ARTICLES

OPTIMAL SOLUTION OF VEHICLE ROUTING PROBLEMS USING MINIMUM K-TREES

MARSHALL L. FISHER

University of Pennsylvania, Philadelphia, Pennsylvania

(Received February 1990; revisions received November 1991, October 1993; accepted November 1993)

We consider the problem of optimally scheduling a fleet of K vehicles to make deliveries to n customers subject to vehicle capacity constraints. Given a graph with $n + 1$ nodes, a K-tree is defined to be a set of $n + K$ edges that span the graph. We show that the vehicle routing problem can be modeled as the problem of finding a minimum cost K-tree with two K edges incident on the depot and subject to some side constraints that impose vehicle capacity and the requirement that each customer be visited exactly once. The side constraints are dualized to obtain a Lagrangian problem that provides lower bounds in a branch-and-bound algorithm. This algorithm has produced proven optimal solutions for a number of difficult problems, including a well-known problem with 100 customers and several real problems with 25–71 customers.

Vehicle routing and scheduling concerns a rich class of problems involving the optimal use of a fleet of vehicles to transport finished goods or manufacturing supplies. This paper considers a version of the vehicle routing problem defined by the following parameters.

K = the number of vehicles;
n = the number of customers to which a delivery must be made; customers are indexed from 1 to n and index 0 denotes the central depot;
b = the capacity (e.g., weight or volume) of vehicle k;
a_i = the size of the delivery to customer i ($a_0 = 0$);
c_{ij} = the cost of direct travel between points i and j ($c_{ij} = c_{ji}$ for all ij).

We are required to assign each customer to a vehicle and to establish the delivery sequence for the customers assigned to each vehicle to minimize total travel cost subject to vehicle capacity constraints. Routes with single customers are not allowed.

Many case studies have reported the successful implementation of algorithms for this problem or variations on it. These real implementations seem to have relied exclusively on heuristics. Although optimization has not been considered a practical approach for real problems in the past, there are now many reasons for changing this viewpoint. Rapidly decreasing computation costs are pushing the tradeoff between computation time and solution quality in the direction of higher quality solutions. The accuracy of data on the cost of travel between customers has been improved greatly by the creation of road network data bases. Finally, a growing body of research on related models like the traveling salesman problem provides a base of theoretical results on which to draw for vehicle routing optimization. Even if optimization algorithms are not run to full optimality, they offer the possibility of better solutions than existing heuristics can provide and increased robustness because they provide a bound on the amount by which a particular solution differs from optimality.

This paper develops an approach to vehicle routing optimization that draws on the ideas in Held and Karp's (1971) successful study of the traveling salesman problem. Given a graph with $n + 1$ nodes, we define a K-tree to be a set of $n + K$ edges that span the graph. We show that the vehicle routing problem can be modeled as the problem of finding a minimum K-tree with degree 2K on the depot, together with side constraints that impose vehicle capacity and the requirement that the degree at each customer must be 2. The side constraints are dualized to obtain a

Subject classifications: Programming, integer, algorithms, relaxation/subgradient: k-tree based, transportation, vehicle routing; optimization algorithm.
Area of review: DISTRIBUTION, TRANSPORTATION AND LOGISTICS.

0030-364X/94/4204-0626 $01.25

Lagrangian relaxation that is a minimum degree-constrained K-tree problem. We have developed a polynomial algorithm for this problem (reported in Fisher 1994) that generalizes the results of Glover and Klingman (1974a) for the degree-constrained minimum spanning tree problem. We also use a novel branch-and-bound procedure in which the problem is partitioned by fixing the edge incidence of selected subsets of clustered customers. The K-tree approach to vehicle routing was first described in Fisher (1988).

To evaluate the hypothesis that optimization can be a practical tool, we have tested our algorithm on a sample of problems that includes a number of real problems and much larger problems than have previously been attempted. Our test problems include six well-known problems with 50–199 customers that are reported in Christofides and Eilon (1969) and Christofides, Mingozzi and Toth (1979), and six real problems with 25–134 customers supplied by Air Products and Chemicals, National Grocers Limited, and Exxon. Customer locations in the six problems taken from the literature were all generated randomly. In five of these problems, customer locations were chosen from a uniform distribution. The other problem, reported in Christofides, Mingozzi and Toth (1979), had 100 customers and was designed to be "realistic" by generating customer locations in clusters. We have solved to optimality this 100-customer realistic problem and five of the six real problems. The remaining real problem with 134 customers was solved to within 1.04% of optimality. Solution times ranged from 6–860 minutes. Calculations were performed on a somewhat outdated workstation, an Apollo Domain 3000. A contemporary version of this workstation sells for about \$10,000, and is about 15 times faster. Thus, one can optimize these problems in a matter of minutes on a computer selling for a fraction of the cost of a single truck. We also describe how the K-tree approach can be extended to accommodate realistic variations, such as asymmetric costs, time windows, and nonuniform fleets. It thus seems that practical optimization of complex, real problems of moderate size can now be contemplated.

It is easy to see that our prohibition of single customer routes is not constraining in many cases. Note that customer j cannot appear alone on a route if $(\sum_{i=1}^{n} a_i) - a_j > (K - 1)b$, because using a dedicated vehicle to deliver customer j would leave insufficient vehicle capacity to service the remaining customers. Rearranging terms in this expression, we can see that our prohibition of single customer routes is not constraining if $a_j/b < \sum_{i=1}^{n} a_i/b - (K - 1)$ for all j, a condition that will be satisfied if vehicle capacity constraints are sufficiently tight and no customer is large relative to vehicle capacity. This condition was satisfied for 8 out of the 12 test problems used here (problems 1, 2, 3, 5, 7, 8, 11, 12). For two other problems (problems 9 and 10), a few customers violated this condition, but the resulting single route solutions were easy to show to be nonoptimal using the lower bounding procedure presented here and a negligible amount of computation time. While it has not been established formally that the optimal solutions to problems 4 and 6 do not contain single customer routes, it seems unlikely because deleting the largest customer and a single vehicle would leave a very tightly constrained problem. Our development could be modified to allow for particular customers to be served on single stop routes by including two edges between these customers and the depot in the graph used to compute lower bounds, although, of course, this could affect lower bound strength.

General research on vehicle routing is surveyed in Bodin et al. (1983), Christofides (1985), and Fisher (1992) and on optimization algorithms, in particular, in Laporte and Nobert (1987). Some theoretical results related to optimization have been developed. Araque (1989) provided the polyhedral description of the unit demand vehicle routing polytope, and Cornuejols and Harche (1989) extended valid traveling salesmen inequalities for vehicle routing. Specific optimization algorithms have been given by Christofides and Eilon (1969), Lucena (1986), Fisher and Jaikumar (1978), Christofides, Mingozzi and Toth (1981a, b), Laporte, Nobert and Desrochers (1985) and Cornuejols and Harche (1989). Computational results reported in these papers were based on randomly generated test problems, and the largest problems solved to optimality had about 50 customers.

Christofides, Mingozzi and Toth (1981a) use spanning trees to obtain lower bounds, but in a different way from the approach described in this paper. For a specified value k, $K \leqslant k \leqslant 2K$ (details on the appropriate choice of k are provided in Christofides, Mingozzi and Toth 1981a) they find a spanning tree of the customers and depot with degree k on the depot (which they call a k-degree center tree) and add K least-cost edges, $2K - k$ incident on the depot and $k - K$ not incident on the depot. Lagrangian penalties on the customer degree constraints are used to tighten this bound. The K least-cost edges added need not be distinct from those in the spanning tree, i.e., a given edge can be used twice, once in the spanning tree and once in the K additional edges, which makes analysis of the lower bounding problem simple. Those duplicated edges incident on the depot allow for the possibility of single customer routes. Vehicle capacity constraints are not used, except in a minor way to regulate the choice of k.

The lower bounds from this procedure appear to be quite weak. For the 10 problems with 10–25 customers solved in Christofides, Mingozzi and Toth (1981a), this lower bound was 85% of the optimum on average. By contrast, the results reported here in Table II for the K-tree relaxation applied to problems with 25–199 customers show lower bounds of 98% of the optimum on average. The weaker bounds could arise from limited use of vehicle capacity constraints, the possibility of choosing edges twice, and the preclusion here of single customer routes.

There is also a growing literature on a class of capacitated spanning tree problems that arise in telecommunications and share some properties with the vehicle routing problem. Gavish (1982, 1985) has developed a Lagrangian algorithm, Bousba and Wolsey (1989) achieved good computational results for a general model, and Araque, Hall and Magnanti (1990) discuss the polyhedral structure of capacitated tree and routing problems. Gavish's approach is closest to ours, except that he works with spanning trees instead of K-trees.

Section 1 of this paper defines the Lagrangian relaxation. We also identify a special case of the vehicle routing problem for which the Lagrangian bound equals the optimal value. Section 2 presents heuristics for generating vehicle capacity constraints, and Section 3 gives two primal heuristics for obtaining feasible solutions using information from Lagrangian solutions. In Section 4 we describe the problems used in computational testing and report lower bounds obtained with the K-tree algorithm. Section 5 describes our branch-and-bound procedure and computational results for problems solved to optimality. Section 6 extends these ideas to incorporate a number of complications found in real problems.

Data for new test problems and improved solutions are provided in an Appendix. One of our findings is that algorithms perform differently on real versus uniform random problems, so it is important to test exact and approximation algorithms on real data. The new test cases provided here should contribute to that goal.

1. FORMULATION AND LAGRANGIAN RELAXATION

We model the problem considered, which we denote **VRP**, as a minimum cost, degree-constrained K-tree problem with side constraints. The side constraints are then dualized to obtain a Lagrangian relaxation.

Let $N = \{1, \ldots, n\}$, $N_0 = N \cup \{0\}$ and x_{ij} denote a $0-1$ variable equal to 1 if edge (i, j) is selected in a solution. Since the edge (i, j) is undirected, it will simplify notation to adopt the convention that the subscripts ij on x_{ij} are an unordered pair, so x_{ij} and x_{ji} denote the same variable. Then x_{ij} is defined for the $n(n+1)/2$ unordered pairs in $N_0 \times N_0$. Also define

$$x = (x_{01}, x_{02}, \ldots, x_{0n}, x_{12}, \ldots, x_{n-1,n});$$

$$X = \left\{ x | x = 0-1 \text{ and defines a K-tree satisfying } \cdot \sum_{i=1}^{n} x_{0i} = 2K \right\}.$$

For $S \subseteq N$, let $\bar{S} = N_0 - S$, $a(S) = \sum_{i \in S} a_i$ and $r(S) = \lceil a(S)/b \rceil$, where $\lceil y \rceil$ denotes the smallest integer not less than y. For $S \subseteq N_0$, let $E(S)$ denote the edge set of a complete, undirected graph on the node set S, i.e., $E(S)$ is the set of all unordered pairs ij, $i \in S$, $j \in S$, $i \neq j$. Then the **VRP** can be formulated as

$$Z^* = \min_{x \in X} \sum_{ij \in E(N_0)} c_{ij} x_{ij} \tag{1}$$

$$\sum_{\substack{j \in N_0 \\ j \neq i}} x_{ij} = 2 \quad \text{for all } i \in N \tag{2}$$

$$\sum_{i \in S} \sum_{j \in \bar{S}} x_{ij} \geq 2r(S) \quad \text{for all } S \subseteq N \text{ with } |S| \geq 2. \tag{3}$$

In this formulation, routes with single customers are not allowed.

Theorem 1. *A solution $x \in X$ is feasible in* **VRP** *if and only if it satisfies* (2)–(3).

Proof. First note that an $x \in X$ satisfying (2) must correspond to K cycles that begin and end at the depot. Index these cycles by $k = 1, \ldots, K$ and let S_k denote the set of customers on cycle k. We will establish the *if* case by showing that if $a(S_k) > b$, then constraint (3) with $S = S_k$ is violated. This follows because $\sum_{i \in S_k} \sum_{j \in \bar{S}_k} x_{ij} = 2$ while $r(S_k) = \lceil a(S_k)/b \rceil > 1$.

To establish the only-if case, first note that any feasible solution to **VRP** must consist of K cycles that begin and end at the depot and hence must satisfy $x \in X$ and (2). To see that (3) must also hold, note that the vehicle capacity constraint implies that at least $r(S)$ vehicles are required to service the demand in set S. Since each vehicle must enter and leave set S, we must have at least $2r(S)$ edges between sets S and $\bar{S}$, exactly as constraint (3) requires.

Letting u_i, $i \in N$ and $v_S \geq 0$ for $S \subseteq N$, $|S| \geq 2$ denote Lagrange multipliers for (2) and (3), we can define the Lagrangian relaxation of (1)–(3) as:

$$Z_D(u, v) = \min_{x \in X} \sum_{ij \in E(N_0)} \bar{c}_{ij} x_{ij} + 2 \sum_{i=1}^{n} u_i + 2 \sum_{S \subseteq N} v_S r(S), \tag{4}$$

where $u_0 = 0$, and

$$\bar{c}_{ij} = c_{ij} - u_i - u_j - \sum_{\substack{S \text{ such that} \\ i \in S, j \in \bar{S} \\ \text{or} \\ i \in \bar{S}, j \in S}} v_S.$$

It is well known and easy to show that $Z_D(u, v) \leq Z^*$. A polynomial algorithm for (4) is given in Fisher (1994). We use the subgradient method to approximate an optimal solution to $\max_{\substack{u \\ v \geq 0}} Z_D(u, v)$. Since there are $0(2^n)$ constraints in the set (3), it is not feasible to tabulate explicitly all these constraints prior to computation. Rather, we will develop procedures described in Section 4 to generate a subset of these constraints dynamically as they are violated. All other constraints have $v_s = 0$ and are ignored.

The quantity $r(S) = \lceil a(S)/b \rceil$ in the right-hand side of (3) is a lower bound on the number of vehicles required to service the customers in the set S. As noted by Laporte and Nobert, (3) can be tightened in some cases by replacing $r(S)$ by the value of a bin packing solution for the customer in S or by a lower bound on the bin packing solution, such as in Martello and Toth (1990). We have not implemented this refinement because in the data we were using a_i/b was small for most customers, so it seemed likely that $\lceil a(S)/b \rceil$ would equal the bin packing values of S in most cases.

Constraints (3) are analogous to the subtour elimination constraints in a well-known formulation of the traveling salesman problem. Nobert (1982) and Laporte, Nobert and Desrochers (1985) showed that vehicle capacity also can be imposed by the generalized subtour elimination constraints:

$$\sum_{ij \in E(S)} x_{ij} \leq |S| - r(S) \quad \text{all } S \subseteq N \text{ with } |S| \geq 2. \tag{3'}$$

It is easy to show that (2) and (3) are equivalent to (2) and (3′), so it might seem inconsequential whether we work with (3) or (3′). However, the choice is important computationally because it is impractical to generate more than a small fraction of the $0(2^n)$ possible constraints in (3) or (3′). Hence, one must work with a modest number of $m << 2^n$ constraints. Informal computational comparisons have shown that the best m constraints from (3) produce a tighter bound than the best m constraints from (3′).

Constraints (3) also lend themselves to tightening in the following way. The rationale underlying (3) is that $br(S)$ defines the minimum vehicle capacity that must enter and leave S to feasibly carry the customers in set S. However, if $x_{ij} = 1$ in the left-hand side of (3) for some customer $j \in \bar{S}$, then customer j is serviced by the same vehicle as at least one customer in set S and its demand subtracts from the capacity available for set S. Hence, the total vehicle capacity entering and leaving set S must be at least $a(S)$ plus the customer demand for all customers $j \in \bar{S}$ with $x_{ij} = 1$. This implies

$$\sum_{i \in S} \sum_{j \in \bar{S}} x_{ij} \geq 2 \left\lceil \frac{a(S) + \sum_{\substack{i \in S \\ j \in \bar{S}}} a_j x_{ij}}{b} \right\rceil. \tag{5}$$

Constraint (5) is not directly useful, because the right-hand side is a nonlinear function of x, but it can be used to derive other constraints. An example is given below.

For any $S \subseteq N$, let

$$S' = \{j \in \bar{S} | j \geq 1 \text{ and } a_j > br(S) - a(S)\}$$

$$e_j = \begin{cases} 0, & j \in S \\ 0, & j \in S' \text{ and } |S'| \leq 2 \\ \dfrac{r(S)}{r(S) + 1}, & j \in S' \text{ and } |S'| > 2 \\ 1, & j \in \bar{S} - S' \end{cases}$$

We can now define the tightened vehicle capacity constraints

$$\sum_{j=0}^{n} e_j \sum_{i \in S} x_{ij} \geq 2r(S) \quad \text{for all } S \subseteq N \text{ with } |S| \geq 2. \tag{6}$$

Theorem 2. *A solution $x \in X$ is feasible in* **VRP** *if and only if it satisfies* (2) *and* (6).

Proof. Proof of the if case is obvious because (6) is the same constraint as (3) except that some coefficients may be smaller in (6). Hence, $x \in X$ that satisfies (2) and (6) also satisfies (2) and (3) and is feasible in **VRP** by Theorem 1.

Now we will consider the only-if case and show that a given $x \in X$ satisfying (2) and (3) also satisfies (6). For a given $S \subseteq N$, first suppose that $|S'| = 0$ or $x_{ij} = 0$ for all $i \in S$, $j \in S'$. Then (3) and (6) are the same constraint, so (6) obviously holds. Alternatively, suppose that $|S'| \geq 1$ and $x_{kj^*} = 1$ for some

$j^* \in S'$ and $k \in S$. We will consider two cases to show that (6) holds.

Case 1. ($|S'| > 2$) At least $r(S) + 1$ vehicles are required to carry the customers in $S \cup \{j^*\}$ which implies that $2r(S) + 2$ edges are incident on $S \cup \{j^*\}$. The fact that $x_{kj^*} = 1$ and $\sum_{i \in N,\, i \neq j^*} x_{ij^*} = 2$ implies that $2r(S) + 2$ edges are incident on the set S so $\sum_{j \in \bar{S}} \sum_{i \in S} x_{ij} \geq 2r(S) + 2$.

Multiplying both sides of this constraint by $r(S)/(r(S)+1)$gives

$$\sum_{j \in \bar{S}} \frac{r(S)}{r(S) + 1} \sum_{i \in S} x_{ij} \geq 2r(S)$$

which implies (6).

Case 2. ($|S'| \leq 2$) Let $S'' = \{j \in S' | \sum_{i \in S} x_{ij} > 0\}$. If $|S''| = 1$, then a single customer $j*$ from S' and one or more customers from S are on the same route. Because $a_{j*} > br(S) - a(S)$, the remaining customers in S require at least $r(S)$ vehicles. Hence, at least $2r(S)$ edges are incident on S from $\bar{S} - S'$. This implies $\sum_{j \in \bar{S} - S'} \sum_{i \in S} x_{ij} \geq 2r(S)$, which is equivalent to (6) in this case.

Now suppose that $|S''| = 2$, so two customers in S' are linked to customers in S. If these two customers are on the same route, then the remaining customers in S require at least $r(S)$ vehicles, so at least $2r(S)$ edges are incident on S from $\bar{S} - S'$. Hence (6) is satisfied.

If the customers in S' are on two different routes, then these two routes contribute at least two edges incident on S from $\bar{S} - S'$. The remaining customers in S require at least $r(S) - 1$ vehicles contributing $2r(S) - 2$ additional edges incident on S from $\bar{S} - S'$. Hence (6) is satisfied.

Constraint (6) clearly dominates (3). We have also found computationally that the constraints produce much tighter bounds in certain cases. We can also obtain a valid linear inequality by removing the upper brackets from the right-hand side of (5) to obtain $2(a(S) + \sum_{j \in \bar{S}}^{i \in S} a_j x_{ij})/b$. While this constraint is generally not tighter than (3), it might be useful in some cases.

A constraint in set (3) or (6) need not be a facet. For example, consider the case where $S = \{1, 2, 3\}$, $a_1 = a_2 = a_3 = 3$, $a_j = 1$, $j \geq 4$ and $b = 5$. Constraints (3) and (6) are the same for this example. Both have a right-hand side of 4 when the constraint is clearly valid with a right-hand side of 6.

Although we have not established the worst-case behavior of the lower bound Z_D, we have considered a potentially troublesome special case and have shown that $Z_D = Z^*$ for this case. Consider the case in which $b = 4$, $n = 4K$, $a_j = 1$ for all j, $2K$ customers are located at the depot and $2K$ are located 1 unit from the depot.

Then $Z^* = K$, while there is a minimum K-tree with degree $2K$ on node 0 that has a cost $Z_K = 1$. Since $Z^*/Z_K \to \infty$ as $K \to \infty$, one might expect that this case could be problematic for K-tree-based bounds. In fact, it is easy to show that $Z_D = Z^*$ for a broader class of problems that includes this case.

Consider a vehicle routing problem in which customer i is distance d_i from the depot ($d_0 = 0$), $c_{ij} = |d_i - d_j|$ for all ij, $a_j = 1$ for all j, and $K = \lceil n/b \rceil$. For notational simplicity, assume that n/b is integral and $d_1 \leq d_2 \leq \ldots \leq d_{n-1} \leq d_n$. A feasible solution to this problem is obtained by assigning customers $1, 2, \ldots, b$ to vehicle 1, customers $b + 1, b + 2, \ldots, 2b$ to vehicle $2, \ldots,$ and customers $n - b + 1, n - b + 2, \ldots, n$ to vehicle K. The objective value for this solution is $Z = 2(d_b + d_{2b} \ldots + d_{n-b} + d_n)$.

It is easy to show that this solution is optimal and $Z_D = Z = Z^*$ by exhibiting values u^* and v^* for the Lagrange multipliers in the K-tree relaxation such that $Z_D(u^*, v^*) = Z$. The required values of u^* and v^* are defined as follows. Set $u_i^* = 0$, $i = 1, \ldots, n - 1$ and $u_n^* = d_n - d_{n-1}$. The variables $v_S^* = 0$ for all S except $n - 1$ sets defined as $S_k = \{k, \ldots, n\}$, $k = 1, \ldots, n - 1$. For these sets, define $v_{S_k} = d_k - d_{k-1}$, $k = 1, \ldots, n - 1$.

Straightforward substitution of these values for u^* and v^* into the definition of $\bar{c}_{ij}$ reveals that $\bar{c}_{ij} = 0$ for all ij. The fact that $\bar{c}_{ij} = 0$ for all ij implies that the value of a minimum K-tree is also 0, so

$$\begin{aligned} Z_D(u^*, v^*) &= 2 \sum_{i=1}^{n} u_i^* + 2 \sum_{S \subseteq N} v_S^* r(S) \\ &= 2(d_n - d_{n-1}) \\ &\quad + 2 \sum_{k=1}^{n-1} \left\lceil \frac{n + 1 - k}{b} \right\rceil (d_k - d_{k-1}) \\ &= 2 \sum_{k=1}^{n} \left(\left\lceil \frac{n + 1 - k}{b} \right\rceil - \left\lceil \frac{n - k}{b} \right\rceil \right) d_k \\ &= Z. \end{aligned}$$

It is clear that for this example only a subset of (3) and $0 \leq x \leq 1$ is needed to define the optimal value.

2. CONSTRAINT GENERATION

Lower bounds are obtained by applying the subgradient method to $\max_{v \geq 0}^{u} Z_D(u, v)$ with $Z_D(u, v)$ determined by (4) in Section 1. Since the number of vehicle capacity constraints (6) is vast, we have developed heuristics to identify a small subset of this enormous family to use explicitly in our computations. Our heuristics include procedures for choosing an initial subset of capacity constraints and for dynamically adding and deleting constraints during subgradient computations.

Our procedure for initializing the constraint set is motivated by the result at the end of Section 1 in which we found for a special case that optimal constraints corresponded to sets nested around the customer farthest from the depot. To mimic this behavior, we define m seed customers $s_1, \ldots, s_m$ and construct nested sets around them that define the initial constraints. In our computational work, $m = K + 3$ and seeds $s_1, \ldots, s_K$ were chosen to be the customers farthest from the depot on the routes of a given initial feasible solution. The remaining three seeds were chosen sequentially, with each picked to be maximally distant from existing seeds and the depot. Specifically, letting S_D denote the set of seeds chosen thus far, seed s_i is picked to $\max_{s_i \notin S_D} \min\{c_{0s_i}, \min_{j \in S_D} c_{js_i}\}$.

Let $i_1, \ldots, i_{n-1}$ be an indexing of the customers in $N - \{s_i\}$ in increasing order of proximity to seed s_i. Then the initial constraints increasing for seed s_i correspond to the sets

$\{s_i, i_1\}$
$\{s_i, i_2\}$
$\{s_i, i_1, i_2\}$
$\{s_i, i_1, i_2, i_3\}$
$\{s_i, i_1, i_2, i_4\}$
$\{s_i, i_1, i_2, i_3, i_4\}$
$\vdots$

In our computational work, the number of sets for each seed was limited to 60.

After some number of iterations (50 in our computational work), we begin adding constraints to the explicit set as they are violated by the Lagrangian solution. To find violated constraints, we delete all edges incident on the depot from the graph corresponding to the Lagrangian solution and then check the customers in each connected component for violation of constraint (3).

We also delete constraints from the active set if v_S has been 0 for some number of consecutive iterations (3 in our computations). Constraint deletion was not begun until the 10th subgradient iteration.

3. LAGRANGIAN HEURISTICS

Feasible solutions are obtained at each subgradient iteration using three alternative heuristics based on the Lagrangian solution or on reduced costs $\bar{c}_{ij}$. None of these heuristics dominates on the others in performance. Each heuristic generates K partial routes. The final steps in each case are to optimally insert unscheduled customers as possible and apply the 3 − opt interchange heuristic (Lin and Kernighan 1973) to each route.

The first heuristic is executed on any iteration in which the Lagrangian solution graph contains exactly K connected components after deletion of edges incident on the depot. Let $C_1, \ldots, C_K$ denote the set of customers in each component. To obtain K feasible partial routes, for any k with $a(C_k) > b$ delete customers in the order of proximity to the depot until $a(C_k) \leq b$.

Our second heuristic was motivated by the observation that Lagrangian solutions were often nearly feasible once the dual variables were close to optimal. This suggested obtaining feasible solutions by applying the minimum K-tree algorithm to the Lagrangian problem with the added restriction that we not violate any vehicle routing constraints. In the resulting procedure, the first route is initialized by selecting a customer closest to the depot as measured by the reduced costs $\bar{c}_{ij}$. (All distances in this heuristic are measured with respect to the reduced costs $\bar{c}_{ij}$.) A second customer is chosen to be closest to the first. We continue to select customers for the first route, with each new customer chosen to be closest to the last selected, until adding further customers would exceed vehicle capacity, or until the depot is selected as the nearest customer. At this point, the first route is complete. (We do not allow selection of the depot if the load on the route is less than $\sum_{i \in N} a_i - (K-1)b$, because this would lead to infeasibility.) The remaining routes are formed according to the same procedure working always with unscheduled customers that remain.

The third Lagrangian heuristic attempts to use as many edges as possible from the Lagrangian solution. Let $\bar{x}_{ij}$ denote the Lagrangian solution at some iteration. Choose a customer i_1 farthest from the depot and select a shortest arc (i_1, i_2) satisfying $\bar{x}_{i_1 i_2} = 1$ and $a_{i_1} + a_{i_2} \leq b$. At any point in the algorithm, we will have a partial route $i_1, \ldots, i_k$ for a vehicle satisfying $\bar{x}_{i_1, i_2} = \bar{x}_{i_2 i_3} = \ldots = \bar{x}_{i_{k-1} i_k} = 1$. At this point, we choose a shortest edge joining some customer j to either i_1 or i_k, satisfying $\bar{x}_{i_1 j} = 1$ or $\bar{x}_{i_k j} = 1$ and $a(\{i_1, \ldots, i_k, j\}) \leq b$. When no edge can be found

Table I
Characteristics of the Test Problems

Problem	(1) n	(2) K	(3) $\frac{\sum_{i \in N} a_i}{Kb}$
1	50	5	0.97
2	75	10	0.97
3	100	8	0.91
4	150	12	0.93
5	199	16	0.999
6	100	10	0.91
7	25	3	0.92
8	29	4	0.94
9	36	4	0.76
10	44	4	0.90
11	71	4	0.96
12	134	7	0.95

that satisfies the required conditions, we select an unscheduled customer farthest from the depot and repeat the process for generating a partial vehicle route until K partial vehicle routes have been generated.

4. COMPUTATIONAL RESULTS

The procedures described in this paper were applied to 12 test problems. Characteristics of these problems are displayed in Table I including the number of customers (column 1), the number of vehicles (column 2) and the average vehicle utilization (column 3). Column 3 measures the tightness of the vehicle capacity constraints. All these problems are tightly constrained. Problem demand is 93% of vehicle capacity on average, and in each problem the fleet size could not be reduced without destroying feasibility. By contrast, some problems solved previously had relatively loose vehicle capacity constraints. For example, the problems in Laporte, Nobert and Desrochers have demand equal to about 66% of vehicle capacity on average.

All problems except 7–9 were planar; that is, customers are located at points in the plane, and c_{ij} is the distance between points i and j, computed as a single precision real value. In problems 7–9, integral values for all c_{ij} were defined as part of the input.

The first six problems are well known test cases taken from the literature. Customer locations for problems 1–3 were randomly generated from a uniform distribution. The data for these problems are given in Christofides and Eilon. Problem 4 was obtained by adding the customers of problems 1 and 3 with the depot and vehicle capacities as in problem 3. Problem 5 was obtained by adding the customers of problem 4 with the first 49 customers of problem 2. Data for problem 6 are given in Christofides, Mingozzi and Toth (1979). As described there, this problem was designed to resemble real problems by generating customer locations that are clustered. We used $K = \lceil \sum_{i=1}^{n} a_i/b \rceil$ for all problems.

These problems have been used in many previous studies to test the performance of various heuristics. These studies computed c_{ij} in different ways (some using real values, others various rounding schemes) and were not always clear on how cost values were being computed. In retrospect, lower cost solutions reported for these problems may have been due as much to how the cost data were rounded as to the inherent superiority of the heuristic used. In this study, all solutions and bounds for these problems are based on real c_{ij}s.

Although many researchers have applied heuristics to problems 1–6, there has been little effort to compute lower bounds to determine how close the heuristic values are to optimal. An exception is a recent analysis by Cornuejols and Harche who applied an LP-based branch-and-bound algorithm enhanced with a facet generation scheme to problem 1 with all c_{ij} rounded to integral values. They obtained an initial lower bound of 514 and then used branching to establish that 521 is the optimal value for this problem with integral costs.

The data in problems 7–12 are taken from real vehicle routing applications. Problems 7–9 are concerned with the delivery of industrial gases in cylinders and are based on data provided by Air Products and Chemicals, Inc. Problems 10 and 12 represent a day of grocery deliveries from the Peterboro and Bramalea, Ontario terminals, respectively, of National Grocers Limited (see Arrizza and Karellas 1983). Problem 11 is concerned with the delivery of tires, batteries and accessories to gasoline service stations and is based on data obtained from Exxon that they had developed for other studies.

These problems required some adaptation for our purposes, because the original problems had nonuniform fleets and time window constraints. Problems 7–12 use the original depot location, customer locations, and order sizes, but omit the time constraints and use a uniform vehicle capacity within the range of capacities spanned by the heterogeneous fleet. In all cases, omitting time constraints resulted in loose vehicle capacity constraints. To obtain a tightly constrained problem, we set $K = \lceil \sum_{i=1}^{n} a_i/b \rceil$.

Table II
Computational Results

Problem	(1) Upper Bounds: Best Previously Known Feasible Solution	(2) Upper Bounds: Optimal or Improved Solution Found in our Testing	(3) Lower Bound	(4) Lower/ Upper	(5) Spanning Tree Cost	(6) K-Tree Cost	(7) Degree-Constrained K-Tree Cost	(8) Vehicle Capacity Constraints Generated	(9) Sub-gradient Iterations	(10) Computation Time Apollo Domain 3000 (Minutes)
1	524.61[a]	—	507.09	0.97	376.49	415.51	445.41	611	3,000	95.75
2	835.26[a]	—	755.50	0.90	472.33	536.87	636.55	177	3,000	183.97
3	826.14[a]	—	785.86	0.95	562.26	587.84	673.25	410	3,000	307.95
4	1028.42[a]	—	932.68	0.91	635.28	669.55	805.96	249	3,000	682.41
5	1334.55[b]	—	1096.72	0.82	691.99	734.70	958.89	233	3,000	1186.00
6	819.56[a]	819.56[e]	817.77	0.998	417.30	444.51	635.94	211	2,000	259.64
7	3104[c]	3070[e]	3070	1.00	1,769	1,769	1,769	24	769	11.23
8	5830[c]	5829[d]	5829	1.00	2,313	2,313	3,255	25	2,444	53.33
9	5032[c]	4961[e]	4961	1.00	2,479	2,479	2,968	27	320	4.87
10	723.54[c]	723.54[e]	720.76	0.996	428.09	436.95	452.76	78	2,000	49.74
11	244.92[c]	241.97[f]	237.76	0.98	150.16	154.16	187.93	347	2,000	105.03
12	1216.66[c]	1163.60[e]	1133.73	0.97	615.03	620.97	688.97	500	2,000	253.84

Sources for Upper Bounds:
[a] Taillard (1992).
[b] Osman (1993).
[c] Algorithm in Bramel and Simchi-Levi (1992). These problems were run and the results communicated to me by Julien Bramel and David Simchi-Levi.
[d] First Lagrangian heuristic described in Section 5.
[e] Second Lagrangian heuristic described in Section 5.
[f] Third Lagrangian heuristic described in Section 5.

Table II presents the results of our lower bound calculations on these 12 problems. The first two columns show the cost of the best known feasible solution for each problem, either obtained from the literature (column 1) or obtained as part of our computational testing (column 2).

Column 3 reports the best lower bound obtained from the Lagrangian relaxation after a number of iterations, and column 4 shows the ratio of lower to upper bounds. The lower bounds are quite tight, especially considering that in most cases we are comparing to a heuristic cost that may be higher than the optimal value.

Columns 5–7 give the cost of a spanning tree, K-tree, and degree-constrained spanning tree. This shows the contribution of the various constraints of the Lagrangian problem to the lower bound value. The difference between the value in column 7 and the final lower bound in column 3 is due to the constraints on vehicle capacity and the degree of each customer. Column 8 lists the number of vehicle capacity constraints active at the end of the run. The number of active constraints increases and decreases during the run but was generally at or near a maximum value at the end of a run.

The subgradient iterations required to obtain the lower bounds are listed in column 8. The step size in the subgradient method was set according to the usual formula (e.g., see p. 14 in Fisher 1985). We initialized $u = v = 0$ and $\lambda = 2$, and reduced λ by a factor of 0.75 if the lower bound did not improve in 30 iterations.

All computations were performed on an Apollo Domain 3000 microcomputer. Although the computation times in column 10 are large, note that we were using a small computer and these are by far the largest problems on which optimization has been attempted.

One insight these computations provide is that real problems appear to be easier for optimization than randomly generated problems. For example, compare real problems 10, 11, and the "realistic" problem 6 with problems 1–3, which are random problems of a similar size. For problems 6, 10 and 11, the lower bounds computed with 2,000 subgradient iterations were 99.1% of the optimum, on average, while for problems 1–3, the lower bounds computed with 3,000

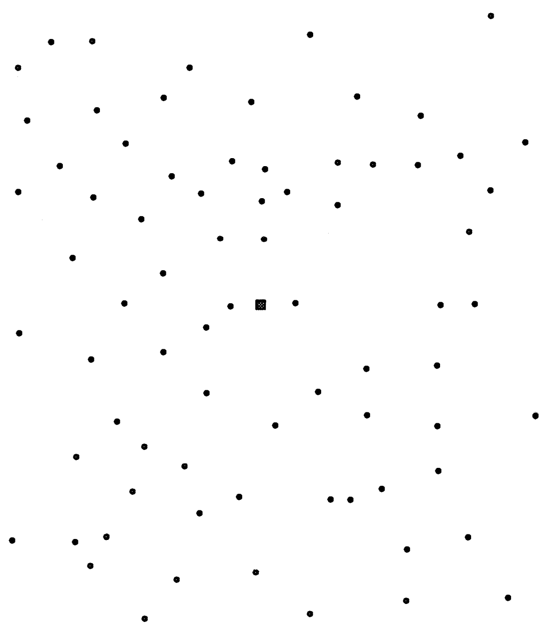

Figure 1. Customer and depot locations for problem 2 (the square is the depot and dots are customers).

Figure 2. Customer and depot locations for problem 11 (the square is the depot and dots are customers).

subgradient iterations were 94% of the optimum, on average. We will see in the next section that real problems 7–11 and "realistic" problem 6 are the problems we are able to solve to optimality.

The difference between random and real problems is illustrated in Figures 1 and 2, which show depot and customer locations for problems 2 and 11, respectively. The rather even distribution of customers in Figure 1 is typical of problems 1–5, just as the grouping of customers into clusters is typical of problems 6–12. This clustering seems to play a role akin to sparsity in linear programming in providing a structure that can make problems easier to solve if properly exploited in computations. For example, a group of customers clustered together tends to act as one customer for vehicle capacity constraints (6); they are either all in or all out of a set S defining a constraint. In the next section, we will see how these clusters also can be used in defining a branch-and-bound algorithm.

5. OPTIMAL SOLUTION WITH BRANCH AND BOUND

In this section, we describe branching rules used to obtain proven optimal solutions to problems 6, 10, and 11. (Problems 7–9 were solved to optimality without branching.) Two approaches to branching were tried.

The first is a traditional approach in which nodes of the search tree correspond to partially-sequenced sets of customers (e.g., see Christofides, Mingozzi and Toth 1981b). We branch on either edges or customers. An edge branch is executed by selecting an edge (i, j) and creating two branches, one with edge (i, j) forced into the solution and one with it forced out.

Customer branching is executed from a node of the branch-and-bound tree at which a sequence has been established for a set of customers $i_1, \ldots, i_k$ that comprises a portion of a vehicle route. We chose either end of the sequence and branch by enumerating various customers that could be appended to the partial route. Assume that we are branching from customer i_k. We execute a customer branch step by identifying a set $T \subseteq N_0 - \{i_1, \ldots, i_k\}$ of unbranched customers or depot satisfying $a\{i_1, \ldots, i_k, j\} \leq b$ for all $j \in T$. We create a node corresponding to the sequence $i_1, \ldots, i_k, j$ for all $j \in T$ and an additional node at which customer i_k cannot link to any customer $j \in T$, i.e., the edge (i_k, j) is excluded for all $j \in T$. Normally, T would be selected so that points in T are close to i_k.

A route is completed when the depot has been appended to both ends in the sequence. We do not allow a route to be completed if the unused capacity on the vehicle is so great that the remaining vehicles have insufficient capacity to deliver the remaining customer orders.

This procedure begins with an edge branch on an edge (i_1, i_2). In our computational work, i_1 was a customer farthest from the depot and i_2 was a customer closest to i_1. At the node of the search tree, where (i_1, i_2) is forced into the solution, we execute a customer branch using the partial sequence i_1, i_2. At the other node, we execute another edge branch. In general, at any node of the tree where we have defined a sequenced subset of customers corresponding to a partial vehicle route, we use customer branching. Otherwise, edge branching is used.

This procedure was applied to several of the test problems, but was unsuccessful in finding a proven optimal solution for any of them. The major problem with this approach is that the decisions resolved when we branch are quite minor. To illustrate the difficulty this can create, consider a problem with a cluster of k customers close to each other. Any solution in which these customers are delivered contiguously on the same route in some sequence will have about the same cost. Hence, when we branch to resolve the sequence for these customers, unless the lower bound is exceptionally tight, we will be unable to fathom any of the $0(k!)$ nodes generated. Looking at Figure 2, one can see many clusters of 4–5 customers where this problem could and did arise.

As suggested in Christofides, Mingozzi and Toth (1981a), a dominance fathoming test can be formulated that mitigates this problem to some extent. A node of the branch-and-bound tree that corresponds to a sequence $i_1, \ldots, i_k$ for a set of customers cannot lead to an optimal solution, and therefore can be fathomed, if there is a different sequence for the customers that begins with customer i_1, ends with customer i_k, and has lower cost. We operationalized this test by applying the Lin and Kernighan 3-opt rule to the customer sequence specified at a node. If the 3-opt rule found an improved sequence, then the node was fathomed. This dominance test reduced somewhat the problem that we have described, but not enough to allow optimal solution of any of the test cases.

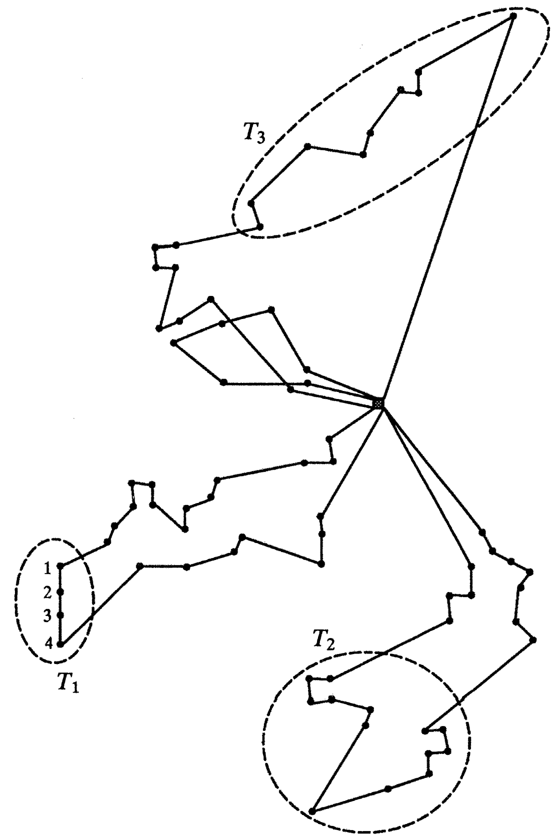

Figure 3. Examples of branch clusters for problem 11.

This experience suggests that we might obtain a better branching procedure by identifying macroproperties of an optimal solution whose violation would have a sufficiently large impact on cost to allow fathoming. Figure 3 shows the optimal solution for problem 11. One property that stands out in this optimal solution is the presence of clusters of customers delivered contiguously on the same route which are close to each other and far from the remaining customers in the problem. Three examples are encircled in Figure 3 and labeled as T_1, T_2, and T_3. Requiring the customers in any of these three clusters to be delivered on two different routes appears to have a significant impact on cost.

This observation was used to develop a new branching rule. Let $I(T)$ denote the incidence of edges on the node set T in a solution graph, i.e., $I(T) = \sum_{i \in T} \sum_{j \in \bar{T}} x_{ij}$. In the solution depicted in Figure 3, $I(T_k) = 2$, $k = 1, 2, 3$. Note that in a feasible solution the incidence on any customer set must be an even integer not less than $2\lceil a(T)/b \rceil$. We branch by selecting any set $T \subseteq N$ and creating two nodes corresponding to $I(T) = 2\lceil a(T)/b \rceil$ and $I(T) \geq 2\lceil a(T)/b \rceil + 2$. The constraint $I(T) \geq 2\lceil a(T)/b \rceil + 2$ is of the same form as (3) and, hence, is easy to incorporate within our current Lagrangian relaxation. If we branch in this way for the sets T_1, T_2, and T_3, the node corresponding to $I(T_k) \geq 4$ is fathomed by obtaining a lower bound greater than the feasible value of 244.92, the best known feasible value prior to applying branch and bound, thus establishing that $I(T_k) = 2$, $k = 1, 2, 3$ in an optimal solution to problem 11.

The constraint $I(T) = 2\lceil a(T)/b \rceil$ can be used to derive additional restrictions that tighten the Lagrangian problem. First, for any $j \notin T$ such that $a_j + a(T) > \lceil a(T)/b \rceil\, b$, we can force out of the solution the edges (i, j) for all $i \in T$. Second, if there is another set S for which $I(S) = 2$, $S \cap T = \phi$ and $a(S \cup T) > \lceil a(T)/b \rceil\, b$, we can force out of the solution the edges (i, j) for all $i \in T$, $j \in S$. Third, if $\lceil a(T)/b \rceil = 1$ and $b - a(T)$ is sufficiently small, it may be feasible to enumerate all combinations of customers that can fit in the remaining space $b - a(T)$ and branch by generating a node corresponding to each combination. Fourth, if $\lceil a(T)/b \rceil = 2$, we can select a subset $S \subset T$ of a few large customers for which it is computationally feasible to enumerate all partitions of S into two sets corresponding to the customers assigned to each of the two vehicles that must deliver the customer orders in T. We can then branch on the choice of a partition.

Finally, we describe a simple dominance test that can sometimes establish an optimal sequence for the customers in a set T with $I(T) = 2$. By way of illustration, consider set T_1 in Figure 3. We have indexed the customers in this set from 1–4. Because $I(T_1) = 2$, there will be precisely two customers in T_1 joined to customers outside of T_1. There are $\binom{4}{2} = 6$ choices for this pair of customers. Index these pairs (i_k, j_k), $k = 1, \ldots, 6$, and assume that $(i_1, j_1) = (1, 4)$. For any pair (i_k, j_k), the path through the remaining two customers must minimize cost and can be determined by enumeration. For example, it is apparent that for (i_1, j_1), the optimal path is (1, 2, 3, 4). Let C_k denote the cost of the optimal path from i_k to j_k, e.g., $C_1 = c_{12} + c_{23} + c_{34}$.

We call a pair (i_k, j_k) *dominated* if, for each pair i, j, $i \neq j$, $i \in N_0 - T$, $j \in N_0 - T$, there exists $k^* \neq k$ such that

$$C_{k^*} + \min(c_{ii_{k^*}} + c_{jj_{k^*}},\ c_{ji_{k^*}} + c_{ij_{k^*}}) < C_k + \min(c_{ii_k} + c_{jj_k},\ c_{ji_k} + c_{ij_k}). \tag{7}$$

It is clear that a path through T joining a dominated pair can be ignored as a sequence for the customers in T, because it could be replaced in any feasible solution by a different sequence (namely, the shortest path through T joining i_{k^*} and j_{k^*}) without increasing cost.

Returning to our example, for the set T_1, all pairs $k = 2, \ldots, 6$ can be shown by direct computation of (7) to be dominated by $k^* = 1$. Hence, we can fix the sequence of customers in T_1 to 1, 2, 3, 4. For the set T_3, there are four sequences that dominate all others. In this event, we can branch by selecting one of those sequences.

In our computational work, we apply the dominance test for each k by computing (7) for all possible ij and each possible k^*. The step of finding the shortest path through T that joins a pair of customers in T can be accomplished by a straightforward modification of the dynamic programming algorithm for the traveling salesman problem given by Held and Karp (1962).

There is another form of set branching that we have not implemented, but which is potentially interesting. Choose a set T, let $\bar{T} = N - T$ and branch two ways according to whether $\sum_{i \in T} \sum_{j \in \bar{T}} x_{ij}$ 0 or $\sum_{i \in T} \sum_{j \in \bar{T}} x_{ij} \geq 1$. Along the branch with $\sum_{i \in T} \sum_{j \in \bar{T}} x_{ij} = 0$, we have effectively partitioned the problem into two parts corresponding to the customer sets T and $\bar{T}$.

The ideas defined above can be combined in many ways to create a branch-and-bound algorithm, depending on how branch sets are chosen and the order in which the various methods of branching are combined. We describe here the particular algorithm used in our computations. Further research would be useful on how best to design a branching procedure using these ideas.

We need to identify sets of customers on which to branch. We used two types of branch sets. One was the set of customers S on a single route or a pair of crossed routes for which average vehicle utilization exceeded 98%. If S was a pair of crossed routes, we added to S any customers within the convex hull of $S \cup \{0\}$ that fit (i.e., $a(S) \leq 2b$ with the customers added), starting with customers farthest from the depot. Sets like these made good branch sets because the branch $I(S) = 2\lceil a(S)/b \rceil$ removes many edges from the problem, namely those edges (i, j) for which $i \in S$, $j \in \bar{S}$ and $a(S) + a_j > \lceil a(S)/b \rceil b$.

The second type of branch sets were clusters of customers $S = \{i_1, \cdots, i_k\}$ delivered contiguously in the order $i_1, \cdots, i_k$ on a single route of a starting feasible solution. We also required S to contain the customer on the route farthest from the depot, to be separable from $\bar{S}$ by a straight line, and to have a sufficiently small value of

$$D(S) = \frac{\sum_{j=1}^{k-1} c_{i_j i_{j+1}}}{(k-1)\ \min_{i \in S, j \in \bar{S}} c_{ij}}.$$

The quantity $D(S)$ is used to measure the extent to which the customers in S are close to each other and far from the remaining points $\bar{S}$. It is easy to see that the sets T_1, T_2 and T_3 in Figure 3 satisfy the required properties and have relatively small values of $D(S)$.

We describe the specific steps in our branch-and-bound procedure in the order they are executed. We first branch on any S with $a(S)$ where S is the customers of a route of the starting feasible solution $= b$ and then on $\lfloor n/10 \rfloor$ of the second type of branch sets described above with the smallest values of $D(S)$, provided we can fathom the node corresponding to $I(S) \geq 4$.

We next apply the dominance test described previously to any set S with $I(S) = 2$ and $|S| \leq 11$ (the computation time for the dominance test was prohibitive for larger sets). We branch on choice of sequence whenever at most four sequences are nondominated. For a set S with $I(S) = 2$ and $b - a(S) > 0$, but sufficiently small that at most one other customer could fit feasibly with S on a single route, we branch by enumerating all feasible completions of the vehicle route containing S.

We then branch on all single routes or pairs of crossed routes with average vehicle utilization exceeding 98%. If S is a pair of routes, at the node corresponding to $I(S) = 4$, we branch on all partitions of $R \subseteq S$ into two sets corresponding to the set of customers assigned to each of the two routes, where R contains the 11 largest customers in $S(R = S$ if $|S| \leq 11)$. A set $T \subset S$ for which $I(T) = 2$ has been imposed can be treated as a single customer for this purpose.

Finally, at any node still unfathomed, we apply the traditional branching method described at the start of this section.

Optimization using real cost coefficients must inevitably be somewhat inexact on a finite word length computer. Because of roundoff errors, the computed value of a lower bound can be less than its true value, so it becomes necessary to modify the usual fathoming criterion for a node of the branch-and-bound tree to $LB \geq UB - \epsilon$, where LB and UB are the relevant lower and upper bounds and ϵ is a small positive scalar. If the c_{ij} are real, this means the solution obtained is ϵ optimal in that it may exceed the true optimum by an additive constant ϵ. If the c_{ij} are integral, we can rely on the integrality of the objective function to use any $\epsilon < 1$ without loss of optimality.

Table III
Optimization Results

Problem	Time to Find Optimal Solution[a] Apollo Domain 3000 Minutes	Nodes in Branch-and-Bound Tree
6	591	89
7	11	1
8	64	1
9	6	1
10	342	148
11	948	37

[a]Time includes the time (reported in column 10 of Table II) to bound the root node.

The algorithm described above was applied to problems 6–11. For problems 7–9, feasible solutions were found with $UB - LB \leq 0.5$. Since the c_{ij} for these problems are integral, we know that these solutions are exactly optimal. For problems 6, 10, and 11, with real c_{ij}, $\epsilon = 0.0001$ was used. Results are shown in Table III.

Note that it is fairly common in computational studies on vehicle routing and the traveling salesman problem to circumvent the problem of limited accuracy in computations by rounding c_{ij} to integral values. While this finesses a messy technical problem, it can also introduce serious inaccuracies, because *each* c_{ij} is now subject to an error of up to 0.5 units. This is particularly a problem if the distance between two customers is less than 0.5, because then these customers can appear on the same vehicle in the wrong order, creating credibility problems with drivers. Optimizing with real costs avoids this problem.

6. EXTENSIONS

The K-tree approach generalizes quite readily to incorporate some complications commonly found in real vehicle routing problems. We will describe some of these extensions. How well the extended algorithm would perform computationally is an open question at this point.

Suppose that vehicle capacity is defined by m dimensions. For example, if $m = 2$ the dimensions of capacity might be weight and volume. Let $a_{\ell j}$ be the size of customer order j in dimension ℓ and b_ℓ the capacity of a vehicle in dimension ℓ. A set S of customers assigned to a vehicle must satisfy $\sum_{j \subset \in S} a_{\ell j} \leq b_\ell$, $\ell = 1, \ldots, m$. To accommodate this variation, simply redefine $r(s) = \max_\ell \lceil \sum_{j \in S} a_{\ell j}/b_\ell \rceil$ in constraint (3).

We next show how to extend our method for vehicle routing problems with asymmetric costs and time windows. Asymmetric costs and time windows are natural to treat together, because both extensions require that we now consider the direction in which a route is driven.

Define

c_{ij} = the cost of direct travel from point i to point j;
t_{ij} = the time required for direct travel from i to j;
e_i = the earliest time that a delivery can begin at customer i (e_0 is the earliest time a vehicle can leave the depot);
ℓ_i = the latest time that a delivery can begin at customer i (ℓ_0 is the latest time a vehicle can return to the depot).

Define the variables

$$y_{ij} = \begin{cases} 1, & \text{a vehicle travels directly from } i \text{ to } j \\ 0, & \text{otherwise;} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{if } y_{ij} \text{ or } y_{ji} = 1 \\ 0, & \text{otherwise;} \end{cases}$$

$Y = \{y : y_{ij} = 0 \text{ or } 1\}$;

$X = \{x : x_{ij} = 0$ or 1 and x is the incidence vector of a K-tree with degree $2K$ on the depot$\}$.

We can represent the asymmetric time window routing problem as

$$Z^* = \min_{x \in X, y \in Y} \sum_{i,j \in N_0, i \neq j} c_{ij} y_{ij} \tag{8}$$

subject to

$$x_{ij} = y_{ij} + y_{ji} \quad \text{for all } ij \tag{9}$$

$$\sum_{i \in N_0, i \neq j} y_{ij} = \begin{cases} K, & \text{if } j = 0 \\ 1, & j \in N \end{cases} \tag{10}$$

$$\sum_{j \in N_0, j \neq i} y_{ij} = \begin{cases} K, & \text{if } i = 0 \\ 1, & i \in N \end{cases} \tag{11}$$

$$\sum_{i \in S} \sum_{j \in \bar{S}} x_{ij} \geq 2r(S) \quad \text{for all } S \subseteq N \text{ with } |S| \geq 2 \tag{12}$$

$$\sum_{j=1}^{m_P - 1} y_{i_j, i_{j+1}} \leq m_P - 2 \quad \text{for any time window violating directed path } P = (i_1, \ldots, i_{m_P}). \tag{13}$$

In this formulation, the path $P = (i_i, \ldots, i_{m_P})$ is time window violating if the start times $T_{i_1} = e_{i_1}$ and $T_{i_j} = \max(e_{i_j}, T_{i_{j-1}} + t_{i_{j-1}, i_j})$, $j = 2, \ldots, m_P$ has $T_{i_j} > \ell_{i_j}$ for at least one j. The quantity $r(S)$ is the tightest lower bound we are able to compute on the number of vehicles required to service the customers in set S.

Clearly, $r(S) \geq \lceil a(S)/b \rceil$. The time window constraints can be used to tighten this lower bound. For example, if $a(S) \leq b$ and $|S|$ is sufficiently small to enumerate all sequences, we can determine whether there is a time window feasible route for the customers in S and set $r(S) = 2$ if none exists. We also know that $r(S) \geq 2$ if $a(S) \leq b$ the length of a minimum spanning tree of S with respect to the t_{ij} exceeds $\max_{i \in S} \ell_i - \min_{i \in S} e_i$.

Let u_i^1, u_i^2, v_S, and w_P denote dual variables for (10)–(13), respectively, and $P(ij)$ denote the set of time window violating paths P that contain nodes i and j with i immediately preceding j. Then we can define the Lagrangian relaxation

$$\begin{aligned} Z_D(u, v, w) = {} & \sum_{i=1}^{n} (u_i^1 + u_1^2) + K(u_0^1 + u_0^2) \\ & + \sum_{S \subseteq N, |S| \geq 2} (v_S) r(S) \\ & - \sum_{P} w_P (m_P - 2) \\ & + \min_{x \in X, y \in Y} \sum_{i,j \in N_0, i \neq j} \tilde{c}_{ij} y_{ij} \end{aligned}$$

subject to $x_{ij} = y_{ij} + y_{ji}$ for all ij where

$$\tilde{c}_{ij} = c_{ij} - u_j^1 - u_i^2 - \sum_{\substack{S \text{ such that} \\ i \in S, j \in \bar{S} \\ \text{or} \\ i \in \bar{S}, j \in S}} v_S + \sum_{P \in P(ij)} w_P.$$

We can choose optimal y given x by setting y_{ij} or y_{ji} to x_{ij}, depending on whether $\tilde{c}_{ij}$ or $\tilde{c}_{ji}$ is smaller. Optimization of the Lagrangian problem then reduces to $\min_{x \in X} \sum_{ij} \bar{c}_{ij} x_{ij}$, where $\bar{c}_{ij} = \min\{\tilde{c}_{ij}, \tilde{c}_{ji}\}$. This is a degree-constrained K-tree problem that can be solved using the algorithm in Fisher (1994).

Now consider a symmetric vehicle routing problem with a nonuniform fleet. Let b_k denote the capacity of vehicle k and assume that $b_1 \geq b_2 \geq \ldots \geq b_k$. In this case, we can define $r(S)$ by $\sum_{i=1}^{r(S)-1} b_i < a(S) \leq \sum_{i=1}^{r(S)} b_i$. Constraints (3) are now necessary, but not sufficient to impose vehicle capacity feasibility. Still, they may be adequate in some cases to obtain reasonable lower bounds. If not, we can reformulate the problem as follows. Let

$$y_{ik} = \begin{cases} 1, & \text{if customer } i \text{ is assigned to vehicle } k \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{vehicle } k \text{ travels directly between customers } i \text{ and } j \\ 0, & \text{otherwise.} \end{cases}$$

Let x be a vector of the x_{ijk} variables. As before, we can define a correspondence between x values and the selection of edges in a complete graph on the node set

N_0. In this case, we assume there are K edges joining nodes i and j, and let $\sum_{k=1}^{K} x_{ijk}$ denote the number of edges selected between nodes i and j. Let X denote the set of x values corresponding to a K-tree with degree $2K$ on the depot. Then we can model the nonuniform fleet vehicle routing problem as

$$\min_{x \in X} \sum_{k=1}^{K} \sum_{ij \in E(N_0)} c_{ij} x_{ijk} \tag{14}$$

$$\sum_{\substack{j \in N_0 \\ j \neq i}} x_{ijk} = 2y_{ik} \quad \text{for } i \in N \text{ and } k = 1, \ldots, K \tag{15}$$

$$\sum_{k} y_{ik} = 1 \qquad i \in N \tag{16}$$

$$\sum_{i \in N} a_i y_{ik} \leq b_k \quad k = 1, \ldots, K. \tag{17}$$

Dualizing (15) and (16) gives a Lagrangian problem consisting of a K-tree problem and a knapsack problem. In the K-tree problem, we have K edges between every pair of points, but we always choose the edge with the currently smallest reduced cost, so we obtain a K-tree problem of the form discussed previously.

Finally, the K-tree approach can be adapted to a problem in which some customer deliveries can be omitted at a penalty cost by using the algorithm in Tang (1989) for a minimum spanning tree problem in which some points may be omitted at a cost.

APPENDIX

Data for New Problems and Solutions

Tables IV–VI contain the data for problems 10–12 as well as the optimal solutions for problems 10–11 and a feasible solution with value 1163.599 for problem 12. These problems are planar and the columns labeled X and Y give the coordinates of each customer. The cost c_{ij} of travel between customers i and j is the Euclidean distance between the customers. The column labeled Demand gives a_j for each customer. Because problems 7–9 are nonplanar, the input data are too voluminous to include, but are available on request from the author. Table VII gives the optimal solution to problem 6 with value 819.56. In reporting solutions, we assume that customers are indexed in the order they appear in the data.

Table IV
Data and Optimal Solution for Problem 10
(n = 44, K = 4, b = 2,010,
depot coordinates 0.00 and 0.00)

X	Y	Demand
3	5	33
2.5	9	15
48	16	10
48	17	40
69	16	15
70	16	5
64	13	77
3	−22	435
2.5	1	165
−13	11.5	120
−20	45	65
−9	52	23
−8.5	53	18
−8	52	550
2	2	78
−2	9	627
−10	20	9
−20	19	96
−15	−21	116
−5	−9	116
−4.5	−9	83
−52	−36	41
−53	−36	645
0	0.01	694
−30	−18	573
−51	−35	1
81	9	181
84	−99	106
82	−6	52
40	−12	117
50	−7	52
51	−8	1300
63	−17	57
45	−1	28
54	8.5	84
29	4	1
21	3	54
22	2	19
39	−3	88
39.5	−3	41
40	−11	238
28	−2	66
24	−18	44
24	−19	42

Route 1: 24 16 2 1 15 9
Route 2: 8 19 22 23 26 25 20 21
Route 3: 43 44 28 33 29 27 6 5 7 35 3 4 14 13 12 11 18 17 10
Route 4: 37 38 36 39 40 34 31 32 30 41 42

ACKNOWLEDGMENT

I am indebted to Baoxing Tang and Gang Yu for useful comments provided on earlier drafts of this paper, to Tom Brinker and Paul Prutzman of Air Products and Chemicals, Inc. for providing the data in problems 7–9, to Stephen Shepard and Carol Galeener of Exxon Corporation for providing the data used in problem 11, and to Julien Bramel and David Simchi-Levi for applying their algorithm to obtain feasible solutions for problems 7–12. The extension to

Table V
Data and Optimal Solution for Problem 11
($n = 71$, $K = 4$, $b = 30{,}000$,
depot coordinates 0.00 and 0.00)

X	Y	Demand
−12	−6	7,063
−15	−5	51
−1	−18	23
2	−21	3,074
−1	−17	349
−9	−12	1,047
2	−22	698
1	−21	3,001
7	−25	31
−7	−17	1,135
−11	−5	21,611
−14	−9	57
−14	−8	51
−11	−2	551
−14	−5	179
−15	−9	6
−15	−8	528
−9	−6	2,832
−14	−4	1,514
3	6	889
5	9	2,554
5	10	1,215
2	8	1,810
1	9	3,050
1	10	4
2	9	1,563
6	14	741
5	12	1,532
3	7	709
4	8	1,022
−6	1	883
−6	2	1,689
−8	−2	10,235
−7	2	29
−7	−3	2,894
−7	−2	450
−20	12	411
−20	13	207
−12	10	496
−20	15	1,021
−6	8	117
2	21	46
2	22	8
1	21	18
−1	24	561
1	22	1,877
−2	20	3,542
−2	21	801
−4	18	967
−4	19	62
−5	18	1,366
−5	26	230
1	23	4
−6	5	12
−6	6	145
−9	6	7,149
−9	7	2,250
−12	5	383
−12	6	134
−11	3	1,947
−11	4	182
−14	4	3,934
−14	5	468
−15	4	18
−15	5	133
−16	7	2,340
−16	8	754
−15	10	1,264
−20	10	806
−5	19	3,665
−9	−11	2,452

Route 1: 20 29 30 21 22 28 27 42 43 44 46 53 45 52 48 47 50 70 51 49 25 24 26 23
Route 2: 35 18 11 19 14 36
Route 3: 9 7 4 8 3 5 10 6 71 12 16 17 13 2 15 1 33
Route 4: 54 55 41 56 57 39 68 40 38 37 69 67 66 65 64 62 63 59 58 61 60 34 32 31

Table VI
Data and Optimal Solution for Problem 12
($n = 134$, $K = 7$, $b = 2{,}210$,
depot coordinates −6 and 15)

X	Y	Demand
3.2	5.1	30
24.6	8.3	226
23.3	1.3	37
27.8	8.3	24
29.0	8.0	36
31.0	8.0	1
33.5	10.5	31
30.0	10.5	24
29.0	10.0	30
26.5	11.7	24
28.3	14.3	24
27.0	14.3	32
23.5	19.0	24
26.0	20.0	24
25.0	20.0	19
20.5	19.0	24
−20.0	13.0	18
−21.0	14.0	36
−30.0	30.0	115
−5.0	30.0	24
1.3	17.8	24
1.8	13.8	61
1.8	13.1	71
2.0	13.6	36
4.8	17.0	18
7.0	15.0	30
9.8	16.6	31
11.4	14.5	36
14.4	11.3	18
11.0	12.0	1,004
9.3	10.7	18
0.6	2.8	34
−30.0	−10.0	504
2.0	0.0	18
14.5	1.0	39
15.0	1.8	24
17.2	2.4	37
17.2	4.2	24
18.2	4.4	99
20.3	2.1	24
22.8	3.1	24
23.0	4.0	36
20.8	4.0	30
20.8	4.0	25
18.5	6.4	24
−14.0	16.0	122
−0.5	6.9	196
3.2	2.8	229
5.6	1.8	83
8.7	2.8	18
9.0	3.3	24
9.0	3.5	306
11.2	3.3	18
10.8	4.7	20
11.5	4.6	18
12.3	4.7	24
12.3	5.5	22
11.2	6.9	24
6.5	9.7	18
5.8	8.5	18
7.2	6.0	24
7.2	4.0	24
−4.0	−4.0	30
−3.0	1.2	24
−40.0	49.0	40
−15.0	10.0	166
−11.0	−10.0	254
−25.0	−20.0	187
−25.0	−35.0	94
−24.0	−35.0	17
−18.0	10.0	285
−2.0	10.0	24
−4.0	8.0	24
−3.0	5.0	205
2.1	6.2	23
−1.7	3.0	28
−3.0	2.0	51

(Table continued)

Table VI
(Continued)

X	*Y*	Demand
−7.0	0.0	49
−3.0	−6.0	19
−30.0	−11.0	262
−62.0	−10.0	120
−8.0	30.0	266
1.0	60.0	704
10.0	52.0	38
10.0	52.0	18
10.0	51.0	30
16.0	29.0	25
26.0	21.0	12
16.0	21.0	18
15.5	19.2	25
0.0	16.5	35
17.2	14.3	18
16.5	7.8	12
16.9	7.7	20
18.0	2.0	1126
16.2	4.0	9
15.0	4.0	36
15.0	3.0	12
14.8	2.4	31
14.5	3.0	96
13.0	2.6	27
11.8	3.0	54
12.0	4.0	137
12.8	3.6	12
13.4	5.5	58
−150.0	8.0	206
−152.0	1.0	178
−152.0	0.0	486
−142.0	−31.0	36
−78.0	−19.0	261
−78.0	−18.0	135
−78.0	−17.0	135
−80.0	−14.0	373
−118.0	22.0	535
−107.0	30.0	42
−85.0	14.0	9
−78.0	15.0	110
−15.0	16.0	36
−62.0	32.0	18
−120.0	−20.0	726
−90.0	−22.0	187
−79.0	−19.0	23
−79.0	−18.5	134
−79.0	−18.0	47
−78.0	−17.5	51
−79.0	−17.0	43
−80.0	−17.0	79
−80.0	−16.0	112
−80.0	−15.0	91
−48.0	37.0	232
−85.0	15.0	483
−62.0	−9.0	828
−15.0	−4.0	11
−1.0	3.2	12

Route 1: 78 133 68 70 69 110 122 123 124 111 125 112 126 127 121 128 129 113 81 17
Route 2: 120 109 108 107 106 114 115
Route 3: 19 65 130 119 117 131 116 132 18 118 46
Route 4: 91 21 25 26 27 28 92 29 93 94 45 44 43 40 3 41 42 2 4 5 6 7 8 9 10 12 11 14 88 15 13 16 90 89 87 86 85 84 83 20 82
Route 5: 60 61 54 55 105 97 96 38 39 95 37 98 100 99 36 35 101 104 102 53 50 49 48 34 32 134 76
Route 6: 22 24 23 59 31 30 58 57 56 103 51 52 62 1 75 47 72
Route 7: 66 71 33 80 67 79 63 64 77 74 73

time windows was developed during discussions with Kurt Jornsten, Ming Lin, and Oli Madsen. The research described in this paper was supported in part by National Science Foundation grants SES-88-12006 and SES-91-09798 and by NATO Research grant 5-2-05-CRG.890409.

Table VII
Optimal Solution for Problem 6

Route	Solution
1	75 1 2 4 6 9 11 8 7 3 5
2	55 54 53 56 58 60 59 57
3	98 96 95 94 92 93 97 100 99
4	32 33 31 35 37 38 39 36 34
5	20 24 25 27 29 30 28 26 23 22 21
6	67 65 63 74 62 66
7	47 49 52 50 51 48 46 45 44 40 41 42 43
8	81 78 76 71 70 73 77 79 80 72 61 64 68 69
9	10 12 14 16 15 19 18 17 13
10	90 87 86 83 82 84 85 88 89 91

DEDICATION

This paper is dedicated to the memory of Darwin Klingman whose untimely death was a great loss, not only to our profession but personally to those who knew him as a friend. His colorful and enthusiastic approach to life was an inspiration; his intellectual innovations will influence the style of our research for many years. This paper both draws on his published work and benefits from personal comments provided near the end of his life.

REFERENCES

ARAQUE, J. R. 1989. Contributions to the Polyhedral Approach to Vehicle Routing. Ph.D. Thesis, SUNY at Stony Brook.

ARAQUE, J. R., L. A. HALL AND T. L. MAGNANTI. 1990. Capacitated Trees, Capacitated Routing, and Associated Polyhedra. OR Center Report 232-90, Massachusetts Institute of Technology, Cambridge, Mass.

ARRIZZA, A., AND P. KARELLAS. 1983. A Feasibility Study of the ROVER System at National Grocers Co. Ltd. B.A. Thesis of Applied Science, Department of Industrial Engineering, University of Toronto, Canada.

BODIN, L. D., B. L. GOLDEN, A. ASSAD AND M. BALL. 1983. Routing and Scheduling of Vehicles and Crews: The State of the Art. *Comp. & Opns. Res.* **10,** 69–211.

BOUSBA, C., AND L. A. WOLSEY. 1989. Finding Minimum Cost Directed Trees With Demands and Capacities. CORE Discussion Paper 8913, Catholic University of Louvain, Louvain-la-Neuve, Belgium.

BRAMEL, J., AND D. SIMCHI-LEVI. 1992. A Location Based Heuristic for General Routing Problems, Department of Industrial Engineering & Operations Research, Columbia University, New York.

CHRISTOFIDES, N. 1985. Vehicle Routing. In *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, E. L. Lawler, J. K. Lenstra,

A. H. G. Rinnooy Kan and D. B. Shmoys (eds.). John Wiley, New York.

Christofides, N., and S. Eilon. 1969. An Algorithm for the Vehicle Dispatching Problems. *Opns. Res.* **20.**

Christofides, N., A. Mingozzi and P. Toth. 1981a. Exact Algorithms for the Vehicle Routing Problem Based on Spanning Tree and Shortest Path Relaxations. *Math. Prog.* **20,** 255–282.

Christofides, N., A. Mingozzi and P. Toth. 1981b. Space State Relaxation Procedures for the Computation of Bounds to Routing Problems. *Networks* **11,** 145–164.

Christofides, N., A. Mingozzi and P. Toth. 1979. The Vehicle Routing Problem. In *Combinatorial Optimization*, N. Christofides, A. Mingozzi, P. Toth, C. Sandi, (eds.). John Wiley, Chichester, U.K., 318–338.

Cornuejols, G., and F. Harche. 1989. Polyhedral Study of the Capacitated Vehicle Routing Problem. Management Science Research Report No. 553, Carnegie Mellon University, Pittsburgh, Pa.

Fisher, M. L. 1985. An Applications Oriented Guide to Lagrangian Relaxation. *Interfaces* **15,** 10–21.

Fisher, M. L. 1988. Lagrangian Optimization Algorithms for Vehicle Routing Problems. In *Operational Research '87*, G. K. Rand (ed). 635–649.

Fisher, M. L. 1992. Vehicle Routing. In *Handbook on Operations Research and Management Science: Volume on Networks and Distribution* (to appear).

Fisher, M. L. 1994. A Polynomial Algorithm for the Degree Constrained K-Tree Problem. *Opns. Res.* **42,** 776–780.

Fisher, M. L., and R. Jaikumar. 1978. A Decomposition Algorithm for Large-Scale Vehicle Routing. Working Paper 78-11-05, Department of Decision Sciences, University of Pennsylvania, Philadelphia.

Fisher, M. L., and R. Jaikumar. 1981. A Generalized Assignment Heuristic for Vehicle Routing. *Networks* **11,** 109–124.

Gavish, B. 1982. Topological Design of Centralized Computer Networks—Formulations and Algorithms. *Networks* **12,** 355–377.

Gavish, B. 1985. Augmented Lagrangean Based Algorithms for Centralized Network Design. *IEEE Trans. Commun.* **COM-33** (12), 1247–1257.

Glover, F., and D. Klingman. 1974a. A Note on Admissible Exchanges in Spanning Trees. *Adv. in Mgmt. Studies* **3.2** 101–104.

Glover, F., and D. Klingman. 1974b. Finding Minimum Spanning Trees With a Fixed Number of Links at a Node. In *Colloquia Mathematica Societatis Janos Bolvai*, Hungary, 425–439.

Held, M., and R. M. Karp. 1962. A Dynamic Programming Approach to Sequencing Problems. *J. Soc. for Ind. and Appl. Math.* **10,** 225–248.

Held, M., and R. M. Karp. 1971. The Traveling-Salesman Problem and Minimum Spanning Trees: Part II. *Math. Prog.* **1,** 6–25.

Laporte, G., and Y. Nobert. 1987. Exact Algorithms for the Vehicle Routing Problem. *Ann. Disc. Math.* **31,** 147–184.

Laporte, G., Y. Nobert and M. Desrochers. 1985. Optimal Routing Under Capacity and Distance Restrictions. *Opns. Res.* **33,** 1050–1073.

Lin, S., and B. Kernighan. 1973. An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Opns. Res.* **21,** 498–516.

Lucena, A. P. 1986. Exact Solution Approaches for the Vehicle Routing Problem. Ph.D. Thesis, Department of Management Science, Imperial College of Science and Technology, University of London, U.K.

Martello, S., and P. Toth. 1990. Lower Bounds and Reduction Procedures for the Bin Packing Problem. *Disc. Appl. Math.* **28,** 59–70.

Nobert, Y. 1982. Construction d'Algorithmes Optimaux pour des Extensions au Probleme du Voyageur de Commerce. Doctoral Thesis, Departement d'Informatique et de Recherche Operationnelle, University of Montreal, Canada.

Osman, I. H. 1993. Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem. *Ann. Opns. Res.* **41,** 421–451.

Taillard, E. 1992. Parallel Iterative Search Methods for Vehicle Routing Problems. ORWP 92/03, Department of Mathematics, Ecole Polytechnique Fédérale de Lausanne, Switzerland.

Tang, B. 1989. Efficient Algorithms for Some Combinatorial Problems. Ph.D. Thesis, Department of Decision Sciences, The Wharton School, University of Pennsylvania, Philadelphia.