CYCLIC TRANSFER ALGORITHMS FOR MULTIVEHICLE ROUTING AND SCHEDULING PROBLEMS

PAUL M. THOMPSON
Santa Clara University, Santa Clara, California

HARILAOS N. PSARAFTIS
National Technical University of Athens, Athens, Greece

(Received July 1989; revisions received August 1991, February, July 1992; accepted January 1993)

This paper investigates the application of a new class of neighborhood search algorithms—cyclic transfers—to multivehicle routing and scheduling problems. These algorithms exploit the two-faceted decision structure inherent to this problem class: First, assigning demands to vehicles and, second, routing each vehicle through its assigned demand stops. We describe the application of cyclic transfers to vehicle routing and scheduling problems. Then we determine the worst-case performance of these algorithms for several classes of vehicle routing and scheduling problems. Next, we develop computationally efficient methods for finding negative cost cyclic transfers. Finally, we present computational results for three diverse vehicle routing and scheduling problems, which collectively incorporate a variety of constraint and objective function structures. Our results show that cyclic transfer methods are either comparable to or better than the best published heuristic algorithms for several complex and important vehicle routing and scheduling problems. Most importantly, they represent a novel approach to solution improvement which holds promise in many vehicle routing and scheduling problem domains.

Vehicle routing and scheduling problems comprise an interesting and important class of combinatorial problems (Magnanti 1981, Bodin et al. 1983, Laporte and Nobert 1987, Golden and Assad 1988). Their economic importance is marked by their presence in many areas of the manufacturing and service industries. In practice, countless variations of these problems exist. The most common may involve physical vehicles, but often the term vehicle is used quite abstractly.

To date, neighborhood search algorithms for vehicle routing and scheduling problems have focused almost exclusively on single vehicle problems such as the TSP (Croes 1958, Lin 1965, Lin and Kernighan 1973, Stewart 1987, Johnson, McGeoch and Rothberg 1987) and constrained versions of the TSP (Psaraftis 1983, Savelsbergh 1985, van der Bruggen, Lenstra and Schuur 1991). Aside from finding locally optimal transfers for each vehicle through its assigned demands, these procedures do not easily extend to multivehicle problems. Bodin and Sexton's (1983) swapper heuristic for dial-a-ride problems, the 3-opt method of Baker and Schaffer (1986) for time constrained vehicle routing problems, and the k-opt procedure of Potvin, Lapalme and Rousseau (1989) for the MTSP are the only local search procedures we are aware of that are specifically designed for multivehicle problems.

In this paper, we investigate the application of a new class of neighborhood search algorithms—cyclic transfers—to multivehicle routing and scheduling problems. We show that despite their poor theoretical worst-case performance and inherent computational complexity, these methods are either comparable to or improve the best published heuristics for several complex and important multivehicle routing and scheduling problems.

The paper is organized as follows. Section 1 reviews the theory of cyclic transfers, and extends it for vehicle routing and scheduling problems. Section 2 analyzes the worst-case performance of cyclic transfer algorithms for this group of problems. Section 3 develops a computationally efficient approximation scheme for finding negative cost cyclic transfers for vehicle

Subject classifications: Programming, integer, heuristic: cyclic transfer algorithms for fleet planning problems. Transportation, vehicle routing: cyclic transfer algorithms for multivehicle problems.
Area of review: DISTRIBUTION, TRANSPORTATION AND LOGISTICS.

0030-364X/93/4105-0935$01.25

routing problems. Finally, Section 4 summarizes computational experience with several different vehicle routing and scheduling problems.

1. CYCLIC TRANSFERS

To keep this paper relatively self-contained, we briefly review the theory of cyclic transfers, as applied to vehicle routing and scheduling. The interested reader should refer to the source papers (Thompson 1988 and Thompson and Orlin 1989) for details.

The central concept behind cyclic transfers, as applied to vehicle routing and scheduling problems, is the attempt to improve the total cost of a set of routes by transferring small numbers of demands among routes, in a cyclical manner. Thus, cyclic transfer algorithms exploit the two-phase "cluster/route" decision structure that underlies these problems. Cyclic transfers apply to vehicle routing and scheduling problems as follows: Let $[I^1, \ldots, I^m]$ be the set of routes that form a feasible solution to a vehicle routing and scheduling problem. Let ρ be a cyclic permutation of a subset of $\{1, \ldots, m\}$, for example, $\rho = (2\ 5\ 3)$ maps 2 into 5, maps 5 into 3, and maps 3 into 2. Thus $\rho(2) = 5$, $\rho(5) = 3$, and $\rho(3) = 2$. The simultaneous transfer of demands from I^j to $I^{\rho(j)}$ for each j is a *cyclic transfer.*

In this paper, we restrict our attention to a special class of cyclic transfers, *cyclic k-transfers.* These transfer exactly k demands from I^j to $I^{\rho(j)}$ for each j for some fixed integer k. A special case is *b-cyclic k-transfers,* which occur if the cyclic permutation has fixed cardinality b. Figure 1 shows a 3-cyclic 2-transfer, with $I^1 = \{A_1, A_2, A_3, A_4, A_5\}$, $I^2 = \{B_1, B_2, B_3, B_4, B_5\}$, $I^3 = \{C_1, C_2, C_3\}$, $I^4 = \{D_1, D_2, D_3, D_4\}$, and $\rho = (1\ 2\ 3)$. This cyclic transfer simultaneously moves $\{A_1, A_3\}$ from I^1 to I^2, $\{B_1, B_5\}$ from I^2 to I^3, $\{C_2, C_3\}$ from I^3 to I^1, and leaves I^4 unchanged.

We also study a generalized class of cyclic k-transfers by allowing k *dummy demands* on each route. This allows the transfer of (real) demand sets among permutations (rather than cyclic permutations) of routes. A special case, applied by Bodin and Sexton to a dial-a-ride problem, occurs when ρ has cardinality 2. These *k-transfers* simply transfer k real demands from one route to another.

The cost of a cyclic transfer is the change in optimal objective function value caused by the cyclic transfer. Let $I = \{I^1, I^2, \ldots, I^p\}$ and $J = \{J^1, J^2, \ldots, J^p\}$ represent a set of p routes before and after a cyclic transfer occurs, and let $f(I)$ represent the optimal cost

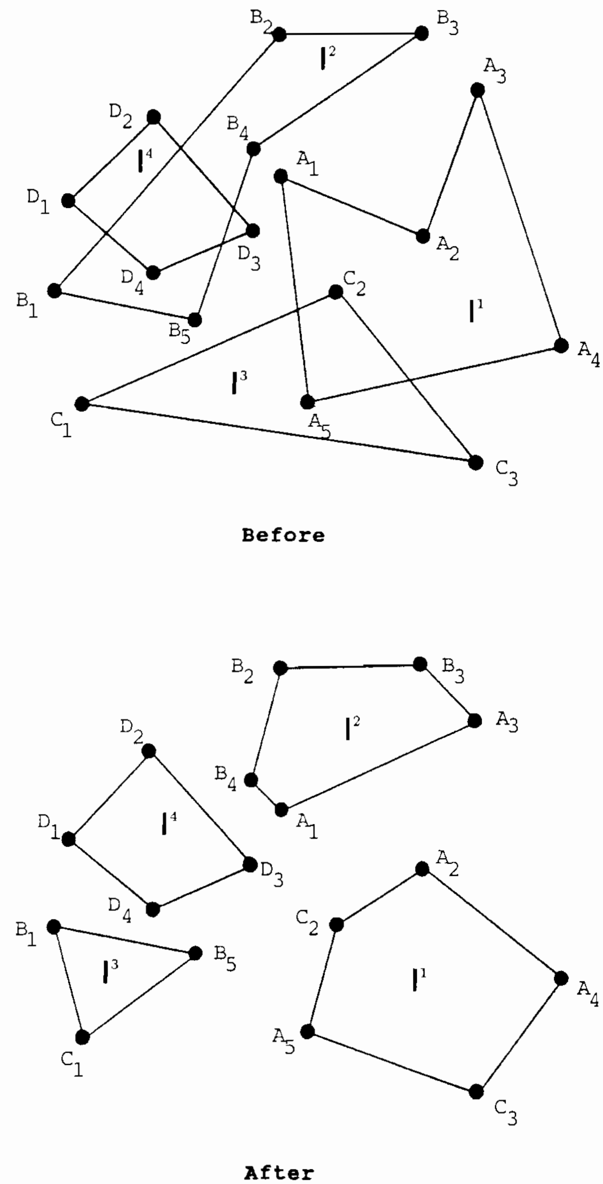

Figure 1. The effect of a 3-cyclic 2-transfer.

of route I. Then the cost of the cyclic transfer is

$$\sum_{i=1}^{p} [f(J^i) - f(I^i)].$$

The *cyclic transfer neighborhood* of a feasible solution r to a vehicle routing and scheduling problem is the set of feasible solutions reachable from r via a cyclic transfer. Moreover, r is *cyclic transfer optimal* (CT-opt) if no member of the cyclic transfer neighborhood of r has a better objective function value, i.e., if all cyclic transfers for r have nonnegative cost. An analogous definition holds for cyclic k-transfer neighborhoods.

Thompson and Orlin develop a general methodology for cyclic transfer neighborhood search, which involves transforming the search for negative cost cyclic transfers into a search for negative cost cycles on an auxiliary graph. We apply their method to vehicle routing and scheduling problems as follows. Let $G^k = (V^k, A^k, C^k)$ be an *auxiliary graph*, defined by

$V^k = \{$sets of k distinct demands on the same route$\}$;
$A^k = \{(i,j): i,j \in V^k, I(i) \neq I(j)$, route $\{I(j)+i-j\}$ is feasible$\}$;
$C^k = \{c_{ij}: (i,j) \in A^k\}$ are arc costs,

where $I(j)$ denotes the route to which a set j of k demands is assigned. For a minimization (maximization) problem, the cost c_{ij} of each arc (i, j) in A^k is equal to the increase (decrease) in the cost of route $I(j)$, due to simultaneously adding i to and removing j from $I(j)$, i.e., $c_{ij} = f(I(j) + i - j) - f(I(j))$. Here c_{ij} does not include any costs due to removing i from $I(i)$ or adding j to any other route.

Thompson and Orlin show the following theorem.

Theorem 1. *Negative cost cyclic k-transfers correspond uniquely to negative cost cycles through distinct clusters, i.e., cycles whose vertices correspond to demand sets in different routes in* G^k.

Thus, the search for negative cost cyclic transfers is equivalent to the search for negative cost cycles through distinct clusters in G^k. We will see in a later section that finding such cycles is difficult. Nonetheless, our computational results show the efficacy of this approach to solving vehicle routing and scheduling problems.

Thompson shows that for any k it is possible to have so poor an assignment of demands to vehicles in a VRP that all arcs in G^k (and, hence, all cyclic transfers) have negative cost. Indeed, it is not hard to construct instances that are 3-optimal as well. While such situations undoubtedly occur rarely in practice, they do illustrate the appeal of methods for improving cluster quality, such as cyclic transfers.

2. WORST-CASE ANALYSIS

In this section we develop performance guarantees for cyclic transfer algorithms for vehicle routing and scheduling problems. In undertaking this analysis, we focus on the Euclidean metric, because worst-case results using this restricted metric provide a valid lower bound for the worst-case performance of more general (e.g., triangle inequality) metrics.

We begin with the classical VRP. We wish to find R_{wc}, the worst-case error ratio of cyclic k-transfer optimal tour length to optimal tour length, for arbitrary k. For any VRP algorithm that optimally routes each vehicle over its demands, it is easy to show (Thompson) that an upper bound on R_{wc} is m, the number of vehicles, if arc costs satisfy the triangle inequality. We demonstrate that this bound is tight for cyclic transfer methods.

Theorem 2. *The Euclidean VRP cyclic k-transfer algorithm has a worst-case performance equal to m, the number of vehicles, for all k.*

Proof. Consider the following Euclidean VRP instance. Let m radial lines emanate from a central depot and be separated, in succession, by angles of δ radians with $\delta \ll 2\pi/m$. On each radial line, locate $(k + 1)$ unit demands a unit distance from the depot. On $(m - 1)$ of the lines, locate an additional $(m - 1)(k + 1)$ unit demand a distance δ from the depot. Set vehicle capacities at $m(k + 1)$ units.

The solution in which each vehicle serves all the demands on one radial line is cyclic k-transfer optimal, for a total distance of $2m$. The optimal solution allocates one vehicle to all customers a unit distance from the depot and $m - 1$ vehicles to the inner customers, for a total distance of $2 + 3(m - 1)\delta$. As δ decreases to zero, R_{wc} asymptotically approaches m.

Now, we show that no performance guarantees exist for cyclic transfer procedures for multidepot and tardiness-minimizing VRPs.

Theorem 3. *The multidepot Euclidean VRP cyclic k-transfer algorithm has unbounded worst-case performance for all k.*

Proof. Consider the VRP instance illustrated in Figure 2, with depots at nodes 1 and 5. Each depot houses one vehicle with capacity $3(k + 1)$ units. Each node, aside from the depots, houses $(k + 1)$ unit demands. The routes [1, 2, 3, 4, 1] and [5, 6, 7, 8, 5] are cyclic k-transfer optimal, with total length $4(1 + \delta)$. If $2\delta \leq L$, then the tours [1, 2, 7, 8, 1] and [5, 6, 3, 4, 5] are optimal, with total length 12δ. As $\delta \to 0$, R_{wc} increases without bound.

Theorem 4. (Thompson) *The cyclic k-transfer algorithm has unbounded worst-case performance for the tardiness-minimizing VRP for all k, where tardiness is the nonnegative portion of lateness.*

We omit the proof. It suffices merely to point out a case with zero optimal tardiness for which there

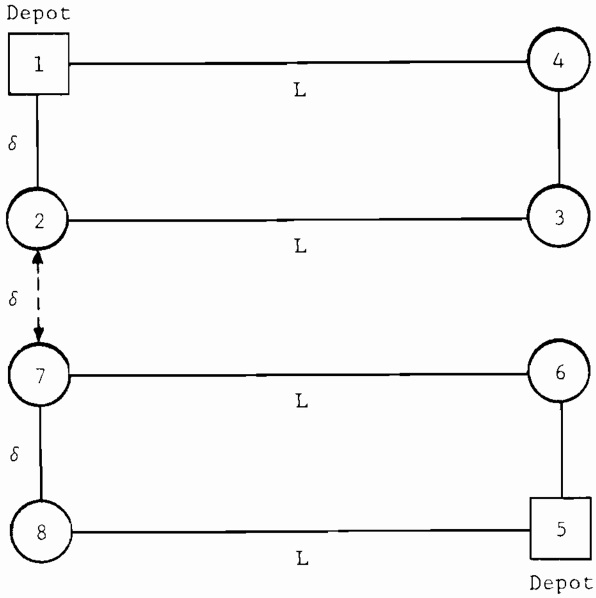

Figure 2. A CT-opt multidepot VRP solution.

is a cyclic transfer neighborhood that does not contain a zero-tardiness solution. This result is to be expected, and we must be cautious in interpreting it. It is not uncommon for heuristics to fail to find a zero-valued solution when one exists and, because of this, have unbounded worst-case error ratio. For such problems, the standard error ratio criterion is relatively meaningless (Fisher 1980).

The results of this section easily extend to the uncapacitated case by setting capacities arbitrarily high. They also extend to *O-D* problems by replacing each demand with an *O-D* demand whose origin is located arbitrarily close to the depot, and whose destination is the original demand site.

These worst-case results contrast with the performance ratios derived by other researchers. For example, Frieze, Galbiati and Maffioli (1982) examine a number of heuristics for the asymmetric TSP, including arc-interchange methods for the asymmetric TSP. Most of these have $R_{wc} = \Omega(n)$. For the symmetric case, the best known worst-case ratio is for Christofides' heuristic for which $R_{wc} = (3n - 2)/n$. In addition, Solomon (1986) finds $\Omega(n)$ worst-case performance for a number of VRSPTW algorithms, including arc interchange algorithms. The difference between these $\Omega(n)$ results and our $O(m)$ ratio stems from the fact that we assume the individual tours are optimized after each cyclic transfer.

Our results show that the worst-case performance of cyclic transfer methods is very poor for vehicle routing problems. Fortunately, however, the structure of worst-case problem instances is not typical of practical problems. Our computational results show that cyclic transfers exhibit behavior similar to the λ-opt methods of Lin, and Lin and Kernighan for the TSP, which have unbounded worst-case performance (Papadimitriou and Steiglitz 1978), but which perform quite well in practice (Lin 1965, Johnson, McGeoch and Rothberg 1987).

3. APPROXIMATE CYCLIC TRANSFER NEIGHBORHOOD SEARCH

Three factors cause cyclic transfer neighborhood search to be inherently complex for vehicle routing and scheduling problems. In this section, we discuss these factors, develop approximate search methods to circumvent the difficulties they cause and, finally, present algorithms based on these methods.

The first factor is that each arc cost calculation in G^k involves solving an NP-hard problem. This is because the cost of each arc (ij) in G^k represents the change in objective function due to simultaneously removing demand set j from and adding demand set i to j's current route $I(j)$, and optimally resequencing $I(j)$'s new set of stops, i.e., solving a TSP or related problem on the new set of stops.

The second factor is that the number of arcs in G^k is large. For a VRP with m vehicles, each serving n demands, there are $m(m-1)\binom{n}{k}^2$ arcs in G^k. For fixed k, this quantity reduces to $O(m^2 n^{2k})$ arcs. Since each arc cost computation is itself difficult, the computational effort required to find all arc costs in G^k is unmanageable even if m, n and k are small.

The third complicating factor is that the problem of finding a negative cost cycle through distinct clusters in G^k is itself NP-hard (Thompson and Orlin). Since this search is equivalent to the search for negative cost cyclic transfers, it is unlikely that one can search the neighborhood completely in polynomial time.

Because of these difficulties, we use two methods to approximate the neighborhood search problem. First, we use a polynomial-time approximation for the auxiliary graph arc costs. Second, we search only a restricted subset of the cyclic transfer neighborhood. The combination of these methods reduces the computational requirements to reasonable levels.

The polynomial-time arc cost function approximation proceeds as follows. For each $c_{ij} \in C^k$, remove demand j from its route; then find the minimum cost insertion of demand i into the resulting route without resequencing. We call cyclic transfers that use this approximate cost function *least-cost insertion cyclic transfers*, or LCI cyclic transfers. We say that a solution is *LCI cyclic transfer optimal* (LCI CT-opt) if no LCI cyclic transfer can improve its value. In this respect, the following considerations are important (see Thompson for the details).

1. Any solution which is CT-opt is LCI CT-opt, but not vice-versa.
2. LCI cyclic transfers are a special class of the λ-change procedures of Lin (1965) and Lin and Kernighan (1973) for sufficiently large λ. In particular, the LCI b-cyclic, k-transfer neighborhood of a vehicle routing and scheduling problem is a subset of the $3bkq$-change neighborhood, where q is the number of stops each demand requires (e.g., two stops for demands with pickup and delivery). An implication of this is that if a set of routes is $3bkq$-opt, then it is also LCI b-cyclic, k-transfer optimal.
3. The above considerations characterize LCI cyclic transfers in terms of λ-changes. However, from a practical standpoint, the number of edges involved in these λ-changes is enormous. For example, b, k, and q have lower bounds of 2, 1, and 1, respectively. For these values $\lambda = 6$. By contrast, reasonable computation time requirements limit λ to a maximum of three in practice for fixed-depth λ-change methods. Hence, LCI cyclic transfers search a subset of a very large (and powerful) λ-change neighborhood.
4. The fact that the LCI cyclic transfer neighborhood is a proper subset of the λ-change neighborhood implies a lack of relation between λ-optimality and LCI cyclic transfer optimality. In practice, this can affect route cost estimates, thereby confounding between-route suboptimality and within-route suboptimality. Thus, unless the LCI cyclic transfer algorithm maintains within-route local optimality for individual routes, LCI route cost estimates may range far from their true values.
5. An advantage of the LCI approximation is that it induces some structure into the cost calculations for distance-minimizing problems, which we can then exploit when designing an algorithm. For instance, consider the cyclic k-transfer auxiliary graph for the classical VRP with $k = 1$. Let $T = [0, 1, 2, \ldots, n, n + 1]$ be the sequence of stops for a VRP route, where stops 0 and $n + 1$ both represent the depot. Without loss of generality, let the three cheapest insertions of stop i into T occur after stops r, s, and t, and have costs C_r, C_s, and C_t, respectively, with $C_r \leq C_s \leq C_t$. Then for all $j \in T$, the least cost insertion of i onto $T - j$ occurs after one of r, s, t, and $j - 1$, i.e., $c_{ij} \in \{C_r, C_s, C_t, C_i(j)\}$, where $C_i(j)$ is the cost of inserting i between stops $j - 1$ and $j + 1$ once stop j is removed.

The proof is as follows. If $n \leq 3$, then we are done. Otherwise, there are three cases:

Case 1. ($j \notin \{r, r + 1\}$) Then $c_{ij} = \min\{C_i(j), C_r\}$.

Case 2. ($j = r$) If $j \notin \{s, s + 1\}$, then $c_{ij} = \min\{C_i(j), C_s\}$. Otherwise, $j = r = s + 1$ because, by assumption, r and s are distinct. Then $j \notin \{t, t + 1\}$, because $t \notin \{r, s\}$. Then $c_{ij} = \min\{C_i(j), C_t\}$.

Case 3. ($j = r + 1$) This reduces to case 2 by labeling r as $r + 1$ and vice versa.

This demonstrates that we know, before removing j from its route, that the minimum cost insertion location of i into the resulting route occurs after one of four stops. Furthermore, for each i, three of these specified stops are identical no matter which demand is removed from the route. This is similar to the separable cost situation (Thompson and Orlin), where the costs of adding and removing demands are independent. Because the independence is incomplete in our case, we say that the LCI approach induces *partial separability* into the auxiliary graph cost matrix.

We exploit this property in the following arc cost calculation algorithm for the VRP.

```
For i := 1 to N do
  For veh := 1 to m, veh ≠ I(i) do
    begin
    Determine C_r, C_s, and C_t
    For j := 1 to n_veh do
      begin
      Determine C_i(j)
      If j ∈ {r, r + 1}
      Then if j = s + 1
        Then c_ij = min{C_i(j), C_t}
        else c_ij = min{C_i(j), C_s}
      else c_ij = min{C_i(j), C_r}
      end {for j}
    end {for veh}.
```

Here *veh* indexes routes, m is the number of vehicles, n_{veh} is the number of demands on vehicle *veh*, and the sequence $[1, \ldots, n_{veh}]$ indexes the demands on vehicle *veh*. The overall work required for this algorithm is $O(N^2)$, because each insertion calculation takes $O(1)$ work. This is considerably better than the $O(N^2 n)$ work required by the brute force procedure, which would perform $O(N^2)$ minimum cost insertion calculations each requiring $O(n)$ work.

These results easily generalize to precedence constrained problems and other classes of distance-minimizing problems. However, they do not apply to problems with time-based objectives such as lateness

Table I
Description of Cyclic Transfer Modules

Module	Description
2C1T	2-cyclic 1-transfer
3C1T	3-cyclic 1-transfer
vC1T	Variable depth cyclic 1-transfer starting with 3C1T
2AC1T	2-cyclic 1-transfer with dummy demands
3AC1T	3-cyclic 1-transfer with dummy demands
vAC1T	Variable depth dummy demand cyclic 1-transfer starting with 3AC1T
MC1T	A sequential combination of variable depth and 2-cyclic 1-transfer, which adds dummy demands only to a specified target vehicle
MC1Tx	Better of the initial solution and MC1T applied to the initial solution
2C2T	2-cyclic 2-transfers
vC2T	Variable depth cyclic 2-transfers starting with 2C2T

and tardiness because, for these problems, costs incurred at different points on a vehicle's route are not independent. Nonetheless, special data structures may partially alleviate the computational requirements of arc cost calculations for these problems as well (Thompson).

The second approximation scheme for cyclic transfer neighborhood search involves searching a restricted subset of the cyclic transfer neighborhood. We do this in two ways. First, we limit the initial search to small negative costs cycle, i.e., 2-cycles or 3-cycles in G^k, and then use a variable depth approach to attempt to increase both the cycle length and the cost improvement. Second, we generate and search only part of the graph G^k. This reduces the computation time requirements both for computing auxiliary graph arc costs and for searching the auxiliary graph. In the spirit of Or's (1976) TSP heuristic, rather than examining the transfer of every set of k demands served by each vehicle, we consider transferring only sets of demands that are served adjacently on a vehicle's route. For example, if a vehicle serves the demand set {5, 7, 12, 3, 8} in that order, then G^3 contains the element sets {5, 7, 12}, {7, 12, 3}, and {12, 3, 8}, but no others involving demands 5, 7, 12, 3, or 8. This reduces the number of nodes in G^k from $O(mn^k)$ to $O(mn)$. Moreover, the number of arcs becomes $O(m^2n^2)$, so that the size of G^k is independent of k.

The generic cyclic transfer algorithm is as follows.

STEP 1. Initialize:
Read in data.
Set up data structures.
Set k and b.
Find an initial solution.

STEP 2. Find a λ-opt solution to each route in the current solution.

STEP 3. Set up G^k:
Define nodes and arcs.
Compute arc costs.

STEP 4. While G^k contains a negative cost cycle through b distinct clusters, do begin
If specified, use variable depth procedures to find a cyclic transfer of more negative cost.
Perform the cyclic transfer: update routes.
Find a λ-opt solution to each route in the current solution.
Update G^k: Redefine nodes and arcs as needed.
Update arc costs as needed.
end
Output final routes and STOP.

The modules that we use to find cyclic transfers are listed in Table I. These modules identify and implement negative cost cyclic 1- or 2-transfers involving two, three, or a variable number of routes. The variable depth modules take 3-cyclic transfers as input. In addition, we vary the parameter MAXCYCS. This parameter determines the maximum number of cyclic transfers that each module finds before the overall procedure calls the next module in sequence. Thus, it regulates the "mixing" of different types of cyclic transfers from different modules.

To obtain good initial solutions, to maintain accurate route cost estimates, and to be able to separate the effects of within-route from (between-route) cyclic transfer improvement, we maintain within-route local optimality at all times, as indicated above. We use Lin's 2-opt heuristic and Or's (1976) 3-opt method for the VRP; and Savelsbergh's 2-opt procedure for the VRSPTW.

The main data structures are organized around routes, vehicles, demands, and G^k. A doubly linked list holds all route information. In addition, we use arrays with partial route data when calculating G^k arc costs. Arrays, linked via pointer elements to the route list, contain vehicle and demand information. The cost matrix C^k is stored in a series of 1-dimensional array records, one per demand.

4. COMPUTATIONAL EXPERIENCE

In this section, we present computational results with cyclic transfer algorithms for three vehicle routing and scheduling problems: the classical vehicle routing problem (VRP); a problem with time-windows, precedence constraints and a tardiness-minimizing objective (PCVRSP); and a problem with time window constraints and a hierarchical objective (VRSPTW). Our motivation for choosing these problems is their diversity. This enables a better assessment of the value of cyclic transfers as a general solution strategy for vehicle routing and scheduling problems.

4.1. The Vehicle Routing Problem

The classical VRP involves a homogeneous, capacitated vehicle fleet, a single depot, and a set of weighted demands. The goal is to form a minimum distance set of routes that services all demands. Christofides, Mingozzi and Toth (1979) and Bodin et al. (1983) treat this problem in detail.

To assess the performance of cyclic transfer methods on the VRP, we tested several different sequences of cyclic transfer modules and two values of the parameter MAXCYCS. Table II lists the three most successful sequences, RP1, RP2, and RP3. For each of these, the sequence of modules repeats until no module can find a negative cost cyclic transfer.

We used the standard problems of Eilon, Watson-Gandy and Christofides (1971) described in Table III for testing. We obtained initial tours by determining the minimum fleet size that could carry the given set of weighted demands and then using random demand sequences to form tentative initial routes. If a set of tentative routes exceeded vehicle capacity, then we transferred a few demands at a time among vehicles until capacities were satisfied. Thus, the initial tours were good with respect to vehicle utilization, but poor with respect to distance.

Table III summarizes the best solution values found for each problem, and compares these solutions with the initial solutions and with solutions found by other methods from the literature: the savings method of Clark and Wright (1964), the sweep heuristic of Gillette and Miller (1974), the 3-opt method of Lin (1965), the tree search procedure of Christofides, Mingozzi and Toth (1981), and the generalized assignment heuristic of Fisher and Jaikumar (1981). Cyclic transfers perform nearly as well as the optimization-based generalized assignment heuristic, and uniformly better than the other methods when started from random solutions. Moreover, the magnitude of the percentage improvements over the initial solutions indicates both the poor quality of the starting clusters and the ability of cyclic transfers to improve cluster quality in suboptimal VRP solutions. In this

Table II
Cyclic Transfer Test Sequences

Sequence Number	MAXCYCS	Module Flow Sequence
RP1	∞	(vC2T, 2C2T), (vC1T, 2C1T), (vAC1T, 2AC1T)
RP2	∞	(vC1T, 2C1T), (vAC1T, 2AC1T)
RP3	5	vAC1T, vC1T, 2AC1T, 2C1T
SP1	∞	MC1T, vAC1T, 2AC1T, vC1T, 2C1T
SP2	3	MC1Tx, vAC1T, 2AC1T, vC1T, 2C1T
SP3	∞	MC1Tx, vC1T, 2C1T, vAC1T, 2AC1T
SP4	∞	vAC1T, 2AC1T, vC1T, 2C1T

Table III
VRP Computational Results and Comparison

Solution Method	50 Nodes 5 Vehicles Route Distance	50 Nodes 5 Vehicles % Over CT-OPT	75 Nodes 10 Vehicles Route Distance	75 Nodes 10 Vehicles % Over CT-OPT	100 Nodes 8 Vehicles Route Distance	100 Nodes 8 Vehicles % Over CT-OPT
Initial OR-opt	946	79.5	1,592	85.5	1,579	88.2
CT-OPT	527		858		839	
General Assignment	524	−0.6	857	−0.1	833	−0.7
Tree Search	534	1.3	871	1.5	851	1.4
Sweep	532	0.9	874	1.9	851	1.4
3-opt (best of 3)	556	5.5	876	2.1	863	2.9
Savings	585	11.0	900	4.9	886	5.6
Best CT method	RP1		RP2		RP3	
CT CPU time (min:sec.)	3:08.7		5:35.5		11:29.7	

regard, they can be used to great advantage as part of a two-phase route construction/improvement method.

4.2. A Precedence Constrained Vehicle Routing and Scheduling Problem

Precedence constrained vehicle routing and scheduling problems arise naturally in freight shipping, dial-a-ride, and other practical distribution systems. The distinguishing feature of these problems is that each demand requires pickup and delivery. Normally, the pickup stop must precede the delivery stop on the same route. Bodin et al. review several of these problems.

The specific problem that we consider (the PCVRSP) is the ship routing and scheduling problem studied by Psaraftis et al. (1985). This problem has a nonhomogeneous capacitated fleet of ships initially located at various points in the ocean, and a set of cargoes. Each cargo has specified locations and earliest time restrictions for pickup and delivery. Other complicating constraints exist for this problem as well, e.g., ship-port and ship-cargo incompatibilities. The objective is to minimize the total tardiness of all cargoes where cargoes accrue tardiness if delivered after their desired delivery time. Ships spend time traveling between demand locations, servicing (e.g., loading and unloading) demands, and idling. Idling occurs when a ship arrives at a pickup (delivery) location prior to the earliest allowable pickup (delivery) time for the respective cargo.

To assess the performance of cyclic transfer algorithms on the PCVRSP, we used module sequence SP4 on problem instances with known zero-tardiness solutions from Psaraftis et al., starting with good and poor quality initial solutions. The good routes were constructed with the rolling horizon heuristic of Psaraftis et al. The poor quality routes were constructed from random demand sequences. Table IV describes the features of these instances and summarizes computational results as well.

The magnitude of the percentage improvements for these problems is quite large. We suggest that this is due, at least in part, to the structure of tardiness-minimizing problems. First, moving from a suboptimal solution to a solution with zero or near-zero objective value results in near-100% improvement. Second, reducing or eliminating a small delay in a vehicle schedule can affect the arrival time at many subsequent stops, thereby possibly eliminating or reducing the tardiness of many subsequent deliveries. Thus, small schedule changes can have an enormous effect on overall tardiness. Despite this, it is clear that cyclic transfers perform extremely well, and represent a significant step beyond the existing state-of-the-art in solution methodology for the PCVRSP.

4.3. The Vehicle Routing and Scheduling Problem With Time Windows

Time window constrained problems have received a great deal of attention in recent years (e.g., Golden and Assad 1986, Solomon and Desrosiers 1988). In this subsection, we present computational results for the Vehicle Routing and Scheduling Problem With Time Windows (VRSPTW), which Solomon (1987) treats in detail. This problem has a homogeneous, capacitated vehicle fleet, a single depot, and a set of weighted demands with 2-sided time windows. The windows are hard, meaning that a feasible solution must assign some vehicle to initiate service during the designated time interval. This contrasts with soft time windows, where there is a penalty for initiating service at a time outside of the designated window. Vehicles spend time servicing (e.g., unloading) demands, traveling between demand locations, and idling (waiting for the beginning of a time window).

The objective is hierarchical. The primary goal is to find a set of feasible routes that minimizes the number of vehicles. Savelsbergh shows that, if the number of vehicles is fixed, then finding a set of feasible routes is itself an NP-hard problem. Thus, the first objective is nontrivial. The secondary goal is to minimize route

Table IV
PCVRSP Computational Results

Problem No.	No. of Demands	No. of Vehicles	Initial Routes	Initial Cost	Final Cost	% Improvement	CPU time (Min:Sec.)	
1	20	5	Random	614	126	79.5	9:45	PC-XT
2	50	10	Random	1,643	244	85.1	169:03	PC-XT
3	18	4	Random	2,003	4	99.8	0:11.6	IBM 370
4	18	4	Good	5	2	60.0	0:11.3	IBM 370
5	40	8	Good	52	14	73.1	0:82.6	IBM 370
6	52	9	Good	42	0	100.0	0:83.9	IBM 370
7	52	9	Good	10	0	100.0	0:6.2	IBM 370

completion time, i.e., the sum of travel, idle and service times, where route time begins at time zero. This allows idle time to be incurred at the first customer on each route. The third objective is to minimize the total distance traveled (equivalently, the total travel time). Somewhat counterintuitively, this third objective is equivalent to maximizing idle time. This follows from the second objective and from the fact that service times are invariable.

For computational analysis, we tested modules SP1, SP2, SP3, and SP4 on the six standard problem sets (R1, C1, RC1, R2, C2, and RC2) from Solomon (1987). These sets contain from 7–12 problem instances each with 100 demands. The instances vary in spatial distribution of demand locations (R = random, C = clustered, and RC = semiclustered), vehicle capacity and length of planning horizon (1 = short horizon and low vehicle capacity, 2 = long horizon and high vehicle capacity), time window density (the number of demands with time windows), and time window width. Fleet size requirements for feasible solutions range from 2 to 19 vehicles.

Table V summarizes our computational work. Numerical entries in this table are means over all instances in the corresponding problem sets. We compare SP3 results with the best of SP1, SP2, SP3, and SP4 (best CT), with our implementation of Solomon's (1987) best route construction heuristic (BI1), with the best solutions that Solomon found for each problem instance using BI1 and several other methods as well (Solomon) and, where data are available, with results from the solution improvement methods of Solomon, Baker and Schaffer (SBS) or of Baker and Schaffer (B&S).

We used the BI1 module to construct high-quality initial solutions for the cyclic transfer route improvement steps. This module outputs the best of eight solutions found with Solomon's (1987) sequential insertion heuristic, using different combinations of parameter and initialization criteria. Solomon reports the best average results with this method. Since initial solutions for the B&S and SBS methods were also found using Solomon's sequential insertion algorithm, we expect that they are similar in quality to the SP3

Table V
VRSPTW Computational Results and Comparison

Problem Set	Method	Mean Fleet Size (NV)	Mean Route Time (Rtime)	Mean Route Distance (Distance)	Mean Idle Time (Idle)	Percentage Over SP3				Mean CPU Time (Sec.)
						NV	Rtime	Distance	Idle	
R1	SP3	13.08	2,484	1,367	117					65
	Best CT	13.00	2,455	1,357	98	−0.6	−1.2	−0.7	−16.2	
	BI1	13.50	2,678	1,407	271	3.2	7.8	2.9	131.6	21
	Solomon	13.6	2,696	1,437	259	4.0	8.5	5.1	121.4	
	B&S	13.42	2,489	1,289		2.6	0.2	−5.7		
C1	SP3	10.00	9,965	939	26					31
	Best CT	10.00	9,927	917	10	0.0	−0.4	−2.3	−61.5	
	BI1	10.00	10,169	968	201	0.0	2.0	3.1	673.1	22
	Solomon	10.0	10,104	952	152	0.0	1.4	1.4	484.6	
RC1	SP3	13.00	2,598	1,534	64					61
	Best CT	13.00	2,578	1,514	63	0.0	−0.8	−1.3	−1.6	
	BI1	13.14	2,759	1,592	167	1.1	6.2	3.8	160.9	20
	Solomon	13.5	2,775	1,597	179	3.8	6.8	4.1	179.7	
R2	SP3	3.09	2,333	1,299	34					260
	Best CT	3.09	2,311	1,276	35	0.0	−0.9	−1.8	2.9	
	BI1	3.27	2,607	1,394	212	5.8	11.7	7.3	523.5	65
	Solomon	3.2	2,590	1,449	141	3.6	11.0	11.5	314.7	
	SBS	3.64	2,785	1,239		17.8	19.4	−4.6	−100.0	
C2	SP3	3.00	9,649	648	1					71
	Best CT	3.00	9,645	645	0	0.0	0.0	−0.5	−100.0	
	BI1	3.12	9,781	687	94	4.0	1.4	6.0	9300.0	49
	Solomon	3.0	9,755	712	43	0.0	1.1	9.9	4200.0	
	SBS	3.50	10,159	756		16.7	5.3	16.6		
RC2	SP3	3.71	2,706	1,672	34					140
	Best CT	3.71	2,671	1,634	37	0.0	−1.3	−2.3	8.8	
	BI1	4.00	3,065	1,774	290	7.8	13.3	6.1	752.9	46
	Solomon	3.9	2,955	1,682	273	5.1	9.2	0.6	702.9	

starting solutions. However, they differ slightly because of objective function variations. Moreover, they differ from the Solomon results because these summarize the best solution found for each problem using several different methods, rather than solely reporting the results of his best overall method.

Table V shows that the SP3 cyclic transfer optimal solutions are superior on average to Solomon's (1987) best reported solutions for each problem set. Moreover, cyclic transfer solutions are at least as good in each objective function category for each problem set. Furthermore, idle time averages 80.3% less for SP3, despite the fact that it does not explicitly enter our objective function hierarchy, whereas it does for Solomon's (1987) method. In our case, however, it serves as a surrogate measure of solution quality, because it and travel time together represent the only controllable route variables.

Baker and Schaffer tested a combination of several insertion heuristics together with 2-opt and 3-opt within-route and between-route solution improvement methods on the R1 and part of the C1 problem set. Unfortunately, a comparison of the R1 results from their best method (B&S) with SP3 results is inconclusive because their objective function is substantially different from ours. SP3 dominates in both fleet size and total route time, and therefore in our hierarchical objective function value. However, B&S dominates in a 50-50 weighted combination of route time and route distance, which is their chosen objective function. In fact, of the twelve individual problems, SP3 finds a better solution five times with the weighted objective, and eight times with the hierarchical objective. Because of this, we tentatively conclude that the SP3 and B&S algorithms are comparable in quality, at least for the problem class R1. The C1 results are more definitive, because in each case B&S finds the known optimal solution. However, they report solutions on only three of the nine C1 instances. Thus, it is likely that for clustered problems the B&S algorithm outperforms SP3. Nonetheless, in both cases, further research is needed to provide a direct comparison of the two methods as well as to evaluate the relative performance of B&S on the other problem types.

Solomon, Baker and Schaffer report results of within-route solution improvement procedures (SBS) on the problem sets R2 and C2. Like Baker and Schaffer, their objective function gives equal weight to route time and distance, and ignores fleet size. In their case, however, SP3 solutions are superior under both objective function structures. For these problem sets, cyclic transfer methods clearly dominate.

Comparisons among solutions found by different cyclic transfer module sequences reveal that better initial solutions yield better final solutions, i.e., cyclic transfers are better suited to route improvement than to route construction. Other comparisons demonstrate the effectiveness of the MC1T module in reducing fleet size, and indicate that 3-cyclic transfer neighborhood search is nearly as powerful as variable depth search, when used in conjunction with other modules. In addition, tests on numerous module sequences show the robustness of the cyclic transfer approach: Different sequences commonly find identical or nearly identical solutions. However, certain module sequences dominate in overall performance. In particular, SP3 performs better than the other sequences tested, especially on problems with long scheduling horizons and large vehicle capacities, and on semi-clustered problems. If the scheduling horizon is short and vehicle capacities are low, then SP3 performs at least as well as other methods, unless demands are not clustered. In this case, SP1 dominates. Overall, SP3 finds the best solution found 50.9% of the time, compared with 29.1%, 25.5%, and 23.6% of the time for modules SP1, SP2, and SP4, respectively. Moreover, SP3 uniquely finds the best solution found 40.0% of the time, compared with 18.2%, 14.6%, and 7.3% of the time for the other methods. Besides showing the dominance of SP3, these results indicate that a better solution to a problem instance may sometimes be found by using multiple solution methods.

The CPU times for BI1 and SP3 in Table V are means for each method and problem set, exclusive of I/O, on a 12 MHz IBM PC-AT clone. For individual problems, mean CPU time is 105.0 seconds, median is 68 seconds, and quartiles are 41.5 and 129 seconds. Times range from 4 to 1,220 seconds.

By comparison, times for our implementation of the BI1 route construction method average 37.1 PC-AT seconds. The fact that cyclic transfer CPU times are generally longer does not diminish the usefulness of cyclic transfers: Their ability to improve VRSPTW solutions of very high quality. Moreover, one may stop the iterative improvement phase when a predetermined computation time limit is reached and still have a better solution.

It is somewhat unfair to compare cyclic transfer methods with the best route construction heuristic for the VRSPTW, because this heuristic (BI1) is itself used to find initial solutions for the cyclic transfer solution improvement methods. Thus, the important result of this section is that cyclic transfer methods are able to improve top-quality initial solutions within reasonable CPU times. In addition, cyclic transfers consistently outperform within-route solution improvement procedures. Finally, cyclic transfers and between-route solution improvement methods appear to be comparable, although their relative quality across objective function and problem types remains an open question.

5. CONCLUSION

In this paper, we investigated the application of a new class of neighborhood search algorithms, cyclic transfers, to multivehicle routing and scheduling problems. First, we described the application of cyclic transfers to this class of problems. Then we determined the worst-case performance of these algorithms for several classes of vehicle routing and scheduling problems. Next, we developed computationally efficient methods for finding solution improving cyclic transfers. Finally, we presented computational results for three diverse vehicle routing and scheduling problems, which collectively incorporate a variety of constraint and objective function structures.

Our results show that, despite their poor worst-case performance and their inherent algorithmic complexity, cyclic transfer methods are either comparable to or better than the best published heuristic algorithms for several complex and important vehicle routing and scheduling problems. Moreover, computation times are reasonable, despite the fact that the neighborhood search problem is itself NP-hard. Most importantly, however, they represent a novel approach to solution improvement which shows promise in the fleet planning domain. We believe that this class of local search methods points the way toward further research into neighborhood structures, other than edge-exchange, for vehicle routing and scheduling problems.

ACKNOWLEDGMENT

We gratefully acknowledge Professors James B. Orlin and Amedeo R. Odoni of the Massachusetts Institute of Technology for their valuable comments during the course of this research, and we thank Professor Marius Solomon of Northeastern University for generously providing data sets for the VRSPTW. The comments of two referees and an associate editor were very helpful in improving the presentation and focus of this paper.

This research has been supported by the Office of Naval Research contract N00016-83-K-0220, a United Parcel Service Foundation fellowship, and the Center for Transportation Studies at MIT.

REFERENCES

BAKER, E. K., AND J. R. SCHAFFER. 1986. Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem With Time Window Constraints. *Am. J. Math. Mgmt. Sci.* **6**(3 & 4), 261–300.

BODIN, L. D., B. L. GOLDEN, A. A. ASSAD AND M. BALL. 1983. Routing and Scheduling of Vehicles and Crews: The State of the Art. *Comp. Opns. Res.* **10,** 63–211.

BODIN, L. D., AND T. R. SEXTON. 1983. The Multi-Vehicle Subscriber Dial-A-Ride Problem. Working Paper 82-005, College of Business and Management, University of Maryland at College Park.

CHRISTOFIDES, N., A. MINGOZZI AND P. TOTH. 1979. The Vehicle Routing Problem. In *Combinatorial Optimization*, N. Christofides, A. Mingozzi, P. Toth and C. Sandi (eds.). John Wiley, Bath.

CHRISTOFIDES, N., A. MINGOZZI AND P. TOTH. 1981. Exact Algorithms for the Vehicle Routing Problem Based on Spanning Tree and Shortest Path Relaxations. *Math. Prog.* **20,** 255–282.

CLARK, G., AND J. WRIGHT. 1964. Scheduling of Vehicles From a Central Depot to a Number of Delivery Points. *Opns. Res.* **12,** 568–581.

CROES, G. A. 1958. A Method for Solving Traveling Salesman Problems. *Opns. Res.* **6,** 791–812.

EILON, S., C. D. T. WATSON-GANDY AND N. CHRISTOFIDES. 1971. *Distribution Management: Mathematical Modelling and Practical Analysis.* Hafner Publishing Co., New York.

FISHER, M. L. 1980. Worst Case Analysis of Heuristic Algorithms. *Mgmt. Sci.* **26,** 1–17.

FISHER, M. L., AND R. JAIKUMAR. 1981. A Generalized Assignment Heuristic for Vehicle Routing. *Networks* **11,** 109–124.

FRIEZE, A. M., G. GALBIATI AND F. MAFFIOLI. 1982. On the Worst-Case Performance of Some Algorithms for the Asymmetric Traveling Salesman Problem. *Networks* **12,** 23–39.

GILLETT, B. E., AND L. R. MILLER. 1974. A Heuristic Algorithm for the Vehicle Dispatch Problem. *Opns. Res.* **22,** 340–349.

GOLDEN, B. L., AND A. A. ASSAD. 1986. Perspectives on Vehicle Routing: Exciting New Developments. *Opns. Res.* **34,** 803–810.

GOLDEN, B. L., AND A. A. ASSAD (eds.). 1988. *Vehicle Routing: Methods and Studies.* North-Holland, Amsterdam.

JOHNSON, D. S., L. A. MCGEOCH AND E. E. ROTHBERG. 1987. Near-Optimal Solutions to Very Large Traveling Salesman Problems. Presented at TIMS/ORSA, New Orleans (May).

LAPORTE, G., AND Y. NOBERT. 1987. Exact Algorithms for the Vehicle Routing Problem. In *Surveys in Combinatorial Optimization*, S. Martello et al. (eds.). North-Holland, Amsterdam.

LIN, S. 1965. Computer Solutions to the Traveling Salesman Problem. *Bell Sys. Tech. J.* **44,** 2245–2269.

LIN, S., AND B. W. KERNIGHAN. 1973. An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Opns. Res.* **21,** 498–516.

MAGNANTI, T. L. 1981. Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects. *Networks* **11,** 179–213.

OR, I. 1976. *Traveling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Blood Banking.* Ph.D. Thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Ill.

PAPADIMITRIOU, C. H., AND K. STEIGLITZ. 1978. Some Examples of Difficult Traveling Salesman Problems. *Opns. Res.* **26,** 434–443.

POTVIN, J., G. LAPALME AND J. ROUSSEAU. 1989. A Generalized K-opt Exchange Procedure for the MTSP. *INFOR* **27,** 474–481.

PSARAFTIS, H. N. 1983. K-Interchange Procedures for Local Search in a Precedence-Constrained Routing Problem. *Eur. J. Opnl. Res.* **13,** 391–402.

PSARAFTIS, H. N., J. B. ORLIN, D. BIENSTOCK AND P. M. THOMPSON. 1985. Analysis and Solution Algorithms of Sealift Routing and Scheduling Problems. Working Paper #1700-85, Sloan School of Management, MIT, Cambridge, Mass.

SAVELSBERGH, M. W. P. 1985. Local Search in Routing Problems With Time Windows. *Annals Opns. Res.* **4,** 285–305.

SOLOMON, M. M. 1986. On the Worst-Case Performance of Some Heuristics for the Vehicle Routing and Scheduling Problem With Time Window. *Networks* **16,** 161–174.

SOLOMON, M. M. 1987. Algorithms for the Vehicle Routing and Scheduling Problems With Time Window Constraints. *Opns. Res.* **35,** 254–265.

SOLOMON, M. M., E. K. BAKER AND J. R. SCHAFFER. 1988. Vehicle Routing and Scheduling Problems With Time Window Constraints: Efficient Implementations of Solution Improvement Procedures. In *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad (eds.). North-Holland, Amsterdam.

SOLOMON, M. M., AND J. DESROSIERS. 1988. Time Window Constrained Routing and Scheduling Problems. *Trans. Sci.* **22,** 1–13.

STEWART, W. R. 1987. Accelerated Branch Exchange Heuristics for Symmetric Traveling Salesman Problems. *Networks* **17,** 423–437.

THOMPSON, P. M. 1988. *Local Search Algorithms for Vehicle Routing and Other Combinatorial Problems.* Ph.D. Thesis, MIT, Cambridge, Mass.

THOMPSON, P. M., AND J. B. ORLIN. 1989. Theory of Cyclic Transfers. Working Paper, Operations Research Center, MIT, Cambridge, Mass.

VAN DER BRUGGEN, L. J. J., J. K. LENSTRA AND P. C. SCHUUR. 1991. A Variable Depth Approach for the Single-Vehicle Pickup and Delivery Problem With Time Windows. Presented at TRISTAN I, Montreal, Quebec, Canada (June).