OPTIMIZATION OF A CARRIER ROUTING PROBLEM

by

HAROLD MERLIN COCHRAN

B. S., Kansas State University, 1965

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

Approved by:

Dr. Frank A. Tillman
Major Professor

TABLE OF CONTENTS

INTRODUCTION

Background and Purpose

The transportation system in the United States is one of the major contributors to the present high level of the national economy. One of the most important commercial components of this system is the trucking industry. To gain an understanding of this importance, some statistics are considered here [1]. At the present time approximately three out of every four tons of commercial goods being transported in the United States are carried at least part of the way by truck. This figure includes virtually all goods moving in local service, and about 38% of the nation's intercity freight tonnage. Today, trucks haul more than 29 billion intercity ton-miles (a ton-mile is a load of one ton carried a distance of one mile). The trucking industry generally spends (including wages) more than $42 billion a year to move these goods.

In addition to the trucking industry another important component of the transportation system in the United States is the school transportation system. Again, some statistics are considered here to quantify this importance [11]. School buses transport more than four times as many students each day as the total number of passengers carried in intercity travel by the nation's railroads and commercial bus lines combined. Total national expenditures for school transportation, which include operation and maintenance (but not purchase), of school buses, amounted to $486 million for the 1959-60 school year.

Considering these vast expenditures, improvements in the methods used by the transportation industry could conceivable result in the saving of considerable sums of money. This consideration applies not only to the trucking industry and school transportation systems, but to many other commercial and private carriers as well, including bus lines, railroads, and airlines.

One of the major areas in which improvement can be made is the routing of these carriers. Routing may be defined as the determination of paths or routes over which to dispatch or send passengers and goods. The typical method in use today is one of trial and error, and generally consists of looking at a map, picking out routes consistent with available carrier capacities, and then by trial and error attempting to find shorter routes.

This situation provides a fertile area for operations research analysis. Consequently, this paper presents a study of the methods and techniques which have been developed to solve problems of this nature. The purpose of this study is to evaluate current methods and establish which method is best for solving large scale carrier routing problems. In addition, modifications are proposed to solve the problem when the system is subject to multiple restraints.

Problem

Basically the problem is one of determining routes so that some objective is optimized and the restrictions on the system are satisfied. An example of this type of problem would be to determine routes where the distance traveled is a minimum with the following conditions being satisfied:

(1) All demands are supplied.

(2) The distance that can be traveled on each route is limited.

(3) The carriers may have different capacities.

Problems such as the above example have been entitled the "carrier routing problem", the solution of which is the principle objective of this paper.

SURVEY OF THE LITERATURE

The carrier routing problem may be regarded as a generalization of the classic traveling salesman problem.[1] Although there is an abundance of literature on the traveling salesman problem, very little can be found which directly relates to the carrier routing problem. The principle methods for solving the carrier routing problem are simulation, dynamic programming, the integer programming formulation of Miller, Tucker, and Zemlin [10], and the algorithms of Dantzig and Ramser [7] and Clarke and Wright [4]. These methods are discussed in the following section.

Approaches to the Problem

Work is being done at the present time in the department of Industrial Engineering at Kansas State University on a simulation approach to the carrier routing problem. This technique is a limited version of complete search. The routine randomly generates the order of stops while loading the carriers and checking against the available capacities. The lower bound for each route is then found using the traveling salesman algorithm developed by Little, Murty, Sweeney, and Karel [9].

A problem very similar in nature to the traveling salesman problem is known as the shortest route problem. It involves finding the path from one city to another such that the distance traveled is a minimum. One approach to this problem is the dynamic programming formulation proposed by Bellman [3].

---

[1]The traveling salesman problem might be described as follows: Find the shortest route for a salesman starting from a given city, visiting each of a specified group of cities, and then returning to his original point of departure.

Dynamic programming has also been applied to the carrier routing problem, in particular the school bus scheduling problem, by Tillman [12]. The sample problem used to illustrate the solution is a small scale (5 stop) problem involving 40 students and the equivalent of 3 buses. The sample problem and the method of solution are shown in the Appendix in the section on sample problems.

Miller, Tucker, and Zemlin [10] formulated the carrier routing problem as an integer programming problem and experimented with several models. However, the integer programming procedures which were known at the time of the experiments were not sufficiently developed to achieve solutions in a number of the experiments tried, although optimal solutions were achieved in two of the reported experiments. The authors stated that they were hopeful that the more efficient integer programming procedures which were being developed, notable by Gomory [8], at the time their experiments were being conducted, will, when applied to their model, yield a satisfactory algorithmic solution to the carrier routing problem.

Finally, two algorithmic methods of solution for the carrier routing problem appear in the literature. The first algorithm was developed by Dantzig and Ramser [7] and was published in 1958. The second algorithm, a modification of the first, was developed by Clarke and Wright [4], and was proposed in 1962.

Evaluation of the Proposed Methods of Solution

The integer programming formulation proposed by Miller, Tucker, and Zemlin [10] was studied as a method of solution for the carrier routing problem. However, the method was rejected for further development and use for two reasons. The first reason is the nature of the algorithms currently available for solving integer programming problems. These algorithms, although they theoretically should produce solutions, often fail to do so in actual applications. Secondly, the integer programming formulation did not include all the restrictions the author desired to include in the study, and it did not appear to be easily modified to include these restrictions.

The work on the simulation routine mentioned above has not yet reached a stage of development such that a statement can be made about its practicability for solving large scale problems. This method appears promising but gives no guarantee of obtaining an optimal solution.

Although dynamic programming gave an optimal solution to the small sample carrier routing problem, it also has its limitations. It appears that the dynamic programming method is too closely related to pure search, and the computational labor would become prohibitive on large scale problems as is the case with pure search.

The need for a more refined and sophisticated method of solution is obvious. The algorithms developed by Dantzig and Ramser [7] and by Clarke and Wright [4] provide this method of solution. Both algorithms, for this reason, are discussed in more detail in the following sections.

DANTZIG AND RAMSER METHOD

The first algorithm for solving the carrier routing problem was developed by Dantzig and Ramser [7] and was published as a paper in 1958. Their paper is concerned with the optimum routing of a fleet of gasoline delivery trucks between a bulk terminal and a large number of service stations supplied by the terminal. The shortest route between any two points and the quantity to be delivered to each station are assumed to be known quantities. Their purpose was to schedule trucks in such a way as to satisfy station demands and minimize the total miles covered by the truck fleet. Their algorithm will be discussed in some detail to facilitate the reader's understanding of the method proposed by Clarke and Wright [4], which is a modification of the above method.

Dantzig and Ramser regard the truck routing problem as a traveling salesman problem generalized to include the conditions that a number of loops must be determined such that all loops have one point in common (equivalent to the condition that the traveling salesman be required to return to his point of departure a number of times), and that specified deliveries be made at every point with the exception of the origin.

For simplicity of presentation, the authors make the assumption that only one product is to be delivered and that all trucks have the same capacity C. They state that the number of carriers does not enter the problem when they all have the same capacity. Even when carriers of different capacities are involved, or when a number of products are to be delivered to each service station or delivery point, the same mathematical model with minor variations may be used.

The basic idea of the method is to synthesize the solution into a number of stages of aggregation in which suboptimizations are carried out on pairs of points or groups of points. The deliveries q_i are first ordered in a sequence $q_1, q_2, \ldots, q_i, q_{i+1}, \ldots, q_n$ such that $q_i \leq q_{i+1}$ for any $i = 1, \ldots, n-1$. The maximum number of deliveries which can be made by a truck of capacity C for a given set of q_i's is represented by t, and is then determined such that

$$\sum_{i=1}^{t} q_i \leq C \text{ and } \sum_{i=1}^{t+1} q_i > C. \qquad [1]$$

The sequence $q_1, q_2, \ldots, q_t$ represents a feasible combination and therefore may be in the optimal solution. Hence, the number of aggregations to be used must allow the combination $q_1, q_2, \ldots, q_t$, or a maximum of t points, in the final aggregation. The number of points aggregated in the first stage is 2^1, in the second stage 2^2, and so on up to the final stage N where the number of points aggregated is 2^N. In the first stage pairs of points are joined, in the second stage pairs of pairs are joined, etc. Therefore, 2^N is the largest number of points aggregated in the N th and final stage of aggregation and may correspond to as many as t points. Thus the number of stages of aggregation N is determined such that

$$2^N = t \text{ or } N \cong \log_2 t. \qquad [2]$$

Assume that the number of stages of aggregation has been determined to be N = 2. In the first stage of aggregation only those points are allowed to pair up whose combined demand does not exceed 1/2C. As a result, in the second stage of aggregation any pair of points joined in the suboptimal first stage may be combined with any other pair of points joined in the first stage without exceeding truck capacity. If the number of stages of aggregation had been determined to be 3, in the first stage only those points whose combined load would not exceed 1/4C would be allowed to pair up. Thus in the second stage any pair of pairs would have a combined demand less than 1/2C, and hence the combined demand of the aggregations formed in the third stage would be less than the available capacity C.

It should be noted that if each delivery truck were scheduled to visit precisely two service stations and return to the terminal point, the total distance traveled by the trucks would be the constant sum of the distances from the terminal point to each service station plus the sum ~~if~~ of interpair distances, the distances between the two service stations served by each delivery truck. The only variables occurring in this situation are the interpair distances. Therefore, to minimize the total distance covered by all trucks, the sum of these interpair distances must be minimized. This is done by determining the optimum pairings corresponding to minimum interpair distances in each intermediate stage. In the final stage aggregations are determined such that the sum of all trip lengths is a minimum.

Dantzig and Ramser's formulation of the truck routing problem may be formally stated as follows:

(1) Given a set of n delivery points P_i ($i = 1, 2, \ldots, n$) to which deliveries are made from a terminal point, designated P_0.

(2) A symmetric distance matrix $[D] = [d_{ij}]$ is given which specifies the distance d_{ij} between every pair of points ($i, j = 0, 1, \ldots, n$). Since the matrix is symmetric, $d_{ij} = d_{ji}$ for all i, j.

(3) A delivery vector $Q = (q_i)$ is given which specifies the demand q_i at each delivery point P_i ($i = 1, 2, \ldots, n$).

(4) The capacity of all delivery trucks is the same and is represented by C, where $C >$ maximum q_i.

(5) If any two points P_i and P_j are paired, $x_{ij} = x_{ji} = 1$ ($i, j = 0,$

1,..., n), and if the points are not paired $x_{ij} = x_{ji} = 0$. Since every point P_i will be connected either to the terminal point P_0, or at most to one other point P_j, the following relation holds:

$$\sum_{j=0}^{n} x_{ij} = 1 \ (i = 1, 2, \ldots, n). \qquad [3]$$

By definition, $x_{ii} = 0$ for every $i = 0, 1, \ldots, n$.

(6) The problem is to find those values of x_{ij} which make the total distance

$$D = \sum_{i,j=0}^{n} d_{ij} \, x_{ij} \qquad [4]$$

a minimum under the conditions specified in [2] to [5].

A few general remarks about the algorithm should be made. Condition [5] limits the values of x_{ij} to be either 0 or 1, which puts this problem in the class of discrete variable problems. At the time Dantzig and Ramser were doing their work, no general method had been developed for solving discrete variable problems. Gomory's method [8] had just been proposed; however, it was considered to be at too early a stage of development to be applied to the problem at hand. It turns out that even with an integer programming algorithm the formulation required to prevent "looping", a sequence of cities not connected to the origin, generally expands the size of the problem beyond the limits where currently available algorithms can provide a solution to the problem. Therefore, the authors admitted the weaker condition

$$0 \leq x_{ij} \leq 1 \qquad [5]$$

and then suggest applying modified methods of linear programming to obtain "best solutions". (It should be noted that the authors did not elaborate on how this would be done.) Admitting this weaker condition may allow fractional values to appear in the solution, indicating the existence of alternative pairings of points or groups of points. The authors state that their experience has shown that the number of such alternative pairings will be small, so that the pairing yielding the least mileage can be readily determined by trial and error. The solution obtained in this manner will satisfy the requirement that x_{ij} be either 0 or 1. However, when the weaker condition is admitted, the solution obtained may no longer be the absolute optimum.

They felt that the solution obtained by their method approaches the absolute optimum as the number of points increases. Moreover, an estimate can be made on the error for the minimum distance D since $x_{ij} = 0$ or 1 lies between the "best solution" obtained by their method and the minimum satisfying $0 \leq x_{ij} \leq 1$.

At the start of the computational procedure all delivery points P_1, P_2, ..., P_n may be paired with the terminal point so that there will be n entries $X_{0,i} = 1$, where $i = 1, 2, \ldots, n$. These n entries constitute the basic set at the start of the computational procedure. During each iteration exactly one element of the basic set is eliminated and replaced by a new element or pairing. Therefore, the total number of basic entries remains constant during stage 1.

The starting solution, in which each delivery point is paired with the terminal point, is then improved by a series of rapid corrections. These rapid corrections are made by bringing into the solution non-basic entries which correspond to relatively small d_{ij} values. This procedure of making rapid corrections is repeated as long as non-basic entries with obviously low d_{ij} values are available.

After a sufficient number of pairs of points with small interpair distances have been brought into the solution, it will become increasingly difficult to bring in additional pairs of points without calculating the total distance in every case. Therefore, a criterion is needed to determine whether to accept or reject a non-basic variable for entry into the basic solution. This criterion is provided by what the authors have chosen to call a "delta-function", defined as

$$\delta_{ij}^{(n)} = \pi_i^{(n)} + \pi_j^{(n)} - d_{ij} \qquad [6]$$

where $\pi_i^{(n)}$ and $\pi_j^{(n)}$ are suitably determined constants characteristic for the n th iteration. By definition $\pi_i^{(n)}$ and $\pi_j^{(n)}$ are determined so that

$$\delta_{ij}^{(n)} = 0 \qquad [7]$$

for all d_{ij} corresponding to basic entries and

$$\delta_{ij} \gtrless 0 \qquad [8]$$

for non-basic entries. The delta function indicates how much the total distance D will decrease per unit increase of a non-basic entry x_{ij}. If $\delta_{ij} \leq 0$ for all non-basic variables, the particular set obtained at this point represents the "best solution". Otherwise, some non-basic variable corresponding to a $\delta_{ij} > 0$ is chosen for entry into the basic set. The standard criterion of the simplex method, that of selecting the non-basic variable corresponding to the largest δ_{ij}, is used to determine which variable will enter the basic set. When the delta function is negative or zero for all non-basic entries, no further improvement is possible and the first stage is concluded. For a more complete discussion of the π_i constants (simplex multipliers or prices) see Dantzig, Fulkerson, and Johnson [5,6].

In the second stage aggregation the d_{ij}, or minimum distances between points, are changed to the corresponding distances between first stage aggregations. The procedure for finding the combination of aggregates which yields minimum mileage is then the same as the one used for first stage aggregations. If it is determined that more than two stages of aggregation are needed, the procedure is repeated as many times as necessary.

If no fractional values appear in the final solution the problem is solved. If fractional values appear, a trial and error procedure is then used to decide which alternative corresponds to minimum mileage.

In their paper [7], Dantzig and Ramser show the solution to a sample problem involving deliveries to 12 service stations. The solution they obtain results in a total distance of 294 units. They believe, however, that a slightly different trip assignment with a total distance of 290 units is the true optimum solution to the problem. Therefore, their algorithm results in a "best solution" which comes very close to the true optimum for the particular numerical example used. They state that experience with the method has shown that similar results may be obtained in other numerical cases, particularly if the station demands do not differ too widely. They also conjecture that the difference between the distance for the "best solution" and that of the true optimum decreases as the number of station points increases.

CLARKE AND WRIGHT METHOD

General Remarks

As was mentioned previously, a modification was proposed in 1962 to the Dantzig and Ramser method by Clarke and Wright [4]. This method was chosen for further study for several reasons, these being:

(1) The procedure is simple but effective in producing a near optimal solution.

(2) It can be used to solve large scale practical problems with reasonable efficiency.

(3) It is well suited for programming on high speed digital computers.

(4) It has been found that this method gives better results than the Dantzig and Ramser method in a number of cases tested. This has been further substantiated by work done in this study, as is shown in the Appendix.

(5) Because of its simplicity the author was able to modify this approach to include additional conditions and restrictions which constituted a significant part of this study.

The formulation is similar to that proposed by Dantzig and Ranser [7]. In using Dantzig and Ramser's method, the restriction which allows only those customers whose combined load does not exceed $C/2^{N-1}$ to be linked in the first of N stages may also allow points to be linked that are far apart, and which may be virtually on opposite ends of a straight line through the terminal point. Although obviously long links may be excluded in the initial stages by rapid corrections, when two points become linked in an aggregation they remain aggregated. As a result, this places more emphasis on filling trucks to near capacity than on minimizing the total distance which must be traveled. This led to the search for a better method of solution.

Theoretical Aspects of the Problem

Included in Clarke and Wright's paper is a discussion of the theory behind their formulation of the problem. The discussion, however, does not appear to fully explain several of the major points in the theoretical development. This led to the development of the following discussion of the theoretical aspects of the problem.

Consider the feasible allocation of trucks to demand points shown in Figure 1 of Plate I. The demand points P_x, P_y, and P_z are initially linked only to the terminal point P_0. Three trucks, each traveling from the terminal point to a demand point and back to the terminal point, are allocated to haul the loads required by the demand points. The routes followed by the trucks are represented by solid lines and the direction of travel indicated by arrow heads. The total distance for all routes is:

$$2\, d_{0,x} + 2\, d_{0,y} + 2\, d_{0,z}. \qquad [9]$$

Linking the two demand points P_x and P_y on a route and severing one link from the terminal point to P_x and one link from the terminal point to P_y results in the allocation shown in Figure 2 of Plate I. The resulting "saving" in total distance over the initial allocation is:

$$d_{0,x} + d_{0,y} - d_{x,y}. \qquad [10]$$

The total distance for all routes now becomes:

$$d_{0,x} + d_{x,y} + d_{0,y} + 2\, d_{0,z}. \qquad [11]$$

Consider the allocation shown in Figure 3 of Plate I. This allocation is obtained from the allocation shown in Figure 2 by linking the demand

EXPLANATION OF PLATE I

Fig. 1. A feasible allocation of trucks to demand points.

Fig. 2. Allocation obtained by linking demand points P_x and P_y.

Fig. 3. Allocation obtained by linking demand points P_x and P_y and demand points P_y and P_z.

PLATE I

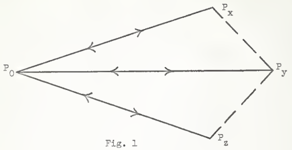


Fig. 1

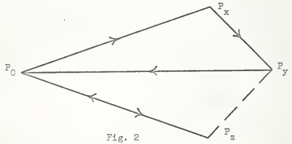


Fig. 2

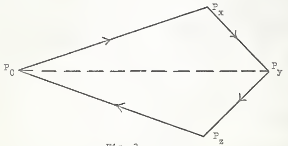


Fig. 3

points P_y and P_z, severing the link from P_0 to P_y, and severing one of the links from P_0 to P_z. The resulting saving in distance is:

$$d_{0,y} + d_{0,z} - d_{y,z}. \qquad [12]$$

The resulting total distance for all routes is:

$$d_{0,x} + d_{x,y} + d_{y,z} + d_{0,z}. \qquad [13]$$

The saving in total distance which would result from the linking of any two demand points which are linked to the terminal point may be calculated as shown above in equations 10 and 12. This saving is calculated for each pair of demand points in the problem. The maximum of these savings is selected that would, if linked, produce feasible routes consistent with truck availabilities and capacities. These two demand points are now linked and the next highest saving is determined and the procedure repeated.

Whenever a demand point is linked to two others (not P_0) it will not be considered again for linking. As a result of this, the only links that will be severed will be those of points linked to the terminal point. Thus the saving from linking two general demand points P_y and P_z is expressed as the following relation:

$$d_{0,y} + d_{0,z} - d_{y,z}. \qquad [14]$$

Computational Procedure

The computational procedure used by Clarke and Wright will now be explained. The procedure is listed and explained step by step so that it can be easily referenced in the next section when the modified procedure is explained.

Step 1.

The first step in the computational procedure is the assigning of identification numbers to the demand points so that they can be more easily referenced and worked with during the computational procedure. The demand points are labeled P_i $(i = 1, 2, \ldots, M)$, where M is the number of demand points. The demand points should first be ordered such that demand point 1 is closest to the terminal point, demand point 2 is the next closest point, and so on.

Step 2.

The second step in the computational procedure involves the initial allocation of trucks to demand points. It is assumed that the values of the demands q_i $(i = 1, 2, \ldots, M)$ are such that one truck can carry q_i. If this assumption does not hold, trucks with the highest capacity available are allocated to the demand point. The remainder of the demand, which will be an amount less than a truckload of the highest capacity, is then considered in the initial allocation, and the full truckloads are excluded from further consideration. After this has been done, all demands which will enter into the computation will be such that $q_j \leq C_n$ $(j = 1, 2, \ldots, M)$, where C_n is the highest available truck capacity. For convenience of computation, the truck capacities C_i are ordered such that $C_{i-1} < C_i$ $(i = 2, \ldots, n)$, where n is the number of different available capacities.

Step 3.

The numerical example used by Clarke and Wright is the same one used by Dantzig and Ramser. This numerical example is also used in this section and will be referred to as sample problem 1 throughout the remainder of this paper.

The third step in the computational procedure includes the calculation of the savings matrix and setting up the initial computational matrix. Since the distance matrix is symmetrical, it is recommended that a half matrix be used for hand computation. The format for this matrix as well as the necessary matrix values for sample problem 1 are shown in Table 1. The entries in the lower right-hand corner of each matrix cell (y:z) are the appropriate distances $d(P_y:P_z)$ between demand points P_y and P_z by the shortest practicable route. The entries in the lower left-hand corner of each cell are the savings. These values are calculated as described above, i.e., for cell (y:z) with $y,z \geq 1$, and $y \neq z$, the value of the saving is $d_{0,y} + d_{0,z} - d_{y,z}$. A column vector $Q = (Q_1, Q_2, \ldots, Q_M)$ is added on the left-hand side of the matrix. At the start of the computational procedure, the values entered in this vector are the loads q_i required by demand point P_i ($i = 1, 2, \ldots, M$). The remaining cell entries, designated as $t_{y,z}$, will always be either 0, 1, or 2. If the two demand points P_y and P_z are linked on a truck's route, $t_{y,z} = 1$ will be entered in the appropriate cell. A demand point served exclusively by a truck will have a corresponding cell value $t_{y,0} = 2$. The cell entry for each pair of demand points not linked and $y,z > 0$ will be $t_{y,z} = 0$.

Step 4.

The initial basic solution is now entered in the matrix set up in step

Table 1. Distance matrix showing distances in lower right-hand corner of each cell, saving in lower left-hand corner of each cell, initial basic solution values in upper left-hand corner of each cell, and the initial Q vector on left-hand side of matrix.

Q	P_0												
1200	2 9	P_1											
1700	2 14	18 5	P_2										
1500	2 21	18 12	28 7	P_3									
1400	2 23	10 22	20 17	34 10	P_4								
1700	2 22	10 21	20 16	22 21	26 19	P_5							
1400	2 25	10 24	16 23	16 30	20 28	38 9	P_6						
1200	2 32	10 26	20 26	26 27	30 25	44 10	50 7	P_7					
1900	2 36	10 35	20 30	20 37	24 35	42 16	50 11	58 10	P_8				
1800	2 38	10 37	16 36	16 43	20 41	38 22	50 13	54 16	68 6	P_9			
1600	2 42	10 41	20 36	32 31	36 29	44 20	50 17	64 10	72 6	68 12	P_{10}		
1700	2 50	10 49	20 44	34 37	42 31	44 28	50 25	64 18	72 14	76 12	84 8	P_{11}	
1100	2 52	10 51	20 46	34 39	46 29	44 30	50 27	64 20	72 16	70 20	84 10	92 10	P_{12}

Table 2. Allocation table in form suggested by Clarke and Wright.

Trucks	Up to 4000 gal.	Over 4000 gal.	Over 5000 gal.	Over 6000 gal.
Available	∞	7	4	0
Allocated	12	0	0	0

3. The initial basic solution as shown in Table 1 is entered as $t_{y,0} = 2$ ($y = 1, 2, \ldots, M$), the values being shown in the upper left-hand corner of each cell. Since a demand point may be linked to at most two other points, one of which may be the terminal point P_0, the following relationship always exists:

$$\sum_{\substack{z=0 \\ \text{with} \\ y=k}}^{k-1} t_{y,z} + \sum_{\substack{y=k+1 \\ \text{with} \\ z=k}}^{M} t_{y,z} = 2 \quad (k = 1, 2, \ldots, M), \qquad [15]$$

i.e., the sum along the k th row plus the sum along the k th column must always equal 2.

Step 5.

The initial allocation table is set up in step 5. Since in the solution some trucks may be only partially loaded, the number of trucks of the smallest capacity, x_1, needs to be sufficiently large to insure that all demands will be allocated. For purposes of computation, it is assumed that an unlimited number of trucks of the smallest capacity are available, and this value is set equal to ∞.

Table 2 shows the number of available trucks above each capacity level and the number of trucks already allocated. In the numerical example shown, it is assumed that there is an unlimited supply of trucks of capacity 4000 gallons, 3 trucks of capacity 5000 gallons, and 4 trucks of capacity 6000 gallons.

The now completed tables, Table 1 and Table 2, show the initial feasible solution.

Step 6.

The sixth step in the computational procedure is a search of the rows and columns of the half matrix of Table 1 for the maximum saving. Only those savings corresponding to links which are still eligible to be formed should be included in the search.

If there are two or more equal maxima in the search, one of these is selected randomly.

Step 7.

Test the maximum saving found in step 6 to see if the conditions listed below are satisfied. If the maximum saving occurs in cell (y:z):

(1) $t_{y,0}$ and $t_{z,0}$ must be greater than zero. If these values are greater than zero, demand points P_y and P_z are still linked to the origin and these links are therefore eligible to be severed.

(2) Demand points P_y and P_z are not already allocated on the same truck run. This restriction is necessary to prevent "looping", a situation in which routes or "loops" are formed which do not include the terminal point.

(3) Amending the allocation table (Table 2) by removing the trucks allocated to loads Q_y and Q_z and adding a truck to cover the load $Q_y + Q_z$ would not cause the trucks allocated to exceed the trucks available in any column of the allocation table.

If one or more of the conditions is not satisfied, the maximum saving being tested is excluded from further consideration and step 6 repeated.

Step 8.

If all of the conditions listed in step 6 are satisfied, $t_{y,z}$ is set equal to 1 and the other values of $t_{i,j}$ are amended so that relation [15] holds. This may be accomplished very easily by reducing the values of $t_{y,0}$ and $t_{z,0}$ by 1.

Step 9.

The Q vector must then be amended in two ways. First, each Q_j corresponding to $t_{j,0} = 0$ is itself set equal to zero and second, each Q_j corresponding to a demand point allocated on the new route is set equal to the total demand for all points on the route.

Step 10.

The allocation table is then changed to correspond to the new allocation. This consists of removing the trucks allocated to loads Q_y and Q_z and adding a truck to cover the load $Q_y + Q_z$.

Step 11.

The first iteration is now completed. If more links are possible repeat the procedure from step 6 on.

Step 12.

If no more links are possible, i.e., no maximum saving will satisfy all conditions, the final solution has been found. The final allocation of demand points to routes and the exact order of visitation of demand points may then be determined from the $t_{i.j}$ half matrix, and the final allocation of available trucks may be obtained from the final allocation table. The distances for each route and the total distance for all routes may then be calculated by referring to the original distance matrix.

The computational procedure will now be discussed in conjunction with sample problem 1 to facilitate the reader's understanding. Steps 1 through 5, in which the problem is set up as shown in Tables 1 and 2, are adequately discussed. The emphasis will be put on discussing the computational aspects of the first iteration starting at step 6. The reader may wish to refer to the computational matrix and allocation table for the first iteration which are shown as Tables 3 and 4, respectively.

Step 6.

The maximum saving of the half matrix of Table 1 is 92, found in cell (12:11). Therefore, $y = 12$ and $z = 11$. No equal maxima are involved in this case.

Step 7.

The maximum saving is then tested to see if it meets all conditions.

(1) $T_{12,0}$ and $t_{11,0}$ are both equal to 2 and are therefore greater than zero.

(2) Demand points P_{12} and P_{11} have not already been allocated on the same truck run, since each is initially on a separate truck run from the terminal point.

(3) Amending the allocation table (Table 2) by removing the trucks allocated to loads Q_{12} and Q_{11}, 1100 gallons and 1700 gallons, respectively, would create an allocation of 10 trucks in the "up to 4000 gal." column, with all other columns remaining as they were. The load $Q_{12} + Q_{11}$, or 2800 gallons, would require a truck

Table 3. Computational matrix after completion of first iteration.

Q	P_0												
1200	2	P_1											
1700	2		P_2										
1500	2			P_3									
1400	2				P_4								
1700	2					P_5							
1400	2						P_6						
1200	2							P_7					
1900	2								P_8				
1800	8									P_9			
1600	2										P_{10}		
2800	1											P_{11}	
2800	1											1	P_{12}

Table 4. Allocation table after completion of first iteration.

Trucks	Up to 4000 Gal.	Over 4000 gal.	Over 5000 gal.	Over 6000 gal.
Available	∞	7	4	0
Allocated	11	0	0	0

having a capacity of 4000 gallons, resulting in an allocation of 11 trucks in the "up to 4000 gal." column. This does not cause the trucks allocated to exceed the trucks available in any column of the allocation table.

All of the conditions are satisfied, therefore procede to step 8.

Step 8.

$T_{12,11}$ is set equal to 1 and the values of $t_{12,0}$ and $t_{11,0}$ are reduced by 1, making them each equal to 1.

Step 9.

The Q vector is amended by setting Q_{12} and Q_{11} equal to 2800 gallons, the total demand for all points on the new route formed by linking demand points P_{12} and P_{11}.

Step 10.

The allocation table is then changed to correspond to the values obtained in step 7, as shown in Table 4.

Step 11.

The first iteration is now completed. The resulting computational matrix is shown in Table 3. More links are possible since not all of the eligible savings have been tested for entry into the basic solution. Therefore, the procedure would be to return to step 6 and select the next eligible maximum saving.

The computational procedure described above has been programmed for the IBM 1620 computer. A complete description of the computer program is found in the Appendix of this paper. As explained there, it is possible to obtain the values of the computational matrix and allocation table for each iteration. Sample problem 1 has been solved in this manner, and the remainder of the iteration-by-iteration solution of sample problem 1 is found in the Appendix in the section on sample problems.

The final solution for sample problem 1 is shown in Table 5 and Table 6. An explanation is now given of how to read the sequence of stops, i.e., the final allocation of routes, from Table 5.

The procedure starts by checking the values of $t_{i,0}$ ($i = 1, 2, \ldots, M$) until one is found for which $t_{i,0} \neq 0$. A $t_{i,0} = 0$ indicates that stop i is linked to two points other than the origin. A $t_{i,0} = 1$ indicates that stop i is linked to the origin and signifies the beginning or ending of a route. A $t_{i,0} = 2$ indicates that stop i is served exclusively by a truck.

When a $t_{i,0} = 1$ is found the stop, other than the origin, which is linked to stop i must be found. This is done by searching row i and/or column i until a $t_{i,j} = 1$ is found. The row or column for which this is the case is the stop linked to stop i. As an example, $t_{1,0} = 1$ in sample problem 1. The other stop linked to stop 1 is found by searching column 1 until a $t_{i,1} = 1$ is found, the value of i for which this is true being i = 2. Therefore, stop 2 is linked to stop 1. The stop, other than stop 1, which is linked to stop 2 is then found by searching row 2 and column 2 until the value $t_{3,2} = 1$ is found, indicating that stop 3 is linked to stop 2. The remainder of the stops allocated on this route are then found in a similar manner, as are the stops allocated on other routes.

The final routes and distances for sample problem 1 are listed below. The route numbering corresponds to that used by Clarke and Wright. The numbering for the computer solution differs slightly as is explained in the discussion of the computer program in the Appendix.

Table 5. Computational matrix after completion of final iteration.

Q	P_0												
5800	1	P_1											
0	0	1	P_2										
0	0		1	P_3									
5800	1			1	P_4								
1700	2					P_5							
5100	1						P_6						
5600	1							P_7					
0	0						1		P_8				
5100	1								1	P_9			
0	0							1			P_{10}		
0	0										1	P_{11}	
5600	1											1	P_{12}

Table 6. Allocation table after completion of final iteration.

Trucks	Up to 4000 gal.	Over 4000 gal.	Over 5000 gal.	Over 6000 gal.
Available	∞	7	4	0
Allocated	1	3	3	0

(1) Route 1: $P_0 - P_7 - P_{10} - P_{11} - P_{12} - P_0$ having a distance of 112 miles and requiring a truck having a capacity of 6000 gallons.

(2) Route 2: $P_0 - P_6 - P_8 - P_9 - P_0$ having a distance of 80 miles and requiring a truck having a capacity of 6000 gallons.

(3) Route 3: $P_0 - P_1 - P_2 - P_3 - P_4 - P_0$ having a distance of 54 miles and requiring a truck having a capacity of 6000 gallons.

(4) Route 4: $P_0 - P_5 - P_0$ having a distance of 44 miles and requiring a truck having a capacity of 4000 gallons. Note that this demand point is served exclusively by a truck and therefore $t_{5,0} = 2$ in the final computational matrix.

The total distance for the routes listed above is 290 miles, believed by Dantzig and Ramser to be the true minimum mileage solution.

Clarke and Wright state that although the improvement in this example is slight, a problem involving 30 demand points resulted in an improvement of 17 per cent over the Dantzig and Ramser method.

It is further suggested that, although the solution gives the order of visitation of demand points, it may be beneficial to solve the traveling salesman problem for each truck in the final allocation to determine the true optimum order of visiting.

The Algorithm Summarized

The basic steps in the computational procedure will now be listed to provide a summary of the algorithm.

Step 1. Order the demand points according to their distance from the origin such that demand point 1 is closest to the origin, demand point 2 is next closest, and so on. Label the demand points P_i $(i = 1, 2, \ldots, M)$.

Step 2. Assign an initial allocation of one truck to each demand point if the allocation is feasible. If the allocation is infeasible split the invalid demands to produce a feasible allocation.

Step 3. Calculate the savings.

Step 4. Enter the initial basic solution in the initial computational matrix (Table 1).

Step 5. Set up the initial allocation and availability table as shown in Table 2.

Step 6. Find the maximum eligible saving in Table 1. If there are two or more equal maxima, choose one of them randomly.

Step 7. Test the maximum saving found in step 6 to see if it meets conditions 1 through 3 listed above. If any one or more of these conditions is not satisfied exclude the maximum saving from further consideration and return to step 6.

Step 8. If all of conditions 1 through 3 are satisfied, set $t_{y,z} = 1$ for the cell corresponding to the maximum saving and amend the rest of the $t_{i,j}$ values so that relation [15] holds.

Step 9. Amend the Q vector.

Step 10. Change the allocation table (Table 2) to correspond to the new allocation.

Step 11. Repeat the procedure from step 6 if more links are possible.

Step 12. If no more links are possible, determine the allocated routes, their respective distances, and the total distance for all routes.

MODIFIED CLARKE AND WRIGHT METHOD FOR MULTIPLE RESTRAINTS

Modifications

This section contains a discussion of the modifications which can be made in the Clarke and Wright method to incorporate additional restraints on the system and to improve the computational procedure. In addition, limitations of the method are pointed out and discussed, and possible procedures for overcoming these limitations are suggested. The modifications are discussed in reference to the steps of the previous method so that the two methods can easily be compared.

Step 1 of the modified procedure is the same as step 1 in the previous method.

Step 2.

The second step in the computational procedure involves the initial allocation of trucks to demand points. It is assumed that the values of the loads required at each demand point are such that an initial allocation of one truck to each demand point is possible. In the case in which one or more demands are larger than the largest available truck an allocation can still be made. This is done by splitting the large load into two (or more) full truckloads of the highest capacities available and only considering the remainder of that load, an amount less than a truck load of the highest capacity. Thus, all loads considered in the problem will be such that $q_i \leq C_n$ $(i = 1, 2, \ldots, M)$.

The solution of example carrier routing problems has pointed up a limitation of the modified Clarke and Wright algorithm occurring in the allocation of carriers to demand points. This limitation will now be discussed and a possible means of overcoming the limitation is also suggested.

An obvious difficulty occurs when there are not enough large trucks available to assign one truck to each demand point. This situation might be remedied by combining demands until they become such that an allocation of one truck to each demand can be made. This should be done before the computational procedure is started. This procedure, however, might cause an otherwise best solution to become a less favorable one.

A difficulty may occur even when there are enough large trucks available to assign one truck to each demand point. A number of the larger trucks may have been assigned small loads in the initial allocation to insure an allocation of one truck to each demand point. These large trucks could be put to better use hauling larger combined loads since the algorithm emphasizes combining small loads into larger ones. This combining of small loads will allow the trucks to which these loads were initially assigned to become available for further use. However, the algorithm does not include a provision for reassigning the small loads initially assigned to large trucks to the small trucks made available by the combination of loads. Therefore, it is suggested that the algorithm be modified to include a reassigning of trucks to loads each time two loads are combined.

The modified allocation procedure is as follows:

(1) Arrange the loads in order of increasing size with the smallest load first.

(2) Each load, starting with the smallest, is then assigned to the smallest available truck which can haul the load.

After each iteration, in which two smaller loads are combined into a larger one, the above procedure is repeated. This procedure should also be used when assigning the initial allocation of trucks to demand points.

It is believed that this modified procedure will make better use of the available trucks in certain cases, thus resulting in a better solution in terms of total miles traveled.

The computer program includes this modified allocation procedure and its use is explained in the discussion of the computer program included in the Appendix. It should be noted that the use of the modified procedure is optional since it is useful only in certain cases and it requires more running time than the normal allocation procedure.

Steps 3 and 4 of the modified procedure are the same as the corresponding steps in the previous method.

Step 5.

For ease of computation and to avoid confusion during the computational procedure, it is suggested that the form shown in Table 7 may be used. In this table the actual values of the capacities and availabilities are shown rather than the cumulative availabilities shown in Table 2.

For purposes of computation, Clarke and Wright set the number of trucks of the smallest capacity equal to ∞. However, the number of trucks of the smallest capacity available may be limited. It is believed that economic considerations will reduce the number of trucks of this capacity which will be allocated in the final solution to a value very nearly equal to or less than the actual number available. However, the problem of requiring more trucks than are available can be avoided in the situation where the largest demand is less than the smallest available truck capacity by adding a "dummy"

Table 7. Allocation table in form suggested by the author.

Trucks	4000 gallons	5000 gallons	6000 gallons	Capacity = ∞
Available	∞	3	4	0
Allocated	12	0	0	0

Table 8. Allocation table with "dummy" capacity of 1900 gallons.

Trucks	1900 gallons	4000 gallons	5000 gallons	6000 gallons	Capacity = ∞
Available	∞	2	3	4	0
Allocated	12	0	0	0	0

Table 9. Allocation table with mileage restrictions.

Trucks	4000 gallons	5000 gallons	6000 gallons	Capacity = ∞
Available	∞	3	4	0
Allocated	12	0	0	0
Distance restriction (miles)	104	104	104	0

capacity. If this is not the case, the procedure suggested by Clarke and Wright should be used. The "dummy" capacity, if used, should be set equal to the largest demand. It can then be assumed that an infinite number of trucks of this capacity are available, and the correct number of trucks with the smallest capacity can then be used in the computation. This procedure is demonstrated in sample problem 1 as is discussed below.

A word of caution is added on the use of the modified procedure explained above, especially if the demands vary widely from demand point to demand point. Consider a situation in which the sum of the demands for two or more demand points is considerably smaller than the largest demand and the capacity of the smallest truck. If a "dummy" capacity equal to the largest demand were used in this situation, it is conceivable that this "dummy" capacity could be allocated in the final solution. If this is true, a check should be made on the allocation of trucks with the smallest capacity. The trucks allocated to the "dummy" capacity can in most cases be transferred to the smallest capacity available without exceeding truck availabilities. It is suggested that this procedure be used when possible.

Table 8 is the initial allocation table for example problem 1 with a "dummy" capacity of 1900 gallons added. As was discussed previously, this "dummy" capacity is assigned the initial allocation of 12 trucks. This allows the actual number of trucks of the smallest capacity, in this case assumed to be 2 trucks of 4000 gallons each to be used in the computation. It is possible to use the "dummy" capacity in this case since the largest demand is less than the smallest available truck capacity.

Step 6.

It was stated in the previous section that Clarke and Wright suggest selecting randomly one of two or more equal maxima in the search. It is noted that the other equal maxima should also be tested for entry into the basic solution. The equal maxima should be tested in row by row order starting with the saving in row 2 and column 1. After a saving has been tested it need not be considered again, regardless of whether or not it was entered into the basic solution. It should be noted that the computer program includes this modification.

Step 7.

It is in this step that the procedure is modified to include multiple restraints. Thus in addition to satisfying the first three conditions of step 7 in the previous procedure, additional restraints on the system can be incorporated. In particular, if the mileage restriction discussed previously is to be included in the problem formulation, the maximum saving would also be subject to the following condition in step 7:

(4) The total mileage of the new route formed by the addition of demand points P_y and P_z must be less than or equal to the mileage restriction for the truck capacity necessary to haul the load $Q_y + Q_z$.

If this additional condition is included in the problem formulation, the allocation table is modified. The allocation table for sample problem 2, which is a modification of sample problem 1, is shown in Table 9. Table 9 is Table 7 with a row added to include the restriction that a truck of a given capacity can travel no more than a specified number of miles on a route. It has been assumed, as shown in Table 9, that all trucks can travel up to 104 miles per route.

Additional restrictions such as time spent on a route could also be included in step 7 in the same manner.

The computer program mentioned previously was modified to include a restriction on the number of miles which can be traveled by each truck. Sample problem 2 with the mileage restriction was solved using this computer program and the solution is included in the sample problem section of the Appendix.

Steps 8, 9, and 10 of the modified procedure are the same as the corresponding steps in the previous method.

Step 11.

Step 10 completes each iteration. If there are more savings to check for entry into the basic solution reassign the trucks to the loads following the procedure outlined under step 2 above. After this has been done return to step 6 as usual.

Step 12 of the modified procedure is the same as step 12 in the previous method.

The Modified Algorithm Summarized

The computational procedure for the modified algorithm may now be summarized as follows:

Step 1. Order the demand points according to their distance from the origin such that demand point 1 is closest to the origin, demand point 2 is next closest, and so on. Label the demand points P_i $(i = 1, 2, \ldots, M)$.

Step 2. Assign the initial allocation following the procedure outlined under step 2 in the discussion of the modified algorithm.

Step 3. Calculate the savings.

Step 4. Enter the initial basic solution in the initial computational matrix (Table 1).

Step 5. Set up the initial allocation and availability table as shown in:

(1) Table 7 if a "dummy" capacity is not used.

(2) Table 8 if a "dummy" capacity is used.

(3) Table 9 if additional restrictions are included in the problem formulation.

Step 6. Find the maximum eligible saving in Table 1. If there are two or more equal maxima, follow the procedure outlined under step 6 in the discussion of the modified algorithm.

Step 7. Test the maximum saving found in step 6 to see if it meets conditions 1 through 4. If any one or more of these conditions is not satisfied exclude the maximum saving from further consideration and return to step 6.

Step 8. If all of conditions 1 through 4 are satisfied, set $t_{y,z} = 1$ for the cell corresponding to the maximum saving and amend the rest of the $t_{i,j}$ values so that relation [15] holds.

Step 9. Amend the Q vector.

Step 10. Change the allocation table (Table 7, 8, or 9) to correspond to the new allocation.

Step 11. If there are more savings to check reassign the trucks to the loads and return to step 6.

Step 12. If no more links are possible, determine the allocated routes, their respective distances, and the total distance for all routes

Discussion of Sample Problems

A number of sample problems have been solved both by hand and using the computer program written for the solution of carrier routing problems. These problems and their solutions are included in the Appendix, and consist of the following:

(1) A complete solution of the 12 stop sample problem referred to as sample problem 1. This problem solution includes the computational matrix and revised allocation table for each iteration, and was obtained using the computer program. Note that the destination identification numbers in the computer output do not correspond to those used in the previous discussion of sample problem 1. This discrepancy and the reason for it are explained in the discussion of the computer program in the Appendix. The two solutions may be easily compared, however, as the destination numbers in the computer used in the previous discussion.

(2) Sample problem 1 with the mileage restriction for all trucks set at 104 miles. Demand point P_{12}, which is 52 miles from the terminal point, may be served exclusively by a truck in the final solution. Therefore, the distance restriction must not be less than 104 miles to admit this possibility in the final solution and insure obtaining a feasible solution. As in sample problem 1, the problem solution includes the computational matrix and revised allocation table for each iteration, and was obtained using the computer program. The problem also includes a "dummy" capacity of 1900 gallons to show the use of a "dummy" capacity in an actual example. It should be noted that a truck of the "dummy" capacity was allocated in the

final solution. As discussed previously, this allocation can be transferred to the smallest capacity, 4000 gallons, since no trucks of this capacity were allocated and 2 were available.

(3) The 5 stop problem referred to previously as having been solved by dynamic programming. The dynamic programming solution is shown and explained in addition to the computer output. The modified Clarke and Wright method results in the same optimal solution as the dynamic programming method.

*(4) An actual 13 stop problem involving the routing of feed delivery trucks. The modified Clarke and Wright method gave a total distance of 1433 miles using 4 trucks as compared to the routing in use by the company which involved a total distance of 1474 miles and 5 trucks. Although the improvement in total mileage is slight, the use of one less truck could save the company a considerable amount of money.

*(5) An actual 33 stop problem involving the routing of feed delivery trucks. In this example all trucks had the same capacity. Therefore, a number of demand points requiring full truck loads were eliminated from the problem before a solution was attempted. The resulting problem which was solved using the computer program, involved 25 demand points. The modified Clarke and Wright method gave a total distance of 1468 miles involving 14 trucks. This was a saving of 119 miles and 2 trucks, a substantial improvement over

---

*Data for these sample problems was graciously furnished by the Grain and Feed Marketing Project of the Agricultural Experiment Station at Kansas State University.

the method in use by the company. It should be noted that the solution to this problem obtained using the Dantzig and Ramser method resulted in the same total distance as the method in use by the company, providing further justification for the use of the modified Clarke and Wright method.

CONCLUSIONS

Because of only recent interest in the carrier routing problem, a limited number of methods for solving the problem are currently available. A summary of these methods and the reasons for their acceptance or rejection is now given.

An integer programming formulation of the generalized traveling salesman problem was studied as a possible method of solution for the carrier routing problem. It was rejected because of the nature of the algorithms currently available for solving integer programming problems, and it did not appear to be easily modified to include the additional restrictions the author desired to include in the study.

Another technique considered as a possible method of solution for the carrier routing problem is a simulation routine which, for purposes of the study, was thought of as a limited version of complete search. The technique was rejected because the work on it has not yet reached a stage of development such that a statement can be made about its practicability for solving large scale problems.

Dynamic programming was also considered as a possible method of solution for the carrier routing problem, and an actual small scale problem was solved using a dynamic programming approach. However, it was also rejected because it was felt that the dynamic programming method was too closely related to pure search, and the computational labor would become prohibitive on large scale problems as is the case with pure search.

Finally, two algorithmic methods of solution for the carrier routing problem were studied. The first algorithm was developed by Dantzig and Ramser [7]. The second algorithm, a modification of the first, was developed by Clarke and Wright [4], and was found to give better results than the Dantzig and Ramser method in a number of cases tested. Therefore, the Dantzig and Ramser method was rejected for further modification and study.

The Clarke and Wright method was then modified to incorporate multiple restraints and to improve the computational procedure. A modified allocation procedure which will make better use of the available carriers was also suggested. This modification of the Clarke and Wright method is practicable and efficient for solving large scale problems. Even though it does not guarantee an optimal solution, it appears to be the "best" method available at the present time for the solution of practical large scale routing problems.

Several sample carrier routing problems were solved using the modified Clarke and Wright method. Two of these were actual problems involving the routing of feed delivery trucks. In the first problem the modified method gave a saving of 41 miles and 1 truck over the method in use by the company, and in the second problem gave a saving of 119 miles and 2 trucks over the method in use by the company. The modified method was never beaten in the solution of the sample problems, although it was tied in the 5 stop problem (sample problem 3) by dynamic programming.

Much work remains to be done on the carrier routing problem. The need for an algorithm which will give a guaranteed optimal solution is obvious. A promising step in this direction is the algorithm for solving integer programming problems developed by Gomory [8]. It appears that this technique, when further developed and applied to the carrier routing problem, may provide an optimal method of solution.

ACKNOWLEDGMENTS

The writer wishes to acknowledge the faculty and staff of the Department of Industrial Engineering for their aid in the development of this thesis. He is particularly indebted to his major professor, Dr. Frank A. Tillman, for his initial inspiration and continual advice and guidance, both on this thesis and on numerous other topics. Thanks go also to the Agricultural Experiment Station for supplying the sample problems (see the footnote on page 40 of this paper), and to the writer's wife, Sherry, for her fine help in the preparation and typing of this manuscript.

REFERENCES

1. ______, "Motor Transportation," Encyclopedia Britannica, Vol. 15, 938-944 (1965).

2. Barachet, L. L., "Graphic Solution of the Traveling-Salesman Problem," Operations Research Journal of America, Vol. 5, 841-845 (December 1957).

3. Bellman, R., "On a Routing Problem," Quarterly Journal of Applied Mathematics, Vol. XVI, No. 1, 87-90 (April 1958).

4. Clarke, G., and Wright, J. W., "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research Journal of America, Vol. 12, No. 4, 568-581 (July-August 1964).

5. Dantzig, G. B., Fulkerson, D. R., and Johnson, S. M., "On a Linear-Programming Combinatorial Approach to the Traveling-Salesman Problem," Operations Research Journal of America, Vol. 7, 58-66 (January-February 1959).

6. Dantzig, G. B., Fulkerson, D. R., and Johnson, S. M., "Solution of a Large-Scare Traveling-Salesman Problem," Operations Research Journal of America, Vol. 2, 393-410 (November 1954).

7. Dantzig, G. B., and Ramser, J. H., "The Truck Dispatching Problem," Management Science, Vol. 6, Nol. 1, 80-91 (October 1959).

8. Gomory, R. E., "Essentials of an Algorithm for Integer Solutions to Linear Programs," Bulletin American Math Society, Vol. 64, No. 5 (1958).

9. Little, J. D. C., Murty, K. G., Sweeney, D. W., and Karel, C., "An Algorithm for the Traveling Salesman Problem," Operations Research Journal of America, Vol. 11, 972-989.

10. Miller, C. E., Tucker, A. W., and Zemlin, R. A., "Integer Programming Formulation of Traveling Salesman Problems," Association for Computing Machinery, Vol. 7, No. 4, 326-329 (October 1960).

11. Rustman, D. G., "Public School Transportation Practices in the State of Missouri," Education Specialist Thesis, Central Missouri State College, 1963.

12. Tillman, F. A., "Dynamic Programming Solution to the School Bus Scheduling Problem," Unpublished working papers, 1965.

APPENDIX

Discussion of Computer Program

Manually solving carrier routing problems can become an extremely tedious and laborious task, even when solving relatively small problems. The iterative nature of the computational procedure provides an ideal situation for the use of a high speed electronic computer. Therefore, a computer program was written in FORTRAN II for the IBM 1620 computer to solve carrier routing problems. The program has the capability of solving problems which include a restriction on the number of miles which can be traveled by each capacity of carrier. The program, dimensioned to solve a problem involving a maximum of 36 demand points (not including the origin) and a maximum of 10 different capacities of carriers, occupies 59,382 positions of core storage. The following discussion of the program is divided into three categories:

(1) Discussion of the output.

(2) Control card and input data cards.

(3) Operating procedure for the IBM 1620 computer.

(1) Output

Normal program output is on cards, and includes the following:

(1) A series of statements for each allocated route listing the demand points allocated to the route in their correct order of visitation, the total distance for the route, and the capacity of carrier required for the route.

(2) The total distance for all routes.

(3) The final allocation of carriers of each capacity.

If it is desired to monitor the course of the solution, additional output may be obtained. The punching of this additional output is under control of SENSE SWITCH 1, the output being punched if the switch is on, and the punching being suppressed if the switch is off. The following additional output will be punched if SENSE SWITCH 1 is on:

(1) The initial allocation of carriers.

(2) The saving half matrix elements with respective row and column identification numbers.

(3) The maximum saving at each iteration with its row and column identification numbers.

(4) If the maximum saving satisfies all conditions, the following series of output will be punched:

(a) All non-zero elements of the $t_{i,j}$ half matrix.

(b) The current Q vector.

(c) The current allocation of carriers.

(5) If the maximum saving does not satisfy all conditions (including the mileage restriction), a statement will be punched indicating whether one or more of conditions 1 through 3 were not satisfied, or whether the mileage restriction was not satisfied.

If the modified allocation procedure is used the output will also include the revised allocation of carriers immediately following the allocation normally assigned by the program. As was stated previously, the use of the revised allocation procedure is optional. Its use is controlled by SENSE SWITCH 2, the modified allocation procedure being used if the switch is on, and the normal allocation procedure being used if the switch is off.

The numbering system used by the computer program differs slightly from that used in the previous discussion of sample problem 1. This was done because the FORTRAN II programming language used does not allow the use of 0 subscripts. Therefore, the origin, which was numbered 0 in the previous discussion, is numbered 1 when using the computer program. The demand points, previously numbered starting with 1, are numbered starting with 2. Therefore, the half matrix rows are numbered from 2 to M, where M is the number of points involved including the origin, and the half matrix columns are numbered from 1 to M.

It should be noted that in certain infrequent instances the computer output will list routes to which no stops have been allocated. This is due to the program itself and is not an error in the input data, and these invalid routes should be disregarded. These routes are formed at a stage in the solution at which it was necessary to assume that a route would be formed before it was known whether or not the route actually would be formed. If, at a later stage, it was determined that the route should have been formed the output is normal. If, however, it was determined that the route should not have been formed a lengthy and difficult reordering procedure would have been necessary. Since program efficiency and core storage requirements, as well as conciseness and quality of output, were factors considered in the programming of the modified algorithm, it was decided to allow the output to contain the invalid routes rather than perform the reordering procedure. Two examples of these invalid routes occur in the output for sample problem 4, these being routes 2 and 5. Therefore, only four routes were actually formed rather than six as the output would at first seem to indicate.

(2) Control Card and Input Data Cards

The necessary control card and input data cards will now be discussed for the benefit of those wishing to use the program. The cards will be discussed in the order in which they are read into the computer and consequently must be arranged.

The first card is a control card and contains two values. The first value is the number of points involved (including the origin) and must be right-justified in columns 1 - 4. The second value is the number of different capacities of carriers available and is right-justified in columns 5 - 8. This number should include only those actual capacities available. It may be seen in the sample problems that an additional capacity is necessary in the computation. This additional capacity is set to equal to ∞ for hand computation and is internally set equal to 999999 by the computer program. The number of carriers corresponding to this capacity is internally set equal to 0, and the mileage restriction for this capacity is also internally set equal to 0.

Three sets of input data cards are required by the program. The values found on the first set of cards are the demands at each demand point. These values are punched one per card, and must be right-justified in columns 1 - 6. They must be arranged in ascending order of identification number, i.e., the demand at point 2 is punched first, followed by the demand at point 3, etc. Note that the demand points should already have been arranged according to their distance from the origin.

The second set of input data cards contains the distance half matrix elements punched one per card. Each element is accompanied by its row-column identification. The row number is right-justified in columns 1 - 4, the column number right-justified in columns 5 - 8, and the distance element right-justified in columns 9 - 12. All distances must be given in the same units, e.g., miles or tenths of miles. If the user desires to work with fractions of units, the distances must all be scaled by the appropriate factor of 10 to make them whole numbers. The distance elements must be punched row by row starting with the element in position [2,1].

The third and final set of input data cards contain three values each. The first value is the number of carriers available of a given capacity and is right-justified in columns 1 - 4. The capacity corresponding to this availability is the second value on the card and is right-justified in columns 5 - 9. The third value is the maximum number of miles which can be traveled by a carrier of the corresponding capacity. This number is right-justified in columns 10 - 13. The number of carriers of the smallest capacity should be entered as 9999 if the "dummy" capacity discussed previously is not used. If the "dummy" capacity is used, its corresponding number of carriers should be entered as 9999. This number assumes the same role as ∞ does in the hand computation.

The input data cards must be arranged in the following order to be read into the computer:

(1) Control card.

(2) Set of demand cards.

(3) Set of distance cards.

(4) Set of cards containing availabilities, capacities, and distance restrictions.

The program has been compiled and is available in object deck form including the necessary subroutines. An explanation of the procedure for solving a carrier routing problem on the IBM 1620 computer is given in the next section.

(3) Problem Solving Procedure

The following section is an explanation of the operating procedure for solving a carrier routing problem on the IBM 1620 computer.

I. Clear core storage.

A. Depress INSTANT STOP key.

B. Depress RESET key.

C. Depress INSERT key.

D. Type the instruction 160001000000R-S.

E. After approsimately four second depress INSTANT STOP key.

F. Depress RESET key.

II. Prepare card punch.

A. Pick up the cards in the punch hopper.

B. Depress the NON-PROCESS RUN OUT key on the 1622.

C. Place blank cards in the punch hopper.

D. Depress the PUNCH START button on the 1622.

III. Set SENSE SWITCHES.

A. Turn SENSE SWITCH 1 on if output is desired at each iteration; off, if iteration output is to be suppressed.

B. Turn SENSE SWITCH 2 on if the modified allocation procedure is to be used; off, if the normal allocation procedure is to be used.

C. SENSE SWITCH's 3 and 4 are not interrogated and should be turned off.

IV. Load object deck and subroutines.

A. Place object deck and subroutine deck in reader hopper.

B. Depress yellow LOAD button on 1622.

C. When the message LOAD SUBROUTINES is printed on the typewriter depress START key.

D. To read in the last two cards, depress the READER START button on the 1622.

Computer Program

```
**********************
*                    *
* SOURCE DECK LISTING *
*                    *
**********************


**    CARRIER ROUTING WITH DISTANCE RESTRICTION
*0806
      DIMENSION LB(10),LCAP(10),MILE(10),MAL(10)
      DIMENSION LQ(37),LR(37),LDIS(37),LRD(37),LQS(37),LSUM(37),LRR(37)
      DIMENSION LQQ(37)
      DIMENSION II(666),JJ(666),LD(666),LS(666),LT(666)
C
C     DATA READ IN
C
  300 READ 1,M,NN
    1 FORMAT(2I4)
C
C     M = NUMBER OF STOPS INCLUDING ORIGIN
C     NN = NUMBER OF DIFFERENT CAPACITIES OF CARRIERS
C     CALCULATE KON1 (TOTAL NUMBER OF DISTANCES INVOLVED)
C     KON1 = SUMMATION FROM I = 1 TO I = M-1 OF I
C
      MM=M-1
      KON1=0
      DO 2 I=1,MM
    2 KON1=KON1+I
C
C     KONO = MAX. NUMBER OF ITERATIONS
C
      KONO=KON1-MM
      READ 3,(LQ(I),I=2,M)
    3 FORMAT(I6)
C
C     LQ(I) = LOAD AT STOP I
C
      DO 4 I=1,KON1
    4 READ 5,II(I),JJ(I),LD(I)
    5 FORMAT(3I4)
C
C     LD(I) = MIN. DISTANCE FROM STOP II(I) TO STOP JJ(I)
C
      DO 6 I=1,NN
    6 READ 7,LB(I),LCAP(I),MILE(I)
    7 FORMAT(I4,I5,I4)
C
C     LB(I) = NUMBER OF CARRIERS OF CAPACITY LCAP(I)
C     MILE(I) = MAX. NUMBER OF MILES WHICH CAN BE TRAVELED BY A
C          CARRIER OF CAPACITY LCAP(I)
C     INITIALIZATION OF CARRIER CAPACITIES
C
```

```
      N=NN+1
      LB(N)=0
      LCAP(N)=999999
      MILE(N)=0
C
C     ASSIGN INITIAL ALLOCATION
C
      IF(SENSE SWITCH 2)801,800
C
C     REVISED INITIAL ALLOCATION PROCEDURE
C
  801 DO 802 I=2,M
  802 LQQ(I)=LQ(I)
      DO 817 I=1,N
  817 MAL(I)=0
      J=2
  816 I=2
      ISAV=I
      LOW=LQQ(I)
  805 I=I+1
      IF(I-M)803,803,806
  803 IF(LQQ(I)-LOW)804,805,805
  804 LOW=LQQ(I)
      ISAV=I
      GO TO 805
  806 K=1
  808 IF(LOW-LCAP(K))809,809,807
  807 K=K+1
      IF(K-NN)808,808,811
  809 MAL(K)=MAL(K)+1
      IF(MAL(K)-LB(K))815,815,810
  810 MAL(K)=MAL(K)-1
      GO TO 807
  811 PRINT 812
  812 FORMAT(70HTHERE ARE NOT ENOUGH AVAILABLE CARRIERS TO ALLOCATE ONE
     1TO EACH DEMAND)
      GO TO 814
  815 LQQ(ISAV)=999999
      J=J+1
      IF(J-M)816,816,12
  800 DO 700 I=1,N
  700 MAL(I)=0
      DO 704 I=2,M
      J=1
  701 IF(LQ*I)-LCAP(J))703,703,702
  702 J=J+1
      GO TO 701
  703 MAL(J)=MAL(J)+1
  704 CONTINUE
C
C     INITIALIZATION
C
```

```
   12 DO 16 I=2,M
      LRD(I)=0
      LDIS(I)=0
      LQS(I)=LQ(I)
      LRR(I)=0
   16 LR(I)=0
C
C     LR(I) = VECTOR TO SAVE ALLOCATED ROUTES
C     LDIS(I) = INTERMEDIATE DISTANCE FOR ROUTE I
C     LRD(I) = TOTAL DISTANCE FOR ROUTE I
C     LQS(I) SAVES Q VECTOR IN CASE DISTANCE REQUIREMENT IS
C          NOT SATISFIED
C     LRR(I) SAVES CHANGED LR(I) IN CASE DISTANCE REQUIREMENT
C          IS NOT SATISFIED
C
      KON2=0
C
C     KON2 = ITERATION NUMBER
C
      LTD=0
      KON10=0
C
C     KON10 SAVES NUMBER OF ROUTES WHICH HAVE BEEN ALLOCATED
C
      KON6=0
C
C     KON6 = ROUTE BEING ALLOCATED
C     CALCULATION OF LSUM(I) TO SAVE I CORRESPONDING TO (I,1)
C
      DO 315 I=2,M
  315 LSUM(I)=0
      I=3
  317 LA=I-2
      DO 316 J=1,LA
  316 LSUM(I)=LSUM(I)+J
      I=I+1
      IF(I-M)317,317,318
C
C     CALCULATION OF SAVINGS
C
  318 LS(1)=+99999
      I=3
   25 J=1
      K=I-1
   22 LA=LSUM(I)+J
      IF(J-1)20,20,21
   20 LS(LA)=-99999
   23 J=J+1
      IF(J-K)22,22,24
   21 LF=LSUM(I)+1
      LSAV1=LD(LF)
      LF=LSUM(J)+1
```

```
      LSAV2=LD(LF)
      LF=LSUM(I)+J
      LSAV3=LD(LF)
      LS(LA)=LSAV1+LSAV2-LSAV3
      GO TO 23
   24 I=I+1
      IF(I-M)25,25,37
C
C     PUNCHOUT OF INITIAL ALLOCATION AND SAVING MATRIX
C
   37 IF(SENSE SWITCH 1)32,40
   32 PUNCH 13
   13 FORMAT(18HINITIAL ALLOCATION/)
      DO 14 I=1,N
   14 PUNCH 15,LCAP(I),MAL(I)
   15 FORMAT(10HCAPACITY =I7,2X,18HNUMBER ALLOCATED =I4)
      PUNCH 33
   33 FORMAT(1H )
      PUNCH 33
      PUNCH 11
   11 FORMAT(14HSAVINGS MATRIX/)
      PUNCH 34
   34 FORMAT(3HROW,3X,3HCOL,3X,6HSAVING/)
      DO 35 I=1,KON1
   35 PUNCH 36,II(I),JJ(I),LS(I)
   36 FORMAT(I3,3X,I3,3X,I6)
C
C     ASSIGN INITIAL BASIC SOLUTION
C
   40 I=1
   43 IF(JJ(I)-1)41,42,41
   41 LT(I)=0
   44 I=I+1
      IF(I-KON1)43,43,135
   42 LT(I)=2
      GO TO 44
C
C     SEARCH OF SAVING VECTOR FOR MAXIMUM SAVING
C
  135 I=1
      MSAV=LS(I)
      IK=II(I)
      JK=JJ(I)
   52 I=I+1
      IF(I-KON1)50,50,290
   50 IF(MSAV-LS(I))51,52,52
   51 MSAV=LS(I)
      IK=II(I)
      JK=JJ(I)
      GO TO 52
  290 IF(MSAV)200,53,53
   53 KON2=KON2+1
```

```
      LHMC=0
C
C     PUNCHOUT OF MAX. SAVING
C
   55 IF(SENSE SWITCH 1)56,60
   56 PUNCH 33
      PUNCH 33
      PUNCH 57,IK,JK,MSAV
   57 FORMAT(3HI =I4,2X,3HJ =I4,2X,13HMAX. SAVING =I6/)
C
C     CHECK TO SEE IF MAX. SAVING MEETS PRESCRIBED CONDITIONS
C     IDI IS THE I CORRESPONDING TO (IK,1)
C
   60 IDI=LSUM(IK)+1
C
C     CONDITION 1
C
   64 IF(LT(IDI))301,301,65
  301 LMN=1
      GO TO 136
C
C     IDJ IS THE I CORRESPONDING TO (JK,1)
C
   65 IDJ=LSUM(JK)+1
C
C     CONDITION 1
C
   69 IF(LT(IDJ))301,301,70
C
C     IDB IS THE I CORRESPONDING TO (IK,JK)
C
   70 IDB=LSUM(IK)+JK
C
C     CONDITION 2
C
   74 IF(LR(IK))75,75,250
  250 IF(LR(IK)-LR(JK))75,302,75
  302 LMN=1
      GO TO 130
   75 I=1
   76 IF(LQ(IK)-LCAP(I))78,78,77
   77 I=I+1
      GO TO 76
C
C     KON3 SAVES THE NUMBER OF THE CAPACITY REQUIRED BY LQ(IK)
C
   78 KON3=I
      MAL(KON3)=MAL(KON3)-1
      I=1
   81 IF(LQ(JK)-LCAP(I))80,80,79
   79 I=I+1
      GO TO 81
```

```
C
C     KON4 SAVES THE NUMBER OF THE CAPACITY REQUIRED BY LQ(JK)
C
   80 KON4=I
      MAL(KON4)=MAL(KON4)-1
      LZ=LQ(IK)+LQ(JK)
      I=1
   83 IF(LZ-LCAP(I))82,82,400
  400 I=I+1
      GO TO 83
C
C     KON5 SAVES THE NUMBER OF THE CAPACITY REQUIRED BY LZ
C
   82 KON5=I
      MAL(KON5)=MAL(KON5)+1
C
C     CONDITION 3
C
      IF(MAL(KON5)-LB(KON5))84,84,303
  303 LMN=1
      GO TO 150
C
C     ASSUME MILEAGE RESTRICTION IS MET
C     CALCULATE NEW VALUES OF LT(I)
C
   84 LT(IDB)=1
      LT(IDI)=LT(IDI)-1
      LT(IDJ)=LT(IDJ)-1
C
C     SET LQ(I) = 0 FOR LT(I) = 0
C
      IF(LT(IDI))86,85,86
   85 LQ(IK)=0
   86 IF(LT(IDJ))88,87,88
   87 LQ(JK)=0
   88 IF(LT(IDI))89,90,89
   89 LQ(IK)=LZ
   90 IF(LT(IDJ))91,92,91
   91 LQ(JK)=LZ
C
C     SET LQ(I) = TOTAL LOAD ON THE ROUTE FOR ALL OTHER STOPS
C          ALLOCATED ON ROUTE
C
   92 DO 120 I=1,M
  120 LDIS(I)=0
C
C     CHECK TO SEE IF STOPS IK AND/OR JK ARE ALREADY ALLOCATED
C          ON A ROUTE
C
      IF(LR(IK))108,102,108
  102 IF(LR(JK))109,103,109
```

```
C
C     NEW ROUTE FORMED
C
  103 KON10=KON10+1
      KON6=KON10
      LR(IK)=KON6
      LR(JK)=KON6
      LDIS(KON6)=LD(IDI)+LD(IDJ)+LD(IDB)
      IF(LDIS(KON6)-MILE(KON5))104,104,155
C
C     DISTANCE REQUIREMENT NOT EXCEEDED
C
  104 LRD(KON6)=LDIS(KON6)
      DO 311 I=2,M
  311 LQS(I)=LQ(I)
C
C     PUNCHOUT ALL NON-ZERO VALUES OF LT(I) AND ALL VALUES
C          OF LQ(I)
C
      IF(SENSE SWITCH 1)93,101
   93 I=1
   96 IF(LT(I))94,97,94
   94 PUNCH 95,II(I),JJ(I),LT(I)
   95 FORMAT(2HT(I3,1H,I3,3H) =I2)
   97 I=I+1
      IF(I-KON1)96,96,98
   98 PUNCH 33
      DO 99 I=2,M
   99 PUNCH 100,I,LQ(I)
  100 FORMAT(3HI =I4,2X,3HQ =I6)
      PUNCH 33
  101 LMN=0
C
C     PUNCHOUT NEW ALLOCATION
C
      IF(SENSE SWITCH 1)131,134
  131 DO 132 I=1,N
  132 PUNCH 15,LCAP(I),MAL(I)
  134 IF(SENSE SWITCH 2)821,130
C
C     REVISED ALLOCATION PROCEDURE
C
  821 DO 818 I=2,M
  818 LQQ(I)=LQ(I)
      DO 819 I=1,N
  819 MAL(I)=0
  825 I=2
      ISAV=I
      LOW=LQQ(I)
  822 I=I+1
      IF(I-M)823,823,826
  823 IF(LQQ(I)-LOW)824,822,822
```

```
  824 LOW=LQQ(I)
      ISAV=I
      GO TO 822
  826 IF(LOW-999999)839,841,841
  841 IF(SENSE SWITCH 1)842,135
  842 PUNCH 33
      DO 843 I=1,N
  843 PUNCH 844,LCAP(I),MAL(I)
  844 FORMAT(10HCAPACITY =I7,2X,20HREVISED ALLOCATION =I4)
      GO TO 135
  839 IF(LOW)827,827,828
  827 LQQ(ISAV)=999999
      GO TO 825
  828 K=ISAV+1
  830 IF(LOW-LQQ(K))829,840,829
  829 K=K+1
      IF(K-M)830,830,831
  831 KK=1
  834 IF(LOW-LCAP(KK))832,832,835
  832 MAL(KK)=MAL(KK)+1
      IF(MAL(KK)-LB(KK))827,827,833
  833 MAL(KK)=MAL(KK)-1
  835 KK=KK+1
      IF(KK-NN)834,834,811
  840 IF(LR(ISAV))829,829,837
  837 IF(LR(ISAV)-LR(K))829,838,829
  838 LQQ(K)=999999
      GO TO 831
C
C     JOIN NEW STOPS TO OLD_ROUTE
C
  108 IF(LR(JK))67,66,67
   66 LRSI=LR(IK)
      LRSJ=0
      LR(JK)=LR(IK)
      GO TO 110
   67 IF(LR(IK)-LR(JK))68,68,71
   68 NA=LR(JK)
      NC=LR(IK)
      GO TO 72
   71 NA=LR(IK)
      NC=LR(JK)
   72 I=2
      NSAV=LRD(NA)
      LRD(NA)=0
      LHMC=1
  114 IF(LR(I)-NA)73,115,73
   73 I=I+1
      IF(I-M)114,114,110
  115 LR(I)=NC
      LRR(I)=1
      GO TO 73
```

```
  109 LRSI=*
      LRSJ=LR(JK)
      LR(IK)=LR(JK)
  110 KON6=LR(IK)
      I=2
  112 IF(LR(I)-KON6)111,113,111
  111 I=I+1
      IF(I-M)112,112,121
  113 K=LSUM(I)+1
  117 IF(LT(K)-1)119,118,118
  118 LDIS(KON6)=LDIS(KON6)+LD(K)
      IF(JJ(K)-1)119,270,119
  270 LQ(I)=LZ
  119 K=K+1
      IF(II(K)-I)111,117,111
C
C     CHECK TO SEE IF DISTANCE REQUIREMENT IS MET
C
  121 IF(LDIS(KON6)-MILE(KON5))104,104,510
  510 IF(LHMC)511,160,511
  511 LRD(NA)=NSAV
      I=2
  513 IF(LRR(I)-1)512,514,512
  512 I=I+1
      IF(I-M)513,513,165
  514 LR(I)=NA
      LRR(I)=0
C
C     FORCE MAX. SAVING OUT OF CONSIDERATION
C
  136 IDB=LSUM(IK)+JK
  130 LS(IDB)=-99999
      IF(SENSE SWITCH 1)304,282
  304 IF(LMN-1)282,281,305
  281 PUNCH 280
  280 FORMAT(43HMAX. SAVING DOES NOT SATISFY ONE OR MORE OF,23H CONDITI
     1NS 1 THROUGH 3/)
      GO TO 282
  305 PUNCH 306
  306 FORMAT(49HMAX. SAVING DOES NOT SATISFY DISTANCE REQUIREMENT/)
  282 IF(KON0-KON2)200,200,135
C
C     REINITIALIZATION IF MAX. SAVING DOES NOT SATISFY CONDITION 3
C
  150 MAL(KON3)=MAL(KON3)+1
      MAL(KON4)=MAL(KON4)+1
      MAL(KON5)=MAL(KON5)-1
      GO TO 130
C
C     REINITIALIZATION IF MILEAGE REQUIREMENT IS NOT MET
C
  155 KON1-=KON10-1
```

```
      LR(IK)=0
      LR(JK)=0
      GO TO 165
  160 LR(IK)=LRSI
      LR(JK)=LRSJ
  165 LMN=2
      MAL(KON3)=MAL(KON3)+1
      MAL(KON4)=MAL(KON4)+1
      MAL(KON5)=MAL(KON5)-1
      LT(IDB)=0
      LT(IDI)=LT(IDI)+1
      LT(IDJ)=LT(IDJ)+1
      DO 310 I=2,M
  310 LQ(I)=LQS(I)
      GO TO 130
C
C     CALCULATION OF TOTAL DISTANCE AND FINAL PUNCHOUT
C
  200 PUNCH 33
      PUNCH 33
C
C     PUT CUSTOMERS SERVED EXCLUSIVELY BY A TRUCK ON NEW ROUTES
C
      I=2
  201 LA=LSUM(I)+1
      IF(LT(LA)-1)202,202,203
  202 I=I+1
      IF(I-M)201,201,204
  203 KON10=KON10+1
      LRD(KON10)=LD(LA)+LD(LA)
      LR(I)=KON10
      GO TO 202
  204 LA=1
      DO 617 I=2,M
  617 LRR(I)=0
  209 I=2
      PUNCH 241
  241 FORMAT(11X,16H****************)
      PUNCH 205,LA
  205 FORMAT(6HROUTE I2,4X,4HFROM,7X,2HTO)
      PUNCH 241
      PUNCH 33
  601 IF(LR(I)-LA)600,603,600
  600 I=I+1
      IF(I-M)601,601,602
  602 LA=LA+1
      IF(LA-KON10)209,209,618
  603 LF=LSUM(I)+1
      IF(LT(LF)-1)600,604,615
  604 PUNCH 213,I
  213 FORMAT(11X,6HORIGIN,6X,I2)
      LSAV1=I
```

```
    LRR(I)=1
614 I=1
605 I=I+1
    IF(LR(I)-LA)605,612,605
612 IF(LRR(I))606,606,605
606 IF(I-LSAV1)613,613,607
607 LF=LSUM(I)+LSAV1
608 IF(LT(LF)-1)605,609,605
613 LF=LSUM(LSAV1)+I
    GO TO 608
609 PUNCH 610,LSAV1,I
610 FORMAT(13X,I2,8X,I2)
    LSAV1=I
    LRR(LSAV1)=1
    LF=LSUM(LSAV1)+1
    IF(LT(LF)-1)614,611,614
611 PUNCH 215,LSAV1
215 FORMAT(13X,I2,6X,6HORIGIN/)
616 PUNCH 219,LRD(LA)
219 FORMAT(3X,22HDISTANCE FOR ROUTE IS I5,6H MILES/)
    K=1
628 IF(LQ(1)-LCAP(K))625,625,627
625 PUNCH 626,LCAP(K)
626 FORMAT(3X,46HROUTE REQUIRES A CARRIER HAVING A CAPACITY OF I5,6H U
   1NITS//)
    GO TO 602
627 K=K+1
    GO TO 628
615 PUNCH 213,I
    PUNCH 215,I
    GO TO 616
618 LTD=0
    DO 619 I=1,KON10
619 LTD=LTD+LRD(I)
    PUNCH 620,LTD
620 FORMAT(32HTOTAL DISTANCE FOR ALL ROUTES IS I6,6H MILES///)
    PUNCH 621
621 FORMAT(16HFINAL ALLOCATION/)
    DO 622 I=1,NN
622 PUNCH 15,LCAP(I),MAL(I)
814 PRINT 240
240 FORMAT(38HTO READ ANOTHER SET OF DATA PUSH START)
    PAUSE
    GO TO 300
    END
```

Sample Problems

```
**********************************
*                                *
* INPUT DATA FOR SAMPLE PROBLEM 1 *
*                                *
**********************************

 130003
 1200
 1700
 1500
 1400
 1700
 1400
 1200
 1900
 1800
 1600
 1700
 1100
  2   1   9
  3   1  14
  3   2   5
  4   1  21
  4   2  12
  4   3   7
  5   1  23
  5   2  22
  5   3  17
  5   4  10
  6   1  22
  6   2  21
  6   3  16
  6   4  21
  6   5  19
  7   1  25
  7   2  24
  7   3  23
  7   4  30
  7   5  28
  7   6   9
  8   1  32
  8   2  31
  8   3  26
  8   4  27
  8   5  25
  8   6  10
  8   7   7
  9   1  36
  9   2  35
  9   3  30
  9   4  37
```

```
 9  5 35
 9  6 16
 9  7 11
 9  8 10
10  1 38
10  2 37
10  3 36
10  4 43
10  5 41
10  6 22
10  7 13
10  8 16
10  9  6
11  1 42
11  2 41
11  3 36
11  4 31
11  5 29
11  6 20
11  7 17
11  8 10
11  9  6
11 10 12
12  1 50
12  2 49
12  3 44
12  4 37
12  5 31
12  6 28
12  7 25
12  8 18
12  9 14
12 10 12
12 11  8
13  1 52
13  2 51
13  3 46
13  4 39
13  5 29
13  6 30
13  7 27
13  8 20
13  9 16
13 10 20
13 11 10
13 12 10
9999040009999
  3050009999
  4060009999
```

```
********************************
*                              *
* SOLUTION FOR SAMPLE PROBLEM 1 *
*                              *
********************************
```

INITIAL ALLOCATION

```
CAPACITY =   4000  NUMBER ALLOCATED =  12
CAPACITY =   5000  NUMBER ALLOCATED =   0
CAPACITY =   6000  NUMBER ALLOCATED =   0
CAPACITY = 999999  NUMBER ALLOCATED =   0
```

SAVINGS MATRIX

ROW	COL	SAVING
2	1	-99999
3	1	-99999
3	2	18
4	1	-99999
4	2	18
4	3	28
5	1	-99999
5	2	10
5	3	20
5	4	34
6	1	-99999
6	2	10
6	3	20
6	4	22
6	5	26
7	1	-99999
7	2	10
7	3	16
7	4	16
7	5	20
7	6	38
8	1	-99999
8	2	10
8	3	20
8	4	26
8	5	30
8	6	44
8	7	50
9	1	-99999
9	2	10
9	3	20
9	4	20
9	5	24

9	6	42
9	7	50
9	8	58
10	1	-99999
10	2	10
10	3	16
10	4	16
10	5	20
10	6	38
10	7	50
10	8	54
10	9	68
11	1	-99999
11	2	10
11	3	20
11	4	32
11	5	36
11	6	44
11	7	50
11	8	64
11	9	72
11	10	68
12	1	-99999
12	2	10
12	3	20
12	4	34
12	5	42
12	6	44
12	7	50
12	8	64
12	9	72
12	10	76
12	11	84
13	1	-99999
13	2	10
13	3	20
13	4	34
13	5	46
13	6	44
13	7	50
13	8	64
13	9	72
13	10	70
13	11	84
13	12	92

```
I =  13  J =  12  MAX. SAVING =    92

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 2
T(  5,  1) = 2
T(  6,  1) = 2
T(  7,  1) = 2
T(  8,  1) = 2
T(  9,  1) = 2
T( 10,  1) = 2
T( 11,  1) = 2
T( 12,  1) = 1
T( 13,  1) = 1
T( 13, 12) = 1

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  1400
I =   8  Q =  1200
I =   9  Q =  1900
I =  10  Q =  1800
I =  11  Q =  1600
I =  12  Q =  2800
I =  13  Q =  2800

CAPACITY =   4000  NUMBER ALLOCATED =  11
CAPACITY =   5000  NUMBER ALLOCATED =   0
CAPACITY =   6000  NUMBER ALLOCATED =   0
CAPACITY = 999999  NUMBER ALLOCATED =   0


I =  12  J =  11  MAX. SAVING =    84

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 2
T(  5,  1) = 2
T(  6,  1) = 2
T(  7,  1) = 2
T(  8,  1) = 2
T(  9,  1) = 2
T( 10,  1) = 2
T( 11,  1) = 1
T( 12, 11) = 1
T( 13,  1) = 1
T( 13, 12) = 1
```

```
I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  1400
I =   8  Q =  1200
I =   9  Q =  1900
I =  10  Q =  1800
I =  11  Q =  4400
I =  12  Q =     0
I =  13  Q =  4400

CAPACITY =   4000  NUMBER ALLOCATED =   9
CAPACITY =   5000  NUMBER ALLOCATED =   1
CAPACITY =   6000  NUMBER ALLOCATED =   0
CAPACITY = 999999  NUMBER ALLOCATED =   0

I =  13  J =  11  MAX. SAVING =     84

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  12  J =  10  MAX. SAVING =     76

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  11  J =   9  MAX. SAVING =     72

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  12  J =   9  MAX. SAVING =     72

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  13  J =   9  MAX. SAVING =     72

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  13  J =  10  MAX. SAVING =     70

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  10  J =   9  MAX. SAVING =    68

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 2
T(  5,  1) = 2
T(  6,  1) = 2
T(  7,  1) = 2
T(  8,  1) = 2
T(  9,  1) = 1
T( 10,  1) = 1
T( 10,  9) = 1
T( 11,  1) = 1
T( 12, 11) = 1
T( 13,  1) = 1
T( 13, 12) = 1

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  1400
I =   8  Q =  1200
I =   9  Q =  3700
I =  10  Q =  3700
I =  11  Q =  4400
I =  12  Q =     0
I =  13  Q =  4400

CAPACITY =   4000  NUMBER ALLOCATED =   8
CAPACITY =   5000  NUMBER ALLOCATED =   1
CAPACITY =   6000  NUMBER ALLOCATED =   0
CAPACITY = 999999  NUMBER ALLOCATED =   0


I =  11  J =  10  MAX. SAVING =    68

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  11  J =   8  MAX. SAVING =     64

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 2
T(  5,  1) = 2
T(  6,  1) = 2
T(  7,  1) = 2
T(  8,  1) = 1
T(  9,  1) = 1
T( 10,  1) = 1
T( 10,  9) = 1
T( 11,  8) = 1
T( 12, 11) = 1
T( 13,  1) = 1
T( 13, 12) = 1

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  1400
I =   8  Q =  5600
I =   9  Q =  3700
I =  10  Q =  3700
I =  11  Q =     0
I =  12  Q =     0
I =  13  Q =  5600

CAPACITY =   4000  NUMBER ALLOCATED =    7
CAPACITY =   5000  NUMBER ALLOCATED =    0
CAPACITY =   6000  NUMBER ALLOCATED =    1
CAPACITY = 999999  NUMBER ALLOCATED =    0


I =  12  J =   8  MAX. SAVING =     64

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  13  J =   8  MAX. SAVING =     64

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   9  J =   8  MAX. SAVING =     58

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  10  J =   8  MAX. SAVING =    54

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   8  J =   7  MAX. SAVING =    50

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   9  J =   7  MAX. SAVING =    50

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 2
T(  5,  1) = 2
T(  6,  1) = 2
T(  7,  1) = 1
T(  8,  1) = 1
T(  9,  7) = 1
T( 10,  1) = 1
T( 10,  9) = 1
T( 11,  8) = 1
T( 12, 11) = 1
T( 13,  1) = 1
T( 13, 12) = 1

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  5100
I =   8  Q =  5600
I =   9  Q =     0
I =  10  Q =  5100
I =  11  Q =     0
I =  12  Q =     0
I =  13  Q =  5600

CAPACITY =    4000  NUMBER ALLOCATED =    5
CAPACITY =    5000  NUMBER ALLOCATED =    0
CAPACITY =    6000  NUMBER ALLOCATED =    2
CAPACITY =  999999  NUMBER ALLOCATED =    0


I =  10  J =   7  MAX. SAVING =    50

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  11  J =   7  MAX. SAVING =    50
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  12  J =   7  MAX. SAVING =    50
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  13  J =   7  MAX. SAVING =    50
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  13  J =   5  MAX. SAVING =    46
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   8  J =   6  MAX. SAVING =    44
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  11  J =   6  MAX. SAVING =    44
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  12  J =   6  MAX. SAVING =    44
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  13  J =   6  MAX. SAVING =    44
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   9  J =   6  MAX. SAVING =    42
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  12  J =   5  MAX. SAVING =     42
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   7  J =   6  MAX. SAVING =     38
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  10  J =   6  MAX. SAVING =     38
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  11  J =   5  MAX. SAVING =     36
MAY. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   5  J =   4  MAX. SAVING =     34

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 1
T(  5,  1) = 1
T(  5,  4) = 1
T(  6,  1) = 2
T(  7,  1) = 1
T(  8,  1) = 1
T(  9,  7) = 1
T( 10,  1) = 1
T( 10,  9) = 1
T( 11,  8) = 1
T( 12, 11) = 1
T( 13,  1) = 1
T( 13, 12) = 1

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  2900
I =   5  Q =  2900
I =   6  Q =  1700
I =   7  Q =  5100
I =   8  Q =  5600
I =   9  Q =     0
I =  10  Q =  5100
I =  11  Q =     0
I =  12  Q =     0
I =  13  Q =  5600
```

```
CAPACITY =   4000  NUMBER ALLOCATED =   4
CAPACITY =   5000  NUMBER ALLOCATED =   0
CAPACITY =   6000  NUMBER ALLOCATED =   2
CAPACITY = 999999  NUMBER ALLOCATED =   0

I =  12  J =   4  MAX. SAVING =     34

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  13  J =   4  MAX. SAVING =     34

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  11  J =   4  MAX. SAVING =     32

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   8  J =   5  MAX. SAVING =     30

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   4  J =   3  MAX. SAVING =     28

T(  2,   1) = 2
T(  3,   1) = 1
T(  4,   3) = 1
T(  5,   1) = 1
T(  5,   4) = 1
T(  6,   1) = 2
T(  7,   1) = 1
T(  8,   1) = 1
T(  9,   7) = 1
T( 10,   1) = 1
T( 10,   9) = 1
T( 11,   8) = 1
T( 12,  11) = 1
T( 13,   1) = 1
T( 13,  12) = 1
```

```
I =   2  Q =  1200
I =   3  Q =  4600
I =   4  Q =     0
I =   5  Q =  4600
I =   6  Q =  1700
I =   7  Q =  5100
I =   8  Q =  5600
I =   9  Q =     0
I =  10  Q =  5100
I =  11  Q =     0
I =  12  Q =     0
I =  13  Q =  5600

CAPACITY =   4000  NUMBER ALLOCATED =   2
CAPACITY =   5000  NUMBER ALLOCATED =   1
CAPACITY =   6000  NUMBER ALLOCATED =   2
CAPACITY = 999999  NUMBER ALLOCATED =   0

I =   6  J =   5  MAX. SAVING =    26

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   8  J =   4  MAX. SAVING =    26

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   9  J =   5  MAX. SAVING =    24

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   6  J =   4  MAX. SAVING =    22

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   5  J =   3  MAX. SAVING =    20

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   6  J =   3  MAX. SAVING =    20

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  7  J =  5  MAX. SAVING =    20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  8  J =  3  MAX. SAVING =    20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  9  J =  3  MAX. SAVING =    20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  9  J =  4  MAX. SAVING =    20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 10  J =  5  MAX. SAVING =    20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 11  J =  3  MAX. SAVING =    20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 12  J =  3  MAX. SAVING =    20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 13  J =  3  MAX. SAVING =    20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =    3  J =    2  MAX. SAVING =     18

T(   2,   1) = 1
T(   3,   2) = 1
T(   4,   3) = 1
T(   5,   1) = 1
T(   5,   4) = 1
T(   6,   1) = 2
T(   7,   1) = 1
T(   8,   1) = 1
T(   9,   7) = 1
T(  10,   1) = 1
T(  10,   9) = 1
T(  11,   8) = 1
T(  12,  11) = 1
T(  13,   1) = 1
T(  13,  12) = 1

I =    2  Q =  5800
I =    3  Q =     0
I =    4  Q =     0
I =    5  Q =  5800
I =    6  Q =  1700
I =    7  Q =  5100
I =    8  Q =  5600
I =    9  Q =     0
I =   10  Q =  5100
I =   11  Q =     0
I =   12  Q       0
I =   13  Q =  5600

CAPACITY =    4000  NUMBER ALLOCATED =    1
CAPACITY =    5000  NUMBER ALLOCATED =    0
CAPACITY =    6000  NUMBER ALLOCATED =    3
CAPACITY = 999999  NUMBER ALLOCATED =    0

I =    4  J =    2  MAX. SAVING =     18

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =    7  J =    3  MAX. SAVING =     16

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =    7  J =    4  MAX. SAVING =     16

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  10  J =   3  MAX. SAVING =     16
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  10  J =   4  MAX. SAVING =     16
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   5  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   6  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   7  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   8  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   9  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  10  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  11  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  12  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  13  J =   2  MAX. SAVING =     10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


          ****************
ROUTE  1    FROM        TO
          ****************

          ORIGIN       8
             8        11
            11        12
            12        13
            13      ORIGIN

   DISTANCE FOR ROUTE IS   112 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF  6000 UNITS


          ****************
ROUTE  2    FROM        TO
          ****************

          ORIGIN       7
             7         9
             9        10
            10      ORIGIN

   DISTANCE FOR ROUTE IS    80 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF  6000 UNITS
```

```
        ***************
ROUTE  3   FROM       TO
        ***************

        ORIGIN        2
           2          3
           3          4
           4          5
           5        ORIGIN

   DISTANCE FOR ROUTE IS    54 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF  6000 UNITS


        ***************
ROUTE  4   FROM       TO
        ***************

        ORIGIN        6
           6        ORIGIN

   DISTANCE FOR ROUTE IS    44 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF  4000 UNITS


TOTAL DISTANCE FOR ALL ROUTES IS   290 MILES


FINAL ALLOCATION

CAPACITY =   4000  NUMBER ALLOCATED =    1
CAPACITY =   5000  NUMBER ALLOCATED =    0
CAPACITY =   6000  NUMBER ALLOCATED =    3
```

```
**********************************
*                                *
* INPUT DATA FOR SAMPLE PROBLEM 2 *
*                                *
**********************************


 130004
 1200
 1700
 1500
 1400
 1700
 1400
 1200
 1900
 1800
 1600
 1700
 1100
  2   1   9
  3   1  14
  3   2   5
  4   1  21
  4   2  12
  4   3   7
  5   1  23
  5   2  22
  5   3  17
  5   4  10
  6   1  22
  6   2  21
  6   3  16
  6   4  21
  6   5  19
  7   1  25
  7   2  24
  7   3  23
  7   4  30
  7   5  28
  7   6   9
  8   1  32
  8   2  31
  8   3  26
  8   4  27
  8   5  25
  8   6  10
  8   7   7
  9   1  36
  9   2  35
  9   3  30
  9   4  37
```

```
   9    5  35
   9    6  16
   9    7  11
   9    8  10
  10    1  38
  10    2  37
  10    3  36
  10    4  43
  10    5  41
  10    6  22
  10    7  13
  10    8  16
  10    9   6
  11    1  42
  11    2  41
  11    3  36
  11    4  31
  11    5  29
  11    6  20
  11    7  17
  11    8  10
  11    9   6
  11   10  12
  12    1  50
  12    2  49
  12    3  44
  12    4  37
  12    5  31
  12    6  28
  12    7  25
  12    8  18
  12    9  14
  12   10  12
  12   11   8
  13    1  52
  13    2  51
  13    3  46
  13    4  39
  13    5  29
  13    6  30
  13    7  27
  13    8  20
  13    9  16
  13   10  20
  13   11  10
  13   12  10
9999019000104
   2040000104
   3050000104
   4060000104
```

```
********************************
*                              *
* SOLUTION FOR SAMPLE PROBLEM 2 *
*                              *
********************************
```

INITIAL ALLOCATION

```
CAPACITY =   1900  NUMBER ALLOCATED =  12
CAPACITY =   4000  NUMBER ALLOCATED =   0
CAPACITY =   5000  NUMBER ALLOCATED =   0
CAPACITY =   6000  NUMBER ALLOCATED =   0
CAPACITY = 999999  NUMBER ALLOCATED =   0
```

SAVINGS MATRIX

ROW	COL	SAVING
2	1	-99999
3	1	-99999
3	2	18
4	1	-99999
4	2	18
4	3	28
5	1	-99999
5	2	10
5	3	20
5	4	34
6	1	-99999
6	2	10
6	3	20
6	4	22
6	5	26
7	1	-99999
7	2	10
7	3	16
7	4	16
7	5	20
7	6	38
8	1	-99999
8	2	10
8	3	20
8	4	26
8	5	30
8	6	44
8	7	50
9	1	-99999
9	2	10
9	3	20
9	4	20

9	5	24
9	6	42
9	7	50
9	8	58
10	1	-99999
10	2	10
10	3	16
10	4	16
10	5	20
10	6	38
10	7	50
10	8	54
10	9	68
11	1	-99999
11	2	10
11	3	20
11	4	32
11	5	36
11	6	44
11	7	50
11	8	64
11	9	72
11	10	68
12	1	-99999
12	2	10
12	3	20
12	4	34
12	5	42
12	6	44
12	7	50
12	8	64
12	9	72
12	10	76
12	11	84
13	1	-99999
13	2	10
13	3	20
13	4	34
13	5	46
13	6	44
13	7	50
13	8	64
13	9	72
13	10	70
13	11	84
13	12	92

I = 13 J = 12 MAX. SAVING = 92

MAX. SAVING DOES NOT SATISFY DISTANCE REQUIREMENT

```
I =  12  J =  11  MAX. SAVING =     84

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 2
T(  5,  1) = 2
T(  6,  1) = 2
T(  7,  1) = 2
T(  8,  1) = 2
T(  9,  1) = 2
T( 10,  1) = 2
T( 11,  1) = 1
T( 12,  1) = 1
T( 12, 11) = 1
T( 13,  1) = 2

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  1400
I =   8  Q =  1200
I =   9  Q =  1900
I =  10  Q =  1800
I =  11  Q =  3300
I =  12  Q =  3300
I =  13  Q =  1100

CAPACITY =   1900  NUMBER ALLOCATED =  10
CAPACITY =   4000  NUMBER ALLOCATED =   1
CAPACITY =   5000  NUMBER ALLOCATED =   0
CAPACITY =   6000  NUMBER ALLOCATED =   0
CAPACITY = 999999  NUMBER ALLOCATED =   0

I =  13  J =  11  MAX. SAVING =     84

MAX. SAVING DOES NOT SATISFY DISTANCE REQUIREMENT
```

```
I =  12  J =  10  MAX. SAVING =     76

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 2
T(  5,  1) = 2
T(  6,  1) = 2
T(  7,  1) = 2
T(  8,  1) = 2
T(  9,  1) = 2
T( 10,  1) = 1
T( 11,  1) = 1
T( 12, 10) = 1
T( 12, 11) = 1
T( 13,  1) = 2

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  1400
I =   8  Q =  1200
I =   9  Q =  1900
I =  10  Q =  5100
I =  11  Q =  5100
I =  12  Q =     0
I =  13  Q =  1100

CAPACITY =   1900  NUMBER ALLOCATED =   9
CAPACITY =   4000  NUMBER ALLOCATED =   0
CAPACITY =   5000  NUMBER ALLOCATED =   0
CAPACITY =   6000  NUMBER ALLOCATED =   1
CAPACITY = 999999  NUMBER ALLOCATED =   0

I =  11  J =   9  MAX. SAVING =     72

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  12  J =   9  MAX. SAVING =     72

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  13  J =   9  MAX. SAVING =     72

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 2
T(  5,  1) = 2
T(  6,  1) = 2
T(  7,  1) = 2
T(  8,  1) = 2
T(  9,  1) = 1
T( 10,  1) = 1
T( 11,  1) = 1
T( 12, 10) = 1
T( 12, 11) = 1
T( 13,  1) = 1
T( 13,  9) = 1

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  1400
I =   8  Q =  1200
I =   9  Q =  3000
I =  10  Q =  5100
I =  11  Q =  5100
I =  12  Q =     0
I =  13  Q =  3000

CAPACITY =    1900  NUMBER ALLOCATED =    7
CAPACITY =    4000  NUMBER ALLOCATED =    1
CAPACITY =    5000  NUMBER ALLOCATED =    0
CAPACITY =    6000  NUMBER ALLOCATED =    1
CAPACITY = 999999  NUMBER ALLOCATED =    0

I =  13  J =  10  MAX. SAVING =     70

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  10  J =   9  MAX. SAVING =     68

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  11  J =  10  MAX. SAVING =     68

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  11  J =   8  MAX. SAVING =     64

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  12  J =   8  MAX. SAVING =     64

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  13  J =   8  MAX. SAVING =     64

T(  2,   1) = 2
T(  3,   1) = 2
T(  4,   1) = 2
T(  5,   1) = 2
T(  6,   1) = 2
T(  7,   1) = 2
T(  8,   1) = 1
T(  9,   1) = 1
T( 10,   1) = 1
T( 11,   1) = 1
T( 12,  10) = 1
T( 12,  11) = 1
T( 13,   8) = 1
T( 13,   9) = 1

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  1500
I =   5  Q =  1400
I =   6  Q =  1700
I =   7  Q =  1400
I =   8  Q =  4200
I =   9  Q =  4200
I =  10  Q =  5100
I =  11  Q =  5100
I =  12  Q =     0
I =  13  Q =     0

CAPACITY =    1900  NUMBER ALLOCATED =    6
CAPACITY =    4000  NUMBER ALLOCATED =    0
CAPACITY =    5000  NUMBER ALLOCATED =    1
CAPACITY =    6000  NUMBER ALLOCATED =    1
CAPACITY =  999999  NUMBER ALLOCATED =    0

I =   9  J =   8  MAX. SAVING =     58

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  10  J =   8  MAX. SAVING =     54

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   8  J =   7  MAX. SAVING =     50

T(  2,   1) = 2
T(  3,   1) = 2
T(  4,   1) = 2
T(  5,   1) = 2
T(  6,   1) = 2
T(  7,   1) = 1
T(  8,   7) = 1
T(  9,   1) = 1
T( 10,   1) = 1
T( 11,   1) = 1
T( 12,  10) = 1
T( 12,  11) = 1
T( 13,   8) = 1
T( 13,   9) = 1

I =   2  Q =   1200
I =   3  Q =   1700
I =   4  Q =   1500
I =   5  Q =   1400
I =   6  Q =   1700
I =   7  Q =   5600
I =   8  Q =      0
I =   9  Q =   5600
I =  10  Q =   5100
I =  11  Q =   5100
I =  12  Q =      0
I =  13  Q =      0

CAPACITY =   1900  NUMBER ALLOCATED =    5
CAPACITY =   4000  NUMBER ALLOCATED =    0
CAPACITY =   5000  NUMBER ALLOCATED =    0
CAPACITY =   6000  NUMBER ALLOCATED =    2
CAPACITY = 999999  NUMBER ALLOCATED =    0

I =   9  J =   7  MAX. SAVING =     50

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  10  J =   7  MAX. SAVING =     50

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

I = 11 J = 7 MAX. SAVING = 50

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 12 J = 7 MAX. SAVING = 50

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 13 J = 7 MAX. SAVING = 50

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 13 J = 5 MAX. SAVING = 46

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 8 J = 6 MAX. SAVING = 44

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 11 J = 6 MAX. SAVING = 44

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 12 J = 6 MAX. SAVING = 44

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 13 J = 6 MAX. SAVING = 44

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 9 J = 6 MAX. SAVING = 42

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

```
I =  12  J =   5  MAX. SAVING =     42

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   7  J =   6  MAX. SAVING =     38

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  10  J =   6  MAX. SAVING =     38

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  11  J =   5  MAX. SAVING =     36

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   5  J =   4  MAX. SAVING =     34

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 1
T(  5,  1) = 1
T(  5,  4) = 1
T(  6,  1) = 2
T(  7,  1) = 1
T(  8,  7) = 1
T(  9,  1) = 1
T( 10,  1) = 1
T( 11,  1) = 1
T( 12, 10) = 1
T( 12, 11) = 1
T( 13,  8) = 1
T( 13,  9) = 1

I =   2  Q =  1200
I =   3  Q =  1700
I =   4  Q =  2900
I =   5  Q =  2900
I =   6  Q =  1700
I =   7  Q =  5600
I =   8  Q =     0
I =   9  Q =  5600
I =  10  Q =  5100
I =  11  Q =  5100
I =  12  Q =     0
I =  13  Q =     0
```

```
CAPACITY =   1900  NUMBER ALLOCATED =   3
CAPACITY =   4000  NUMBER ALLOCATED =   1
CAPACITY =   5000  NUMBER ALLOCATED =   0
CAPACITY =   6000  NUMBER ALLOCATED =   2
CAPACITY = 999999  NUMBER ALLOCATED =   0

I =  12  J =   4  MAX. SAVING =    34

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  13  J =   4  MAX. SAVING =    34

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  11  J =   4  MAX. SAVING =    32

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   8  J =   5  MAX. SAVING =    30

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   4  J =   3  MAX. SAVING =    28

T(  2,  1) = 2
T(  3,  1) = 1
T(  4,  3) = 1
T(  5,  1) = 1
T(  5,  4) = 1
T(  6,  1) = 2
T(  7,  1) = 1
T(  8,  7) = 1
T(  9,  1) = 1
T( 10,  1) = 1
T( 11,  1) = 1
T( 12, 10) = 1
T( 12, 11) = 1
T( 13,  8) = 1
T( 13,  9) = 1
```

```
I =   2  Q =  1200
I =   3  Q =  4600
I =   4  Q =     0
I =   5  Q =  4600
I =   6  Q =  1700
I =   7  Q =  5600
I =   8  Q =     0
I =   9  Q =  5600
I =  10  Q =  5100
I =  11  Q =  5100
I =  12  Q =     0
I =  13  Q =     0

CAPACITY =   1900  NUMBER ALLOCATED =   2
CAPACITY =   4000  NUMBER ALLOCATED =   0
CAPACITY =   5000  NUMBER ALLOCATED =   1
CAPACITY =   6000  NUMBER ALLOCATED =   2
CAPACITY = 999999  NUMBER ALLOCATED =   0

I =   6  J =   5  MAX. SAVING =    26

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   8  J =   4  MAX. SAVING =    26

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   9  J =   5  MAX. SAVING =    24

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   6  J =   4  MAX. SAVING =    22

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   5  J =   3  MAX. SAVING =    20

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =   6  J ^   3  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   7  J =   5  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   8  J =   3  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   9  J =   3  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =   9  J =   4  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  10  J =   5  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  11  J =   3  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  12  J =   3  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  13  J =   3  MAX. SAVING =     20
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =    3  J =    2  MAX. SAVING =      18

T(  2,   1) = 1
T(  3,   2) = 1
T(  4,   3) = 1
T(  5,   1) = 1
T(  5,   4) = 1
T(  6,   1) = 2
T(  7,   1) = 1
T(  8,   7) = 1
T(  9,   1) = 1
T( 10,   1) = 1
T( 11,   1) = 1
T( 12,  10) = 1
T( 12,  11) = 1
T( 13,   8) = 1
T( 13,   9) = 1

I =    2  Q =   5800
I =    3  Q =      0
I =    4  Q =      0
I =    5  Q =   5800
I =    6  Q =   1700
I =    7  Q =   5600
I =    8  Q =      0
I =    9  Q =   5600
I =   10  Q =   5100
I =   11  Q =   5100
I =   12  Q =      0
I =   13  Q =      0

CAPACITY =   1900  NUMBER ALLOCATED =    1
CAPACITY =   4000  NUMBER ALLOCATED =    0
CAPACITY =   5000  NUMBER ALLOCATED =    0
CAPACITY =   6000  NUMBER ALLOCATED =    3
CAPACITY = 999999  NUMBER ALLOCATED =    0

I =    4  J =    2  MAX. SAVING =      18

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =    7  J =    3  MAX. SAVING =      16

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  7  J =  4  MAX. SAVING =    16
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 10  J =  3  MAX. SAVING =    16
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 10  J =  4  MAX. SAVING =    16
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  5  J =  2  MAX. SAVING =    10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  6  J =  2  MAX. SAVING =    10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  7  J =  2  MAX. SAVING =    10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  8  J =  2  MAX. SAVING =    10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I =  9  J =  2  MAX. SAVING =    10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3

I = 10  J =  2  MAX. SAVING =    10
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =  11   J =   2  MAX. SAVING =     10

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  12   J =   2  MAX. SAVING =     10

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =  13   J =   2  MAX. SAVING =     10

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


              ****************
ROUTE  1       FROM       TO
              ****************

              ORIGIN       10
                10         12
                12         11
                11       ORIGIN

     DISTANCE FOR ROUTE IS   100 MILES

     ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF  6000 UNITS


              ****************
ROUTE  2       FROM       TO
              ****************

              ORIGIN        7
                 7          8
                 8         13
                13          9
                 9       ORIGIN

     DISTANCE FOR ROUTE IS   104 MILES

     ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF  6000 UNITS
```

```
          ****************
ROUTE  3    FROM        TO
          ****************

          ORIGIN         2
             2           3
             3           4
             4           5
             5        ORIGIN

   DISTANCE FOR ROUTE IS    54 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF  6000 UNITS


          ****************
ROUTE  4    FROM        TO
          ****************

          ORIGIN         6
             6        ORIGIN

   DISTANCE FOR ROUTE IS    44 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF  1900 UNITS


TOTAL DISTANCE FOR ALL ROUTES IS   302 MILES


FINAL ALLOCATION

CAPACITY =   1900  NUMBER ALLOCATED =   1
CAPACITY =   4000  NUMBER ALLOCATED =   0
CAPACITY =   5000  NUMBER ALLOCATED =   0
CAPACITY =   6000  NUMBER ALLOCATED =   3
```

SAMPLE PROBLEM 3

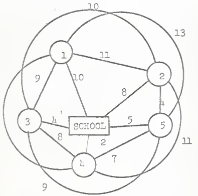
The dynamic programming solution to the carrier routing problem developed by Tillman [12] is included as it was presented by him. The numbering system used in the presentation does not correspond to that used by the computer program. However, the two solutions may be easily compared by referring to the following table.

Table 10. Comparison of numbering systems used.

DYNAMIC PROGRAMMING NUMBERING	COMPUTER PROGRAM NUMBERING
SCHOOL	1
1	6
2	5
3	3
4	2
5	4

DYNAMIC PROGRAMMING SOLUTION TO THE SCHOOL BUS SCHEDULING PROBLEM

Nearly all school districts have the problem of scheduling school busses for transporting students to school and home again. In determing the schedule, the objective is usually to minimize the number of miles traveled while fully utilizing the busses. The following example illustrates the problem of scheduling two busses to pick up passengers at five stops. Each bus has a capacity of twenty passengers and one bus makes two trips which increases the fleet size to the equivalent of three busses. The distances and number of passengers are illustrated in the following figure:



Distance to School

Distance from School	1	2	3	4	5	to sch.
1	0	11	9	12	13	10
2	11	0	10	11	4	8
3	9	10	0	8	9	4
4	12	11	8	0	7	2
5	13	4	9	7	0	5
fr. sch	10	8	4	2	5	0

STOP	NO. OF PASSENGERS
1	10
2	8
3	6
4	9
5	7
TOTAL	40

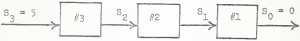
Fig. 4

In this example the objective is to schedule the busses so that the number of miles traveled is a minimum.

The dynamic programming solution to this problem follows.

Decision Variable	P_3	P_2	P_1
Stage Input	S_3	S_2	S_1
Stage Trans.	$S_3 - P_3 = S_2$	$S_2 - P_2 = S_1$	$S_1 - P_1 = S_0$

Stages



RESTRAINTS

1) Number of Stops $\quad P_3 \leq 5 \qquad P_3 + P_2 \leq 5 \qquad \sum_{i=1}^{3} P_i \leq 5$

2) Bus Capacity $\quad \sum_{i=1}^{P_3} N_{i3} \leq 20 \qquad \sum_{i=1}^{P_2} N_{i2} \leq 20 \qquad \sum_{i=1}^{P_1} N_{i1} \leq 20$

N_{ij} = No. of passengers at stop i for bus j

P_j = Stops made by bus j $\qquad i = 1,\ldots, 5$

S_j = Stops yet to be made at stage or bus j $\qquad j = 1,\ldots, 3$

Returns:	Min. No. of Miles Necessary to Load Bus #3	Min. No. of Miles Necessary to Load Bus #2	Min. No. of Miles Necessary to Load Bus #1 Which is Bus Which Returned
	D_3	D_2	D_1

OBJECTIVE: $\quad \text{Min } Z = \sum_{j=1}^{3} D_j$

To convert this problem to a maximizing problem subtract every distance from 15 and maximize the complement of the distances to each stop. This insures that the busses are loaded and that the distance traveled is minimized. This is illustrated in Figure 5.

To School

From School	1	2	3	4	5	To Sch.
1	x	4	6	3	2	5
2	4	x	5	4	11	7
3	6	5	x	7	6	11
4	3	4	7	x	8	13
5	2	11	6	8	x	10
From Sch.	5	7	11	13	10	x

Fig. 5

It is noted that for this example, it is necessary for two busses to make two stops and one bus one stop, so that all stops are made and the bus capacities are not exceeded.

The decision at the first stage for the various values of S_1 is as follows:

P_1 & S_1	$f_1(S_1)$
0	0
1	10
2	14
3	22
4	26
5	20
1,2	16
1,3	22
1,4	21
1,5	17
2,3	23
2,4	24
2,5	28
3,4	31
3,5	27
4,5	31

The decision at the second stage and the return of the second and the first stage,

$$f_2(S_2) = D_2 + f_1(S_1),$$

for the various values of S_2 is as follows:

S_2	P_2'	or	P_2''	$f_2(S_2)$	S_1'	or	S_1''
1,2,3	3		1,2	38	1,2		3
1,2,4	4		1,2	42	1,2		4
	{1		2,5		2,5		1
1,2,5	5		1,2	38	1,2		5
1,3,4	4		1,3	48	1,3		4
1,3,5	5		1,3	42	1,3		5
1,4,5	4		1,5	43	1,5		4
2,3,4	4		2,3	49	2,3		4
2,3,5	5		2,3	43	2,3		5
2,4,5	4		2,5	54	2,5		4
	{3		4,5		4,5		3
3,4,5	5		3,4	53	3,4		5
1,2,3,4	1,2		3,4	47	3,4		1,2
1,2,3,5	1,3		2,5	50	2,5		1,3
2,3,4,5	2,5		3,4	59	3,4		2,5
1,2,4,5	1,4		2,5	49	2,5		1,4
1,3,4,5	1,3		4,5	53	4,5		1,3

If P_2' is made then S_1' is the input to the first stage and if P_2'' is made then S_1'' is the input to the first stage.

Finally the decision at the third stage and the return at the third and the second stage,

$$f_3(S_3) = D_3 + f_2(S_2),$$

for S_3 is as follows:

For S_3 = 1, 2, 3, 4, 5

P_3	$f_3(S_3)$	S_2
1	10 + 59 = 69	2,3,4,5
2	14 + 53 = 67	1,3,4,5
3	22 + 49 = 71	1,2,4,5
(1) 4	26 + 50 = 76	1,2,3,5
5	20 + 47 = 67	1,2,3,4
1,2	16 + 53 = 69	3,4,5
(2) 1,3	22 + 54 = 76	2,4,5
1,4	21 + 43 = 64	2,3,5
1,5	17 + 49 = 66	2,3,4
2,3	23 + 43 = 66	1,4,5
2,4	24 + 42 = 66	1,3,5
(3) 2,5	28 + 48 = 76	1,3,4
3,4	31 + 38 = 69	1,2,5
3,5	27 + 42 = 69	1,2,4
4,5	31 + 38 = 69	1,2,3

From the above there are three optimal decisions with their associated distance as follows:

Optimal Decision	Bus #3	Bus #2	Bus #1	
1	$P_3 = 4$	$P_2 = 1,3$	$P_1 = 2,5$	Total = 44 Miles
2	$P_3 = 1,3$	$P_2 = 2,5$	$P_1 = 4$	Total = 44 Miles
3	$P_3 = 2,5$	$P_2 = 4$	$P_1 = 1,3$	Total = 44 Miles

The complete search method yields the following results.

P_3	P_2	P_1	D	
1	2,3	4,5	56	
1	2,4	3,5	59	
1	2,5	4,3	51	
2	1,3	4,5	53	
2	1,4	3,5	58	
2	1,5	4,3	58	
3	1,2	4,5	51	
3	1,4	2,5	49	
3	1,5	4,2	57	
4	1,2	3,5	51	
4	1,3	2,5	44	OPTIMAL
4	1,5	2,3	54	
5	1,2	3,4	53	
5	1,3	2,4	54	
5	1,4	2,3	56	

15 Schedules Total

There are 3! or 6 [15] = 90 schedules for the three busses but the remaining 75 are included in the above for this problem.

A MULTISTAGE PROCESS WITH N STAGES

Decision Variable:	P_N	....	P_2	P_1
Stage Input:	S_N		S_2	S_1
Stage Transformation:	$T_N(P_N,S_N)$ S_{N-1}		$T_2(P_2,S_2)$ S_1	$T_1(P_1,S_1)$ S_0
Stages:	$S_N \rightarrow \boxed{N} \rightarrow S_{N-1}$		$S_2 \rightarrow \boxed{2}$ — S_1	$\rightarrow \boxed{1} \rightarrow S_0$
Restraints:	P_N & R_N		P_2 & R_2	P_1 & R_1
Returns:	$D_N(P_N,S_N)$	....	$D_2(P_2,S_2)$	$D_1(P_1,S_1)$

Table 11. Distance half matrix and delivery vector for sample problem 3.

Q	P_1					
9	2	P_2				
6	4	8	P_3			
7	5	7	9	P_4		
8	8	11	10	4	P_5	
10	10	12	9	13	11	P_6

Table 12. Carrier availabilities and capacities for sample problem 3.

CAPACITY	20
NUMBER AVAILABLE	3

```
********************************
*                              *
* SOLUTION FOR SAMPLE PROBLEM 3 *
*                              *
********************************


INITIAL ALLOCATION

CAPACITY =     20  NUMBER ALLOCATED =    5
CAPACITY = 999999  NUMBER ALLOCATED =    0


SAVINGS MATRIX

ROW   COL   SAVING

  2     1   -99999
  3     1   -99999
  3     2       -2
  4     1   -99999
  4     2        0
  4     3        0
  5     1   -99999
  5     2       -1
  5     3        2
  5     4        9
  6     1   -99999
  6     2        0
  6     3        5
  6     4        2
  6     5        7

I =    5  J =    4  MAX. SAVING =       9

T(  2,  1) = 2
T(  3,  1) = 2
T(  4,  1) = 1
T(  5,  1) = 1
T(  5,  4) = 1
T(  6,  1) = 2

I =    2  Q =      9
I =    3  Q =      6
I =    4  Q =     15
I =    5  Q =     15
I =    6  Q =     10

CAPACITY =     20  NUMBER ALLOCATED =    4
CAPACITY = 999999  NUMBER ALLOCATED =    0
```

```
I =   6  J =   5  MAX. SAVING =      7

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   6  J =   3  MAX. SAVING =      5

T(  2,  1) = 2
T(  3,  1) = 1
T(  4,  1) = 1
T(  5,  1) = 1
T(  5,  4) = 1
T(  6,  1) = 1
T(  6,  3) = 1

I =   2  Q =     9
I =   3  Q =    16
I =   4  Q =    15
I =   5  Q =    15
I =   6  Q =    16

CAPACITY =     20  NUMBER ALLOCATED =    3
CAPACITY = 999999  NUMBER ALLOCATED =    0

I =   5  J =   3  MAX. SAVING =      2

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   6  J =   4  MAX. SAVING =      2

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   4  J =   2  MAX. SAVING =      0

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


I =   4  J =   3  MAX. SAVING =      0

MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3
```

```
I =   6  J =   2  MAX. SAVING =      0
MAX. SAVING DOES NOT SATISFY ONE OR MORE OF CONDITIONS 1 THROUGH 3


              ***************
ROUTE  1      FROM       TO
              ***************

             ORIGIN       4
                4         5
                5      ORIGIN

   DISTANCE FOR ROUTE IS    17 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF    20 UNITS

              ***************
ROUTE  2      FROM       TO
              ***************

             ORIGIN       3
                3         6
                6      ORIGIN

   DISTANCE FOR ROUTE IS    23 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF    20 UNITS

              ***************
ROUTE  3      FROM       TO
              ***************

             ORIGIN       2
                2      ORIGIN

   DISTANCE FOR ROUTE IS     4 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF    20 UNITS

TOTAL DISTANCE FOR ALL ROUTES IS    44 MILES


FINAL ALLOCATION

CAPACITY =      20  NUMBER ALLOCATED =   3
```

Table 13. Distance half matrix and delivery vector for sample problem 4.

Q	P_1													
1000	25	P_2												
8700	32	8	P_3											
19500	48	31	32	P_4										
8580	51	77	84	78	P_5									
6400	63	87	94	10	88	P_6								
12200	65	59	53	105	25	111	P_7							
12120	92	85	79	123	43	137	26	P_8						
7800	100	88	101	149	121	156	128	154	P_9					
4550	133	155	162	113	158	98	173	199	223	P_{10}				
4000	161	193	200	118	196	94	211	237	250	38	P_{11}			
10500	186	190	197	188	206	98	211	237	289	85	94	P_{12}		
12000	212	215	223	214	222	185	237	263	315	111	120	26	P_{13}	
37260	222	234	228	223	188	232	173	164	301	166	192	145	159	P_{14}

Table 14. Carrier availabilities and capacities for sample problem 4.

CAPACITY	45000
NUMBER AVAILABLE	∞

```
********************************
*                              *
* SOLUTION FOR SAMPLE PROBLEM 4 *
*                              *
********************************


          ****************
ROUTE  1   FROM       TO
          ****************

          ORIGIN      10
            10        11
            11        12
            12        13
            13      ORIGIN

   DISTANCE FOR ROUTE IS   503 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF 45000 UNITS


          ****************
ROUTE  2   FROM       TO
          ****************

          ****************
ROUTE  3   FROM       TO
          ****************

          ORIGIN       5
             5         8
             8         7
             7      ORIGIN

   DISTANCE FOR ROUTE IS   185 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF 45000 UNITS
```

```
         ****************
ROUTE 4   FROM       TO
         ****************

         ORIGIN      6
            6        4
            4        3
            3        2
            2        9
            9     ORIGIN

   DISTANCE FOR ROUTE IS   301 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF 45000 UNITS


         ****************
ROUTE 5   FROM       TO
         ****************

         ****************
ROUTE 6   FROM       TO
         ****************

         ORIGIN      14
           14     ORIGIN

   DISTANCE FOR ROUTE IS   444 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF 45000 UNITS

TOTAL DISTANCE FOR ALL ROUTES IS  1433 MILES


FINAL ALLOCATION

CAPACITY =  45000  NUMBER ALLOCATED =    4
```

Table 15. Distance half matrix and delivery vector for sample problem 5.

Q	P_1																									
60	2	P_2																								
80	5	5	P_3																							
50	12	14	17	P_4																						
90	14	14	19	26	P_5																					
15	18	20	23	30	32	P_6																				
60	18	18	23	30	32	23	P_7																			
100	20	20	25	32	6	38	38	P_8																		
60	22	24	27	22	36	40	40	42	P_9																	
60	24	26	29	36	38	5	28	44	46	P_{10}																
30	24	24	29	36	31	29	14	31	46	34	P_{11}															
90	27	29	32	31	41	45	45	47	37	51	51	P_{12}														
60	35	35	40	47	36	40	25	42	57	45	23	62	P_{13}													
80	35	35	40	47	36	40	25	42	57	45	23	62	0	P_{14}												
60	35	35	40	47	42	33	17	48	57	38	17	62	42	42	P_{15}											
40	41	41	46	53	45	41	23	56	63	46	20	68	26	26	11	P_{16}										
60	72	72	77	84	73	73	55	79	94	78	48	99	48	48	45	34	P_{17}									
60	80	80	85	92	81	98	70	87	102	104	68	107	48	48	60	49	33	P_{18}								
30	91	91	96	103	102	91	73	106	113	96	87	118	98	98	55	50	78	99	P_{19}							
30	93	93	98	105	102	93	75	108	115	98	89	120	100	100	57	52	80	101	8	P_{20}						
50	97	97	102	109	108	97	79	114	119	102	93	124	104	104	61	56	84	105	12	14	P_{21}					
60	133	135	138	121	147	130	151	153	143	125	157	156	168	168	168	174	205	213	133	142	145	P_{22}				
30	133	135	138	121	147	130	151	158	143	125	157	156	168	168	168	174	205	213	133	142	145	3	P_{23}			
90	135	137	140	123	149	132	153	155	145	127	159	158	170	170	170	176	207	215	132	141	144	8	6	P_{24}		
30	138	140	143	126	152	135	156	158	148	130	162	161	173	173	173	179	210	218	135	144	147	10	9	3	P_{25}	
30	141	143	146	129	155	138	159	161	151	133	165	164	176	176	176	155	213	221	141	150	153	8	11	13	18	P_{26}

Table 16. Carrier availabilities and capacities for sample problem 5.

CAPACITY	120
NUMBER AVAILABLE	∞

```
********************************
*                              *
* SOLUTION FOR SAMPLE PROBLEM 5 *
*                              *
********************************


          ****************
ROUTE  1    FROM        TO
          ****************

          ORIGIN       24
            24         25
            25       ORIGIN

   DISTANCE FOR ROUTE IS   276 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS


          ****************
ROUTE  2    FROM        TO
          ****************

          ORIGIN       23
            23         22
            22         26
            26       ORIGIN

   DISTANCE FOR ROUTE IS   285 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS


          ****************
ROUTE  3    FROM        TO
          ****************

          ORIGIN       20
            20         19
            19         21
            21       ORIGIN

   DISTANCE FOR ROUTE IS   210 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS
```

```
         ****************
ROUTE  4   FROM        TO
         ****************

         ORIGIN       17
           17         18
           18       ORIGIN

   DISTANCE FOR ROUTE IS   185 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS


         ****************
ROUTE  5   FROM        TO
         ****************

         ORIGIN       15
           15         16
           16       ORIGIN

   DISTANCE FOR ROUTE IS    87 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS


         ****************
ROUTE  6   FROM        TO
         ****************

         ORIGIN        6
            6         10
           10       ORIGIN

   DISTANCE FOR ROUTE IS    47 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS


         ****************
ROUTE  7   FROM        TO
         ****************

         ORIGIN       11
           11         13
           13       ORIGIN

   DISTANCE FOR ROUTE IS    82 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS
```

```
         ****************
ROUTE  8    FROM        TO
         ****************

         ORIGIN          4
            4            9
            9       ORIGIN

   DISTANCE FOR ROUTE IS     56 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF    120 UNITS


         ****************
ROUTE  9    FROM        TO
         ****************

         ORIGIN          2
            2            7
            7       ORIGIN

   DISTANCE FOR ROUTE IS     38 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF    120 UNITS


         ****************
ROUTE 10    FROM        TO
         ****************

         ORIGIN          3
            3       ORIGIN

   DISTANCE FOR ROUTE IS     10 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF    120 UNITS


         ****************
ROUTE 11    FROM        TO
         ****************

         ORIGIN          5
            5       ORIGIN

   DISTANCE FOR ROUTE IS     28 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF    120 UNITS
```

```
         ****************
ROUTE 12   FROM        TO
         ****************

         ORIGIN        8
            8       ORIGIN

   DISTANCE FOR ROUTE IS    40 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS

         ****************
ROUTE 13   FROM        TO
         ****************

         ORIGIN       12
           12       ORIGIN

   DISTANCE FOR ROUTE IS    54 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS

         ****************
ROUTE 14   FROM        TO
         ****************

         ORIGIN       14
           14       ORIGIN

   DISTANCE FOR ROUTE IS    70 MILES

   ROUTE REQUIRES A CARRIER HAVING A CAPACITY OF   120 UNITS

TOTAL DISTANCE FOR ALL ROUTES IS  1468 MILES


FINAL ALLOCATION

CAPACITY =     120  NUMBER ALLOCATED =  14
```

OPTIMIZATION OF A CARRIER ROUTING PROBLEM

by

HAROLD MERLIN COCHRAN

B. S., Kansas State University, 1965

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

The purpose of this study was to evaluate methods for the solution of large scale carrier routing problems.

An algorithm developed by Clarke and Wright [4] was chosen for further study and modification. The modifications allowed for the inclusion of additional restraints on the system. The particular restraint which was incorporated limited the number of miles which could be traveled by a carrier on its allocated route. A modified allocation procedure was suggested which the author believes will make better use of the available carriers, thus resulting in a better solution in terms of total miles traveled. Improvements in the computational procedure were also suggested.

The modified algorithm was programmed for the IBM 1620 computer and several sample problems were then solved.

Experience with the method has shown that the modified algorithm is practicable and efficient for solving large scale problems. Even though it does not guarantee an optimal solution, it appears to be the "best" method currently available for the solution of practical large scale routing problems.