( 6, 1004 ) ( LSCH(I), I = 1, 6 )
WRITE ( 6, 1005 ) MAXRL, MAXRT
WRITE ( 6, 1009 )
1009 FORMAT ( 1H0, 5X, 5HROUTE, 6X, 15HORIGIN OF ROUTE, 6X, 4HTIME,
1 6X, 4HLOAD )
K = MNRT(I)
WRITE ( 6, 1010 ) ( KRT(J), KORN(J), KORA(J), KT(J), KL(J), J = 1, K)
1010 FORMAT ( 1H0, 5X, I5, 6X, I4, 5X, A6, 6X, I4, 6X, I4 )
KTOT = 0
DO 771 J = 1, K
KTOT = KTOT * KT(J)
771 CONTINUE
WRITE ( 6, 1011 ) KTOT
1011 FORMAT ( 1H0, 5X, 2BHTOTAL TIME FOR SET OF ROUTES, I8 )
READ (2) LBS, ( KN(E(I), I = 1, N ), ( M(I,J), I = 1, N ), J = 1, N )
GO TO ( 660, 773 ), KODE4

POOR ORIGINAL COPY - BEST AVAILABLE AT TRAF FILMED
WRITE ( 6, 1012 ) LB  

FORMAT ( 1H0, 5X, 25H LOWER BOUND FOR SET OF ROUTES, I7 )  
AKTOT = KTOT  
ALB = LB  
RATIO) = AKTOT / ALB  
PER = KPER  
RATTO = ( PER / 100.0 ) + 1.0  
WRITE ( 6, 1013 ) RATTO  

FORMAT ( 1H0, 21X, 12H ACTUAL RATIO, 6X, F5.2 )  
WRITE ( 6, 1014 ) RATTO  

FORMAT ( 1H0, 21X, 13H ALLOWED RATIO, 5X, F5.2 )  
GO TO 145  

C AT 775 THERE IS ONLY 1 ORIGIN - NO NEED TO ASSIGN ORIGINS  
C NUS(l,j) = 0 MEANS ORIGIN I DOES NOT SERVICE ROUTE J  
C NUS(l,j) = 1 MEANS ORIGIN I SERVICES ROUTE J  

LIST(1) = 1  
I = 1  

NR = MINRT(1)  
C NCR = NUMBER OF ORIGINS WHICH SUPPLY BUSES FOR SCHOOL K1  
DO 779 J = 1, NR  
NUS(l, j) = 1  
CONTINUE  
KODE4 = 2  

C AT THIS POINT SET KODE4 = 1  
C IF MAXUS(1) = MINBUS + KHUS  
C IN ORDER TO CALCULATE CORRECT LOWER BOUND  
C ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **  
C IF (NR = MAXBUS(1) ) 787 787 787  
C LR = FINAL LOWER BOUND FOR A SET OF ROUTES  

787 LR = LR01(I)  
GO TO 713  
787 LR = LR02(I)  
GO TO 713
The nearest-city approach has been exhausted.

Store the best infinite bus route into matrix \( M \).

Vector \( h_I \) contains the best route calculated.

\[
\begin{align*}
M(\, N, \, nP_1 \,) & = 1 \\
M(\, nP_1, \, 1 \,) & = N \\
D_n & = \text{if } I = 1, \text{NM}_1 \\
J & = I + ( I + 1 ) \\
K & = I + ( I + 1 ) \\
M(\, K, K \,) & = M(\, K, J \,) \\
M(\, K, nP_1 \,) & = J \\
M(\, nP_1, J \,) & = K
\end{align*}
\]

Continue generating the next infinite bus route by algorithms.

At \( 801 \) generate the infinite bus route by using the algorithms.

Algorithm A

At \( 801 \):
- \( D_n \) if \( I = 1, \text{NM}_1 \)
- \( D_n \) if \( J = 1, \text{NM}_1 \)
- IF ( \( I = J \) ) \( A_22, \, A_50, \, A_22 \)
- IF ( \( M(\, I, nP_1 \,) - J \) ) \( A_24, \, A_50, \, A_24 \)

At \( 824 \): \( \text{Kn}_1 = I \)
- \( \text{Kn}_1 = J \)
- \( \text{Kn}_2 = J \)
- \( \text{Kn}_3 = M(\, nP_1, \, J \,) \)
- \( \text{Kn}_3 = M(\, J, nP_1 \,) \)

At \( 824 \): \( K_01 = M(\, K_01, \, K_01 \,) + M(\, K_02, \, K_02 \,) + M(\, K_03, \, K_03 \,) \)
- \( \text{NEW} = M(\, K_01, \, \text{KT}_1 \,) + M(\, K_02, \, \text{KT}_2 \,) + M(\, K_03, \, \text{KT}_3 \,) \)
- IF ( \( \text{NEW} = K_01 \) ) \( A_70, \, A_27, \, A_27 \)
- IF ( \( \text{KT}_1 = \text{KT}_3 \) ) \( A_72, \, A_50, \, A_28 \)
- IF ( \( \text{KT}_3 = N \) ) \( A_30, \, A_39, \, A_30 \)
- IF ( \( K_01 = 1 \) ) \( A_31, \, A_50, \, A_31 \)
- \( K_02 = 1 \)
KT3 = M( I, NP1 )
GO TO 826

830 K02 = KT3
KT3 = M( K02, NP1 )
GO TO 826

850 CONTINUE
860 CONTINUE

C AT THIS POINT ALGORITHM A IS EXHAUSTED
C WRITE THE BEST SET OF ROUTES AS THE FINAL SOLUTION
C
GO TO 610

C AT R70 AN IMPROVEMENT IN THE INFINITE BUS ROUTE IS POSSIBLE
870 M( K01, K01 ) = M( K01, KT1 )
M( K02, K02 ) = M( K02, KT2 )
M( K03, K03 ) = M( K03, KT3 )
M( K01, NP1 ) = KT1
M( K02, NP1 ) = KT2
M( K03, NP1 ) = KT3
M( NP1, KT1 ) = K01
M( NP1, KT2 ) = K02
M( NP1, KT3 ) = K03
C
C PARTITION THE IMPROVED INFINITE BUS ROUTE
GO TO 219

C AT R75 ORIGIN NSUPER(KI) SUPPLIES ALL THE REQUIRED BUSES
C THERE IS ONLY ONE ORIGIN - NO NEED TO ASSIGN ORIGINS TO ROUTES
875 LIST(1) = NSUPER(KI)
I = LIST(1)
GO TO 777

END

SUBROUTINE TRAN ( NORIG, NDEST, IORIG, IDEST, ICOST, IBAS, KE ) 
DIMENSION IORIG(10), IDEST(35), ICOST(10,35), IBAS(10,35)
DIMENSION IU(10), IV(35), IU1(10), IV1(35)
DIMENSION INET(88), INET1(10), INET2(35)
ARRAYS MUST BE DIMENSIONED AS FOLLOWS -

LET I = NUMBER OF ORIGINS

.1 = NUMBER OF DESTINATIONS

IRAS(I+1, J+1), ICOST(I+1, J+1), IU(I+1, J+1), IV(I+1, J+1), IU1(I+1)

IV1(J+1), INET(2I + 2J + 2), INFL1(I+1), INFL2(J+1)

IC1R(I+1), IDEST(J+1)

** IORIG(I) MUST BE GREATER THAN ZERO FOR ALL I

*** IDEST(J) MUST BE GREATER THAN ZERO FOR ALL J

** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **

TRAN007

TRAN008

TRAN009

TRAN010

TRAN011

TRAN012

TRAN013

TRAN014

TRAN015

TRAN016

TRAN017

TRAN018

TRAN019

TRAN020

TRAN021

TRAN022

TRAN023

TRAN024

TRAN025

TRAN026

TRAN027

TRAN028

TRAN029

TRAN030

TRAN031
C ICOST(I,J) = COST OF SHIPPING A UNIT FROM I TO J

1101 DO 1104 I = 1, NORIG
  ICOST(I,INDEST) = 999999
1104 CONTINUE
  GC TO 1112
C DUMMY ORIGIN REQUIRED
  1111 NORIG = NORIG + 1
  NINRIG(NORIG) = TSUM = 1SUMO
1112 DO 1115 J = 1, NDEST
  ICOST(NORIG,J) = 999999
1115 CONTINUE
C ICAS(I,J) = NUMBER OF UNITS SHIPPED FROM I TO J
1116 DO 1117 I = 1, NORIG
  ICAS(I,J) = 0
1117 CONTINUE
C INhorn = 1
  1122 NROW = 1
  NCOL = 1
  NSAV = K1
C IU( I ) = CODE FOR DEPLETION OF RESOURCES AT ORIGIN I
1111 IF ( ICOST(NROW, NCOL) = TSUM ) 1152, 1143, 1143
  IU( I ) = 1
1117 CONTINUE
1118 CONTINUE
  IU( J ) = 1
C IV( J ) = CODE FOR SATISFACTION OF REQUIREMENTS AT DESTINATION J
1111 IF ( ICOST(NROW, NCOL) = TSUM ) 1152, 1143, 1143
  IV( J ) = 1
1117 CONTINUE
C GENERATE INITIAL FEASIBLE SOLUTION - COLUMN MINIMA RULE
1142 IF ( ICOST(NROW, NCOL) = TSUM ) 1152, 1143, 1143
1143 NROW = NROW + 1
1145 IF ( NROW = NORIG ) 1142, 1142, 1163
1152 IF ( IU(NROW) = 1 ) 1143, 1143, 1143
C ISAV = VALUE OF CURRENT MINIMUM COST

C  NROW1 = ROW NUMBER OF CURRENT MINIMUM COST
1162  ISAV = ICOST( NROW, NCOL )
NROW1 = NROW
GO TO 1143
1163  IF ( IORIG( NROW1 ) = IDEST( NCOL ) ) 1114, 1173, 1164
C ORIGIN IS DEPLETED - DESTINATION IS NOT SATISFIED
1174  IPAR( NROW1, NCOL ) = IORIG( NROW1 ) + K1
1124  INDEST( NCOL ) = INDEST( NCOL ) - IORIG( NROW1 )
IU( NROW1 ) = 0
NROW = 1
ISAV = K1
GO TO 1142
C ORIGIN IS DEPLETED - DESTINATION IS SATISFIED
C ASSIGN AN EPSILON TYPE SHIPMENT PROPERLY
1177  IPAR( NROW1, NCOL ) = IORIG( NROW1 ) + K1
IU( NROW1 ) = 0
IV( NCOL ) = 0
NROW = 1
ISAV = K1
IF ( NCOL - NDEST ) 1183, 1211, 1211
1183  IF ( ICOST( NROW, NCOL ) = ISAV ) 1193, 1182, 1182
1193  IF ( IU( NROW ) = 1 ) 1182, 1123, 1182
1127  ISAV = ICOST( NROW, NCOL )
NROW1 = NROW
1197  NROW = NROW + 1
IF ( NROW = NOKIG ) 1183, 1183, 1174
C MAKE AN EPSILON TYPE SHIPMENT
1174  IPAR( NROW1, NCOL ) = K1
NCOL = NCOL + 1
1194  IF ( NCOL - NDEST ) 1194, 1194, 1211
1194  NROW = 1
ISAV = K1
GO TO 1142
C ORIGIN IS NOT DEPLETED - DESTINATION IS SATISFIED

POOR ORIGINAL COPY, BEST
AVAILABLE AS reduction
\begin{verbatim}
1144 IRAST( NROW, NCOL ) = IDEST( NCOL ) + K1
    IV( NCOL ) = 0
    INRTG( NROW ) = IDIRG( NROW ) = IDEST( NCOL )
1145 NCOL = NCOL + 1
    GO TO 1184

C DETERMINE SHADOW COSTS FOR CURRENT FEASIBLE SOLUTION
1211 DO 11n9 I = 1, NROW
    IV( I ) = VALUE OF U FOR ROW I
    IC( I ) = VALUE OF V FOR COLUMN J
C IV1( J ) = CODE FOR KNOWLEDGE OF V FOR COLUMN J
C IC( J ) = VALUE OF V FOR COLUMN J
    IV( J ) = 0
    IV1( J ) = 0
11n9 CONTINUE

110a CONTINUE
    DO 1110 J = 1, NDEST
    IV( J ) = VALUE OF V FOR COLUMN J
    IV1( J ) = CODE FOR KNOWLEDGE OF V FOR COLUMN J
    IV( J ) = 0
    IV1( J ) = 0
111a CONTINUE

1221 IU0( 1 ) = 1
    NROW = 1
    NCOL = 1
C CHECK FOR BASTS ELEMENT
1231 IF ( IRAST( NROW, NCOL ) = K1 ) 1232, 1241, 1241
C CURRENT ELEMENT IS A BASIS ELEMENT AT 1241
1241 IV( NCOL ) = ICOST( NROW, NCOL ) - IU( NROW )

1232 NCOL = NCOL + 1
1242 IF ( NCOL = NDEST ) 1231, 1231, 1252
1252 NROW = NROW + 1
1257 IF ( NROW = NROW ) 1272, 1272, 1261
1277 IF ( IU1( NROW ) - 1 ) 1282, 1252, 1282
1282 NCOL = 1
1292 IF ( IRAST( NROW, NCOL ) = K1 ) 1202, 1263, 1263
1297 NCOL = NCOL + 1
\end{verbatim}
IF ( NCOL = NDEST ) 1292, 1297, 1252
1269 IF ( IU ( NCOL ) ) 1273, 1262, 1273
1277 IU ( NRW ) = ICOST ( NRW, NCOL ) - IV ( NCOL )
    IU1 ( NRW ) = 1
1289 NCOL = 1
    GO TO 1231
1261 NRW = 1
1271 IF ( IU ( NRW ) ) 1291, 1282, 1281
1281 NRW = NRW + 1
1291 IF ( NRW = NOKIG ) 1271, 1271, 1213
C CALCULATE ALL CELL EVALUATIONS AND FIND LARGEST ONE
1213 ISAV = 0
    NRW = 1
1223 NCOL = 1
1233 IF ( TRAS ( NRW, NCOL ) = K1 ) 1234, 1243, 1243
1243 NCOL = NCOL + 1
1253 IF ( NCOL = NDEST ) 1233, 1233, 1254
1254 NRW = NRW + 1
1264 IF ( NRW = NOKIG ) 1223, 1223, 1274
1274 IF ( ISAV ) 1311, 1440, 1311
1284 IS1 = IU ( NRW ) + IV ( NCOL ) - ICOST ( NRW, NCOL )
1234 IF ( IS1 = ISAV ) 1243, 1243, 1245
1245 ISAV = IS1
    NRW1 = NRW
    NCOL1 = NCOL
    GO TO 1243
C THE CURRENT SOLUTION IS NOT OPTIMAL
C DETERMINE THE NEXT BASIC FEASIBLE SOLUTION
1311 K = ( NOKIG + NDEST ) * 2
C INET VECTOR STORES THE LABELS FOR THE ELEMENTS IN THE LOOP
DC 1126 I = 1, K
    INET ( I ) = 0
1124 CONTINUE
DC 1127 I = 1, NOKIG
IAET1(I) = 0
1127 CONTINUE
   DC 1128 J = 1, NDEST
   IAET2(J) = 0
1129 CONTINUE
   I = 1
1321 IAET(I) = NROW1
   IAET(I+1) = NCOL1
   NROW = NROW1
   NCOL = 1
   I = I + 2
1331 IF ( IBAS(NROW, NCOL) = K1 ) 1332, 1341, 1341
   CURRENT ELEMENT IS A RASIS ELEMENT AT 1341
   1341 IF ( NCOL = NCOL1 ) 1351, 1332, 1351
   IAET(I) = NROW
   IAET(I+1) = NCOL
   I = I + 2
   GO TO 1313
   1332 NCOL = NCOL + 1
   1333 IF ( NCOL = NDEST ) 1331, 1332, 1312
   ERROR TYPE 1 AT 1312
   NC RASIS ELEMENT IN ROW WHICH CONTAINS ENTERING ELEMENT
   1312 KE = 1
   GO TO 1481
1313 IAET2(NCOL) = 1
   NROW = 1
1327 IF ( TRAS(NROW, NCOL) = K1 ) 1333, 1324, 1324
1333 NROW = NROW + 1
1344 IF ( NROW = NOHIG ) 1323, 1323, 1353
1353 I = I - 2
1363 IF ( I ) 1315, 1315, 1373
1373 NROW = IAET(I)
   NCOL = IAET(I+1)
INET2( NCOL ) = 0
GO TO 1375

C  ERROR TYPE 2 AT 1315 = NO BASIS LOOP EXISTS
1315  KF = 7
GO TO 1481

C  CURRENT ELEMENT IS A BASIS ELEMENT AT 1324
1324 IF ( NROW = INET( I-2 ) ) 1334, 1333, 1334
1333 IF ( INET1( NROW ) ) 1353, 1344, 1353
1344 INET( I ) = NROW
INET( I+1 ) = NCOL
I = I + 2
1354 IF ( NROW = NROW1 ) 1364, 1355, 1364
1355 I = I + 2
GO TO 1353
1364 INET1( NROW ) = 1
NCOL = 1
1374 IF ( TRAS( NROW, NCOL ) = K1 ) 1375, 1384, 1384
1375 NCOL = NCOL + 1
1384 IF ( NCOL = NDEST ) 1374, 1374, 1371
1371 I = I - 2
1381 IF ( I ) 1315, 1315, 1382
1382 NROW = INET( I )
NCOL = INET( I+1 )
INET1( NROW ) = 0
GO TO 1333

C  DETERMINE THE VARIABLE WHICH LEAVES THE BASIS
1411 I = 3

POOR ORIGINAL COPY - BEST
AVAILABLE AT TIME FILMED
I\(\text{SAV} = K_2\)

1421 \(\text{NROW} = \text{INET}(1)\)
\(\text{NCOL} = \text{INET}(1 + 1)\)

1431 IF ( \(\text{TSAS}(\text{NROW, NCOL}) = \text{ISAV}\) ) 1432 \(1441, 1441\)

1432 \(\text{ISAV} = \text{IBAS}(\text{NROW, NCOL})\)
\(\text{NROW2} = \text{NROW}\)
\(\text{NCOL2} = \text{NCOL}\)

1441 IF ( \(\text{NCOL} = \text{NCOL1}\) ) 1451 \(1442, 1451\)

1451 I = I + 4

1461 IF ( I = K ) 1421 \(1471, 1471\)

C ERROR TYPE 3

C BASIS LOOP HAS MORE ELEMENTS THAN THERE ARE IN THE BASIS

1471 KF = 3
GO TO 1481

1442 IF ( \(\text{ISAV} = K_2\) ) 1443 \(1452, 1452\)

C ERROR TYPE 4

C NC ELEMENT IN THE BASIS LOOP LESS THAN 10**8

1452 KF = 4
GO TO 1481

1443 J = -1
\(\text{NROW} = \text{INET}(1)\)
\(\text{NCOL} = \text{INET}(1 + 1)\)
\(\text{IRAS}(\text{NROW, NCOL}) = \text{ISAV}\)
\(\text{ISAV} = \text{ISAV} - K_1\)

1444 \(\text{NROW} = \text{INET}(1)\)
\(\text{NCOL} = \text{INET}(1 + 1)\)

1453 IF ( \(\text{NROW} = \text{NROW2}\) ) 1463 \(1454, 1454\)

1454 IF ( \(\text{NCOL} = \text{NCOL2}\) ) 1463 \(1455, 1455\)

1455 \(\text{IRAS}(\text{NROW, NCOL}) = 0\)
GO TO 1473

1463 \(\text{IRAS}(\text{NROW, NCOL}) = \text{IBAS}(\text{NROW, NCOL}) + (J + \text{ISAV})\)

1473 J = -J
I = I + 2
IF ( NCOL = NCOL1 ) OPTIMUM SOLUTION AT 14A0

C

KF = 5
C

RFSTORE AVAILABILITY AND REQUIREMENTS VECTORS, NORIG, NDEST

DO 14A4 I = 1, NORIG
DO 14A3 J = 1, NDFST
IF ( IHAS(I,J) = KF ) 1483, 1484
14A2 IRAS(I,J) = IHAS(I,J) = KF
CONTINUE

DO 14A7 I = 1, NORIG
IRIG(I) = 0
DO 14A6 J = 1, NDEST
IRIG(I) = IRIG(I) + IRAS(I,J)
CONTINUE

DO 14A9 J = 1, NDEST
IDEST(J) = IDEST(J) + IRAS(I,J)
CONTINUE

NORIG = NORIG -1
NDEST = NDEST -1
RETURN
END