Capacitated Arc Routing Problems

Bruce L. Golden
University of Maryland, College Park, Maryland 20742

Richard T. Wong
Rensselaer Polytechnic Institute, Troy, New York 12181

A capacitated node routing problem, known as the vehicle routing or dispatch problem, has been the focus of much research attention. On the other hand, capacitated arc routing problems have been comparatively neglected. Both classes of problems are extremely rich in theory and applications. Our intent in this paper is to define a capacitated arc routing problem, to provide mathematical programming formulations, to perform a computational complexity analysis, and to present an approximate solution strategy for this class of problems. In addition, we identify several related routing problems and develop tight lower bounds on the optimal solution.

I. INTRODUCTION

Uncapacitated routing problems can be classified as node routing problems, arc routing problems, or general routing problems (see Bodin's taxonomy [2]). The problem of visiting all nodes in a network and returning to the starting point (node routing) while incurring minimal cost is the traveling salesman problem (TSP). The problem of covering all arcs in a network while minimizing the total distance traveled (arc routing) is the Chinese postman problem (CPP). The general routing problem (GRP) on network $G(N, E, C)$ (N is the set of all nodes, E the set of all arcs, and C the matrix of costs) is a generalization which includes the TSP and CPP as special cases. Here we seek the minimum-cost cycle which visits every node in subset $Q \subseteq N$ and covers every arc in subset $R \subseteq E$. In each case, we assume that all arcs are undirected and that there is a vehicle of unlimited capacity. We point out that the terms *cost* and *distance* are used interchangeably in this paper.

Capacitated variations, of course, reflect real-life situations more directly. A capacitated node routing problem, known as the vehicle routing problem, has been the focus of much research attention (e.g., see Beltrami and Bodin [1], Golden et al. [8], and Russell [14]). On the other hand, capacitated arc routing problems, also extremely practical, have been comparatively neglected (the recent work by Stern and Dror [17] is an exception). Our paper is primarily concerned with this class of problems. Our intent will be to define a capacitated arc routing problem, to provide mathematical programming formulations, to perform a computational complexity analysis, and to present an approximate solution strategy for this class of problems. In addi-

NETWORKS, Vol. 11 (1981) 305-315
© 1981 John Wiley & Sons, Inc.
CCC 0028-3045/81/030305-11$01.00

tion, we will identify some related routing problems and develop tight lower bounds on the optimal solution.

Applications of arc routing problems include routing of street sweepers, snow plows, household refuse collection vehicles, the spraying of roads with salt-grit to prevent ice formation, the inspection of electric power lines, and gas or oil pipelines for faults, etc.

The capacitated arc routing problem (CARP) that we focus on is as follows: given an undirected network $G(N, E, C)$ with arc demands $q_{ij} \geqslant 0$ for each arc (i, j) which must be satisfied by one of a fleet of vehicles of capacity W, find a number of cycles each of which passes through the domicile (node 1) which satisfy demands at minimal total cost. Several related problems are posed below.

1. The Chinese postman problem (CPP). Find a minimum-cost cycle which traverses every arc in E at least once.
2. The rural postman problem (RPP). Find a minimum-cost cycle which traverses each arc in a given subset $R \subseteq E$ at least once. See Orloff [12, 13] for more discussion and Lenstra and Rinnooy Kan [11] for an NP-completeness proof.
3. The capacitated Chinese postman problem (CCPP). Given arc demands $q_{ij} > 0$ for each arc (i, j) which must be satisfied by vehicles of capacity W, find a set of cycles each of which passes through the domicile which satisfy demands at minimal total cost. Christofides [3] proposes a heuristic procedure for obtaining reasonable solutions to this problem and Golden [7] formulates it as a set covering problem.
4. The traveling salesman problem (TSP). Find a minimum-cost tour which visits each node in N exactly once.
5. The vehicle routing problem (VRP). Given node demands d_i for each node i which must be satisfied by vehicles of capacity W, find tours which satisfy demand at minimal total cost and which originate at a central domicile.
6. The general routing problem (GRP). Find a minimum-cost cycle which visits every node in subset $Q \subseteq N$ and covers every arc in subset $R \subseteq E$. This problem was first addressed by Orloff [12].

Now we examine the relationship between the CARP and these other problems. If we assume $q_{ij} > 0$ for all $(i, j) \in E$, the CARP reduces to the CCPP. If, in addition, we have only one vehicle of capacity $W \geqslant \Sigma_i \Sigma_j q_{ij}$, we obtain the CPP. With one vehicle of capacity $W \geqslant \Sigma_i \Sigma_j q_{ij}$ and $q_{ij} > 0$ for all arcs $(i, j) \in R \subseteq E$, we have the RPP. Thus, the CCPP, the CPP, and the RPP are special cases of the CARP. Since we can incorporate the constraint that a node must be serviced by splitting that node into two nodes joined by a zero-length arc with a demand equal to the original node demand, the TSP, VRP, and GRP are also special cases of the CARP.

II. INTEGER PROGRAMMING FORMULATIONS OF THE CARP

A mathematical programming formulation for the CARP is given below.

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{K} c_{ij} x_{ij}^{p} \tag{1}$$

subject to

$$\sum_{k=1}^{n} x_{ki}^{p} - \sum_{k=1}^{n} x_{ik}^{p} = 0 \quad \text{for } i = 1, \ldots, n \quad p = 1, \ldots, K, \tag{2}$$

$$\sum_{p=1}^{K} (l_{ij}^p + l_{ji}^p) = \left\lceil \frac{q_{ij}}{W} \right\rceil \qquad \text{for } (i,j) \in E, \tag{3}$$

$$x_{ij}^p \geqslant l_{ij}^p \qquad \text{for } (i,j) \in E,\ p = 1, \ldots, K, \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} l_{ij}^p q_{ij} \leqslant W \qquad \text{for } p = 1, \ldots, K, \tag{5}$$

$$\left.\begin{aligned} &\sum_{i \in \tilde{Q}} \sum_{j \in \tilde{Q}} x_{ij}^p - n^2 y_{1\tilde{q}}^p \leqslant |\tilde{Q}| - 1 \\ &\sum_{i \in \tilde{Q}} \sum_{j \notin \tilde{Q}} x_{ij}^p + y_{2\tilde{q}}^p \geqslant 1 \\ &y_{1\tilde{q}}^p + y_{2\tilde{q}}^p \leqslant 1;\ y_{1\tilde{q}}^p, y_{2\tilde{q}}^p \in \{0,1\} \\ &x_{ij}^p, l_{ij}^p \in \{0,1\} \end{aligned}\right\} \quad \begin{aligned} &\text{for } p = 1, \ldots, K \quad (6) \\ &\tilde{q} = 1, \ldots, 2^{n-1} - 1 \\ &\text{and every nonempty} \\ &\text{subset } \tilde{Q} \text{ of } \{2, 3, \ldots, n\}, \quad (7) \end{aligned}$$

where n = the number of nodes; K = the number of available postmen or vehicles; q_{ij} = the demand on arc (i, j); W = the vehicle capacity ($W \geqslant \max q_{ij}$); c_{ij} = the length of arc (i, j); $x_{ij}^p = 1$, if arc (i, j) is traversed by postman p, 0 otherwise; $l_{ij}^p = 1$ if postman p services arc (i, j), 0 otherwise; $\lceil z \rceil$ = the smallest integer greater than or equal to z.

The objective function (1) seeks to minimize total distance traveled. Equations (2) ensure route continuity. Equations (3) state that each arc with positive demand is serviced exactly once. Equations (4) guarantee that arc (i, j) can be serviced by postman p only if he covers arc (i, j). Vehicle capacity is not violated on account of eqs. (5). Equations (6) prohibit the formation of illegal subtours. We point out that each index $\tilde{q}$ corresponds to a set $\tilde{Q}$. Integrality restrictions are given in (7).

Equations (6) may require a bit more in the way of explanation. In Figure 1, illegal and legal subtours are exhibited. In constraints (6), we want to prevent the former but

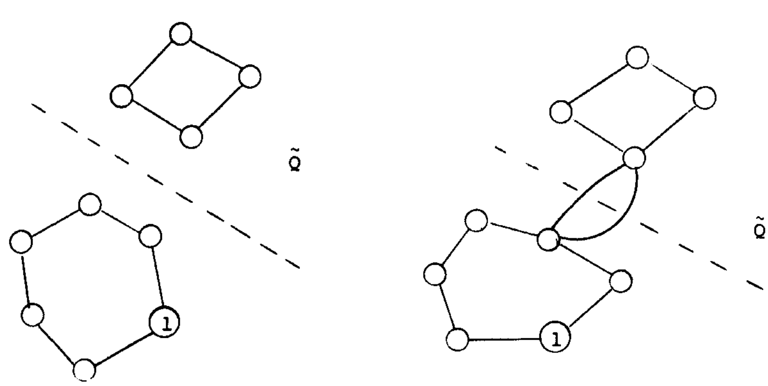

FIG. 1. Illegal and legal subtours.

allow the latter type of configuration. This is taken care of since

$$\sum_{i\in\tilde{Q}}\sum_{j\in\tilde{Q}} x_{ij}^p > |\tilde{Q}| - 1 \Rightarrow y_{1\tilde{q}}^p = 1 \Rightarrow y_{2\tilde{q}}^p = 0 \Rightarrow \sum_{i\in\tilde{Q}}\sum_{j\notin\tilde{Q}} x_{ij}^p \geqslant 1.$$

Formulating the capacitated arc routing problem is important for several reasons. First, the CARP formulation provides a unified framework for viewing a number of related combinatorial optimization problems. Second, the underlying mathematical structure and the complexity of the linear-integer program become visible. The formulation (1)-(7), however, is rather messy in that (6) consists of $3K\{2^{n-1} - 1\}$ inequalities.

We next present an alternative set of subtour-breaking constraints which is much more compact. The new mathematical program is the same as before except that the set of equations in (6) is replaced by (6′)

$$\begin{aligned}
&\sum_{k=1}^{n} f_{ik}^p - \sum_{k=1}^{n} f_{ki}^p = \sum_{j=1}^{n} l_{ij}^p && \text{for } i = 2, \ldots, n \\
& && \quad p = 1, \ldots, K, \\
&f_{ij}^p \leqslant n^2 x_{ij}^p && \text{for } (i,j) \in E \\
&f_{ij}^p \geqslant 0. && \quad p = 1, \ldots, K,
\end{aligned} \tag{6′}$$

Note that f_{ij}^p is a flow variable which can take on positive values only if $x_{ij}^p = 1$. The new constraints again prohibit the formation of illegal subtours. Furthermore, the number of constraints in the new formulation is now less than $3Kn^2$.

III. COMPUTATIONAL COMPLEXITY OF THE CARP

Recent research in computational complexity theory has focused on the class of NP-complete problems [6, 9, 10, 15] which includes such notoriously difficult combinatorial problems as the traveling salesman problem, the general integer programming problem, and the multi-commodity network flow problem. An NP-completeness proof is a strong indication that the problem is intractable. A problem is NP-hard [6] if the existence of a polynomially bounded algorithm for it implies the existence of a polynomially bounded algorithm for all NP-complete problems. In this section, we show that even an approximate, restricted version of the CARP is NP-hard. Unless otherwise stated, we assume that the triangle inequality holds for all arc lengths. At the end of this section, we briefly discuss the implications of relaxing this condition. In order to prove the NP-hard result, we need several definitions:

Definition. The capacitated Chinese postman problem (CCPP). Given a set of nodes N and arcs E, a matrix of arc lengths $[c_{ij}]$, a matrix of positive arc demands $[q_{ij}]$, a postman capacity W, and a designated depot, find a set of postman routes (cycles passing through the depot) such that the total distance traveled is minimized.

Recall that the CCPP is a special case of the CARP.

Definition. The 0.5-approximate capacitated Chinese postman problem is to find a CCPP solution whose cost is less than 1.5 times the optimal solution.

Definition. The partition problem. Given a set of $N (N \geq 2)$ positive integers $\{d_1, d_2, \ldots, d_N\}$, determine if this set can be divided into two subsets such that the sum of the elements in each set is equal.

Karp [9] has shown that the partition problem is NP-complete.

Theorem 1. The 0.5-approximate capacitated Chinese postman problem is NP-hard.

Proof: We show that the NP-complete partition problem can be reduced in polynomial time to the 0.5-approximate CCPP. This will prove the theorem. Figure 2 depicts the CCPP on a tree. We transform an arbitrary partition problem into a CCPP of the type depicted in Figure 2 as follows. Let N be the number of integers in the partition problem. Set $c_{bi} = 0$ for $i = 1, 2, \ldots, N$ and $c_{ab} = 1$. Any arc (i, j) not contained in the tree is assumed to have a length equal to the distance in the tree between nodes i and j. Also, $q_{ab} = 1, q_{bi} = 10\, d_i$ for $i = 1, 2, \ldots, N$, and $W = \frac{1}{2}\ \{\Sigma_{i=1}^{N} 10 d_i\} + 1$, where the d_i are the integers in the partition problem. Let node a be the depot node in the CCPP.

If the partition problem admits a partitioning of the input set, then v = total cost of the CCPP in Figure 2 = 4. Otherwise, $v = 6$. Thus, we could solve the partitioning problem by considering the transformed problem and then checking to see if the 0.5-approximate CCPP solution is less than 6. So we see that the partitioning problem can be reduced in polynomial time to a special case of the 0.5-approximate CCPP. Therefore, the 0.5-approximate CCPP must be NP-hard.

This demonstrates that even finding a near-optimal (within 50%) solution to a special case of the CARP is extremely difficult. So the general CARP must also be hard to solve.

Corollary 1. Define the 0.5-approximate vehicle routing problem as the problem of obtaining a VRP solution whose cost is less than 1.5 times the optimal solution. This problem is also NP-hard.

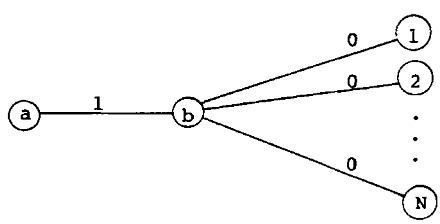

FIG. 2. CCPP tree network.

This follows since the VRP on a tree is a special case of the general VRP. If in Figure 2, we replace arc demands by node demands

$$Q(b) = 1$$
$$Q(i) = 10\, d_i \text{ for } i = 1, 2, \ldots, N,$$

then the corollary follows.

Our results indicate that even when the triangle inequality holds, finding a near-optimal (within 50%) solution for the CARP and also the VRP is extremely difficult. This degree of computational complexity arises on account of the vehicle capacity constraint since the proof of these results crucially depends on this constraint.

For problems where the triangle inequality does not necessarily hold, an even stronger result applies. Sahni and Gonzalez [16] have shown that the problem of finding a k-approximate solution (for any finite k) to the TSP without the triangle inequality condition is NP-hard. Since the TSP is a special case of the CARP and VRP, the same result holds for these two problems when there is *no* triangle inequality restriction.

For the TSP with the triangle inequality condition, the situation is much different; Christofides [4] has described a polynomial-bounded algorithm that solves the 0.5-approximate TSP. At this point, it is not known whether the ϵ-approximate TSP with the triangle inequality condition is NP-hard for $\epsilon < \frac{1}{2}$.

Therefore, for the common cases of the CARP and VRP problems *with* the triangle inequality, Theorem 1 and Corollary 1 state the strongest complexity results that are currently known.

IV. LOWER BOUNDS FOR CAPACITATED ARC ROUTING PROBLEMS

Given the computational complexity of the CARP, it becomes necessary to apply approximate solution techniques or heuristics. These procedures generate upper bounds on the optimal solution. In order to assess deviations from optimality, tight lower bounds are desired as well.

First, we point out that in the otherwise excellent paper by Christofides [3], an incorrect lower bounding procedure for the CCPP is given. Suppose the CCPP is solved as a CPP. Clearly, this gives a lower bound. Since $M = \lceil \Sigma_i \Sigma_j q_{ij}/W \rceil$ is the minimal number of cycles required in the CCPP solution, then twice that many arcs must be incident to the depot node. If the solution to the CPP has I nodes incident to the depot, then at least $2M - I$ additional arcs are required. Christofides next argues that we can multiply the length of the shortest arc incident to the depot by $2M - I$, and add this to the CPP solution to obtain a tighter lower bound on the CCPP. This is not true as indicated in Figure 3. Here, we assume that node 1 is the depot, $W = 3$, and $q_{ij} = 1$ for all (i, j). Now the Chinese postman problem solution is to traverse arc (2, 3) twice yielding a total cost of 22 units. According to Christofides' lower bound, the total cost should be at least 22 + 2(3) = 28. But the capacitated postman routes are $\underline{1\ 3\ 2}\ 1$ and $1\ \underline{3\ 4\ 2\ 1}$ (underlines indicate serviced arcs) with total cost 24.

It is possible to obtain in polynomial time a valid lower bound for the CARP if we

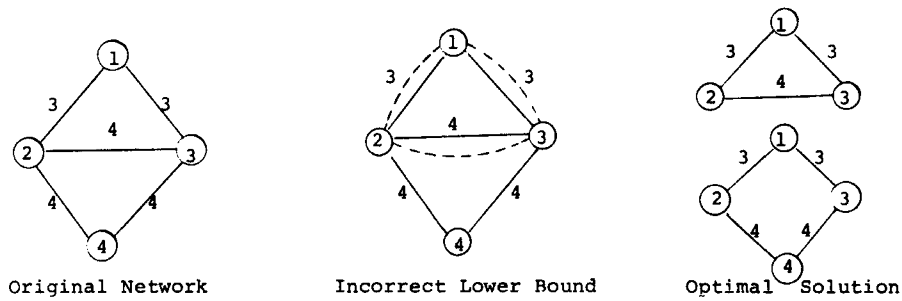

FIG. 3. A counterexample to Christofides' lower bound.

consider a particular matching problem derived from the original CARP. Let $\tilde{R}$ be the set of arcs in the CARP network that have nonzero demand. Define $S = \{s_1, s_2, \ldots, s_t\}$ to be the set of nodes in the original CARP network that have an odd number of arcs from $\tilde{R}$ incident to them. We again let $M = \lceil \Sigma_i \Sigma_j q_{ij}/W \rceil$ be the minimal number of cycles. Assume that I, the degree of the depot node, is even and no greater than $2M$ (if I is odd, set $I \leftarrow I + 1$ and proceed). Thus, the degree of the depot node must be increased by at least $2M - I$. Define two new sets of nodes $A = \{a_1, a_2, \ldots, a_{2M-I}\}$ and $B = \{b_1, b_2, \ldots, b_{2M-I}\}$, and let N_{SAB} be the network with node set $S \cup A \cup B$. Arc costs are defined as follows

$$d_{lm} = \begin{cases} \text{the shortest path distance between nodes } l \text{ and } m \\ \text{from the original CARP network} & \text{for } l, m \in S, \\ \text{the shortest path* distance between } l \text{ and the} \\ \text{depot node from the original CARP network} & \text{for } l \in S, m \in A, \\ \text{the length of the shortest arc that leaves the} \\ \text{depot node} & \text{for } l = a_i, m = b_i, \\ 0 & \text{for } l, m \in B. \end{cases}$$

Now consider a minimum cost 1-matching problem over N_{SAB}. The motivation for solving this problem is summarized in the following theorem:

Theorem 2. Let Z_{SAB} be the optimal solution value of the minimum cost 1-matching problem over N_{SAB}. Then Z_{SAB} + (sum of the lengths of all required arcs in the original CARP network) is a lower bound on the optimal solution value of the CARP.

Proof: Consider an optimal solution to the CARP. The set of arcs traversed consists of the set of required arcs $\tilde{R}$ and the set of auxiliary arcs T that have been added to the CARP network. We prove the theorem by showing that the optimal matching problem value is a lower bound to the total length of the arcs in T. The proof is given

*A path, in this definition, is presumed to consist of at least one arc.

for the case where the depot node in the original CARP network has even degree. The extension to the odd degree case is straightforward and will not be discussed here.

We consider a network with node set N, the set of nodes in the original CARP problem, arc set T, the set of arcs as defined above, and arc lengths taken from the original CARP problem. Initially, regard all nodes in S (the set of odd degree nodes relative to $\tilde{R}$) as *unmarked*. Note that since the set of arcs $\tilde{R} \cup T$ constitutes an Euler tour, the nodes which have odd degree with respect to $\tilde{R}$ will also have odd degree with respect to T.

Now remove arcs from this network according to the following steps:

Step 1: Start at an unmarked node n_i and follow a path until another unmarked node n_j or the depot node is reached, whichever comes first. In the matching problem, associate node n_i with node n_j or node a_k. If there are no nodes a_k available in N_{SAB}, continue the path until an unmarked node n_j is found. Remove every arc in the path selected, from the network. Mark all previously unmarked nodes that were involved as path endpoints.

Since $d_{n_i n_j}$ and $d_{n_i a_k}$ represent the shortest distance between n_i and n_j and n_i and 1, respectively, the length of the path removed is at least equal to the length of the matching over N_{SAB}.

Continue until all unmarked nodes are marked (or equivalently, until all nodes from S are matched with another node in the matching problem).

Step 2: Check to see if Step 1 reduced the degree of the depot node by at least $2M - I$ (or equivalently, if all nodes a_i have been matched with another node). If not, then find and remove a cycle which starts at the depot node. In the matching problem, for some i and j match node a_i to b_i and node a_j to b_j. By definition of $d_{a_i b_i}$, the cycle that was removed has length at least $d_{a_i b_i} + d_{a_j b_j} = 2\, d_{a_i b_i}$.

Continue until all nodes a_i have been matched with another node in the matching problem.

We remark that the paths described above can always be found. During the execution of Step 1, only the unmarked nodes and possibly the depot node have odd degree. So there must always be a path of arcs between an unmarked node and another unmarked node or the depot node.

During Step 2, since all nodes have been marked and since we assumed the depot node originally had even degree, every node has even degree. So if the depot node still has arcs incident to it (or equivalently, if an a_i remains unmatched), a cycle starting at the depot must exist.

For the matching problem, Steps 1 and 2 associate every odd degree node s_i and every node a_i with some other node. There may be nodes b_i and b_j which have not been matched. For these nodes, just match b_i and b_j. The matching problem objective value is not altered since $d_{b_i b_j} = 0$. Since every node in N_{SAB} is matched to another node, we finally have a feasible solution to the 1-matching problem. Steps 1 and 2 show that a subset of the arcs in T (those arcs removed by the two steps) have total length at least equal to the length of a feasible solution to the matching problem. Therefore, the optimal matching problem objective value is a lower bound on the total length of the arcs in T.

Theorem 2 has an interesting interpretation with respect to the CARP formulation. The 1-matching problem solution and the required arcs in the original CARP network form an Euler cycle. The only CARP constraint that this solution may violate is the vehicle capacity constraint. So the resulting lower bound can be viewed as a solution to the CARP problem with the capacity constraint relaxed. We remark that the valid lower bound for the CARP in Figure 3 is 24.

The lower bounding procedure of Theorem 2 can also be applied to special cases of the CARP in order to derive bounds for these well-known routing problems. Recall that the vehicle routing problem (VRP) is a particular case of the CARP with all the required arcs having length equal to zero. Now consider an analogous 1-matching problem over the network N_{SAB} defined for the VRP. We can immediately infer:

Corollary 2. Let Z_{SAB} be the optimal solution value of the minimum cost 1-matching problem over N_{SAB}. Then, Z_{SAB} is a lower bound on the optimal solution value of the VRP.

Since so many researchers have sought to obtain approximately optimal solutions to the VRP, Corollary 2 may prove useful in evaluating the accuracy of heuristic procedures for this problem.

Results analogous to Corollary 2 can also be derived for the other special cases of the CARP described in this paper.

V. AN ALGORITHM FOR THE CARP

A heuristic algorithm for the CCPP has been suggested by Christofides [3]. This procedure has some attractive features and has performed well on a sample of test problems. In this section, we introduce another class of heuristics for solving the CCPP and the more general CARP. We prefer the algorithm whose steps are outlined below due to its simplicity and logical appeal.

Step 1: Initialize—All demand arcs are serviced by a separate cycle.

Step 2: Augment—Starting with largest cycle available, see if a demand arc on a smaller cycle can be serviced on the larger cycle.

Step 3: Merge—Subject to capacity constraints, evaluate the merging of any two cycles (possibly subject to additional restrictions). Merge the two cycles which yield the largest positive savings.

Step 4: Iterate—Repeat Step 3 until finished.

This procedure is intuitively reasonable. In Step 2, we take advantage of the observation that since demand arcs "far away" from the depot need to be serviced anyway, it makes sense to plan for those trips to be as profitable as possible. We can afford, in some sense, to be less cautious in servicing the demand arcs "close" to the depot. Steps 3 and 4 are closely related to the well-known Clarke-Wright vehicle routing algorithm [5].

We have purposely been somewhat vague in describing the details of the algorithm (in particular, the Merge step) due to the fact that it is not yet clear what the best implementation strategy is. Much experimental testing remains to be done.

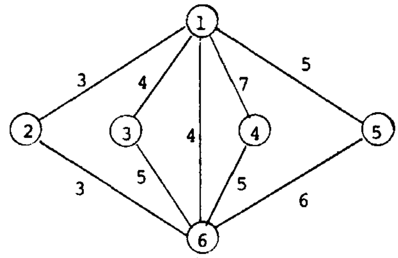

FIG. 4. Illustration of algorithm.

We do demonstrate, however, an implementation of the algorithm on the small example given in Figure 4. Here, vehicle capacity is 4 and each arc has unit demand. Initially, there are nine cycles–one for each demand arc. The augmentation step leaves the four cycles 1 2 6 1, 1 3 6 1, 1 6 4 1, 1 6 5 1. Step 3 merges 1 2 6 1 and 1 6 5 1 to obtain 1 2 6 5 1 reducing the number of cycles to three and we are finished. Note that merging 1 2 6 1 and 1 3 6 1 to obtain 1 2 6 3 1 or 1 3 6 1 and 1 6 5 1 to obtain 1 3 6 5 1 would have given the same total distance.

Two implementations of this heuristic have been coded and we expect to discuss these and other implementations in further detail in a sequel paper, after an appropriate set of test problems has been developed.

At this point, we would like to display some preliminary results. We tested the two implementations on seven CCPP's each of which had between 11 and 13 nodes and between 19 and 26 arcs. Let IMP1 and IMP2 denote the heuristic solutions obtained using the first and second implementations, respectively. Also, let LB denote the lower bound described in Theorem 2. Table I indicates that, at least for the CCPP, the proposed heuristic looks quite promising. The heuristic never yields a solution that deviates from the lower bound by more than about 18%.

Note added in proof. The lower bound result may be improved if the following two minor changes are made:

(i) Define I, the incidence of the depot node, with respect to the set of demand arcs $\tilde{R}$.

(ii) Let arc cost d_{lm} equal the length of the shortest path from the depot node to an end node of a demand arc for $l = a_i, m = b_i$.

TABLE I Computational results.

Problem	IMP1	IMP2	LB	IMP1/LB	IMP2/LB
1	351	349	310	1.13	1.13
2	394	394	339	1.16	1.16
3	316	316	275	1.15	1.15
4	316	316	274	1.15	1.15
5	429	383	370	1.16	1.04
6	340	348	295	1.15	1.18
7	325	325	312	1.04	1.04

The effect of these changes is evident if one considers a CARP in which the depot is not adjacent to a demand arc. The proof of this extension is easily handled within the framework of the existing proof of Theorem 2. We thank Professor Edward Baker for bringing this to our attention.

The authors would like to thank Professor Michael Ball for helpful comments. In particular, his suggestions resulted in a more streamlined proof of Theorem 2 than the earlier version. In addition, we thank Jim DeArmon for computer programming assistance.

References

[1] E. Beltrami and L. Bodin, "Networks and Vehicle Routing for Municipal Waste Collection," *Networks*, **4**, 65–94 (1974).

[2] L. Bodin, "A Taxonomic Structure for Vehicle Routing and Scheduling Problems," *Comput. Urban Soc.*, **1**, 11–29 (1975).

[3] N. Christofides, "The Optimum Traversal of a Graph," *Omega*, **1**, 719–732 (1973).

[4] N. Christofides, "Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem," Technical Report, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA (1976).

[5] G. Clarke and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Oper. Res.*, **12**, 568–581 (1964).

[6] M. Garey and D. Johnson, *Computers and Intractability*, Freeman, San Francisco, 1979.

[7] B. Golden, "Recent Developments in Vehicle Routing," in *Computers and Mathematical Programming*, W. White, Ed., National Bureau of Standards Special Publication 502, Washington, D.C., (1978) 233–240.

[8] B. Golden, T. Magnanti, and H. Nguyen, "Implementing Vehicle Routing Algorithms," *Networks*, **7**, 113–148 (1977).

[9] R. Karp, "Reducibility Among Combinatorial Problems," *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., Plenum, New York, (1972) 85–104.

[10] R. Karp, "On the Computational Complexity of Combinatorial Problems," *Networks*, **5**, 45–68 (1975).

[11] J. Lenstra and A. Rinnooy Kan, "On General Routing Problems," *Networks*, **6**, 273–280 (1976).

[12] C. Orloff, "A Fundamental Problem in Vehicle Routing," *Networks*, **4**, 147–162 (1974).

[13] C. Orloff, "On General Routing Problems: Comments," *Networks*, **6**, 281–284 (1976).

[14] R. Russell, "An Effective Heuristic for the *M*-Tour Traveling Salesman Problem with Some Side Conditions," *Oper. Res.*, **25**, 517–524 (1977).

[15] S. Sahni, "General Techniques for Combinatorial Approximation," *Oper. Res.*, **25**, 920–936 (1977).

[16] S. Sahni and T. Gonzalez, "*P*-Complete Approximation Problems," *J. ACM*, **23**, 555–565 (1976).

[17] H. Stern and M. Dror, "Routing Electric Meter Readers," *Comput. Oper. Res.*, **6**, 209–223 (1979).

Received October 10, 1979
Accepted September 19, 1980