Pergamon

0305-0548(95)00026-7

Computers Ops Res. Vol. 23, No. 3, pp. 229–235, 1996

0305-0548/96 $15.00+0.00

A TABU SEARCH HEURISTIC FOR THE MULTI-DEPOT VEHICLE ROUTING PROBLEM

Jacques Renaud[1,2,3,4]†, Gilbert Laporte[3,5]‡§ and Fayez F. Boctor[2,4]¶

[1]Télé-Université, 2600 boulevard Laurier, Tour de la Cité, 7e étage, Case postale 10700, Sainte-Foy, Canada G1V 4V9, [2]Faculté des Sciences de l'Administration, Université Laval, Sainte-Foy, Canada G1K 7P4, [3]Centre de recherche sur les transports, Université de Montréal, Case postale 6182, succursale Centre-Ville, Montréal, Canada H3C 3J7, [4]Centre de service d'orientation et de recherche sur la compétitivité internationale et l'ingénierie de l'entreprise réseau (SORCIIER), Université Laval, Sainte-Foy, Canada G1K 7P4 and [5]École des Hautes Études Commerciales, 5255, avenue Decelles, Montréal, Canada H3T 1V6

(Received December 1994; in revised form April 1995)

Scope and Purpose—In several physical distribution problems, goods must be delivered from several depots to a set of geographically dispersed customers, under capacity or route length constraints. Documented examples include the delivery of meals, of chemical products, of machines, of industrial gases, of petroleum products, of packaged food, etc. This article describes a powerful heuristic for this difficult problem.

Abstract—This article describes a tabu search algorithm for the multi-depot vehicle routing problem with capacity and route length restrictions. The algorithm is tested on a set of 23 benchmark instances. It is shown to outperform existing heuristics.

1. INTRODUCTION

The purpose of this article is to describe a new heuristic algorithm for the *Multi-Depot Vehicle Routing Problem* (MDVRP) defined as follows. Let $G = (V, E)$ be a graph where V is the vertex set and E is the edge set. The vertex set V is partitioned into two subsets $V_c = \{v_1, \ldots, v_n\}$ and $V_d = \{v_{n+1}, \ldots, v_{n+p}\}$ representing, respectively, the set of *cities* or *customers*, and the set of *depots*. With each city $v_i \in V_c$ is associated a non-negative *demand* q_i and a *service time* δ_i. A cost matrix $C = (c_{ij})$ corresponding to *travel times* is defined on E. We restrict our attention to problems for which C is symmetric and satisfies the triangle inequality, i.e. $c_{ij} = c_{ji}$ for all i, j and $c_{ik} \leqslant c_{ij} + c_{jk}$ for all i, j, k. At each depot $v_{n+k} \in V_d$ are based m_k identical vehicles of capacity Q, where m_k belongs to some interval $[\underline{m}_k, \overline{m}_k]$. Here we assume that $\underline{m}_k = 0$ and $\overline{m}_k = n (k = 1, \ldots, p)$, in other words not all depots are necessarily used. The MDVRP consists of constructing a set of vehicle routes in such a way that: (1) each route starts and ends at the same depot, (2) each customer is visited exactly once by a vehicle, (3) the total demand of each route does not exceed the vehicle capacity Q, (4) the total duration of each route (including travel and service time) does not exceed a preset limit L and (5) the total routing cost is minimized.

The MDVRP is encountered in a large variety of contexts and has considerable economic importance. Documented case studies include the delivery of meals [1], of chemical products [2], of soft drinks [3], of machines [4], of industrial gases [5] of petroleum products [6], of packaged food [7], etc. These studies show that substantial savings can be achieved through the use of optimization techniques. In addition, the MDVRP arises naturally in a family of inventory-routing problems (see,

†Jacques Renaud is Assistant Professor at Télé-Université, Quebec City. This paper is part of his Ph.D. thesis on vehicle routing problems.

‡Gilbert Laporte is Professor at the École des Hautes Études Commerciales, Montreal. His research interests lie in the fields of vehicle routing, location, and scheduling. He is currently the Editor of *Transportation Science*.

§To whom correspondence should be addressed.

¶Fayez F. Boctor is Professor at the Faculty of Administration, Université Laval. His research interests include production and logistics problems.

for example, Dror and Levy [8]). We propose in this article a new heuristic algorithm that has proved highly competitive on a series of test problems.

The remainder of this paper is organized as follows. In Section 2, we briefly review the main exact and approximate algorithms for the MDVRP. Our heuristic is described in Section 3, followed by computational results in Section 4 and the conclusion in Section 5.

2. LITERATURE REVIEW

The MDVRP is NP-hard and very difficult to solve to optimality even for relatively small size instances. For example, when $p = 1$, the problem reduces to the single depot *Vehicle Routing Problem* (VRP) which can rarely be solved to optimality when n exceeds 50 in the symmetric case, and a few hundreds in the asymmetric case [9]. To our knowledge, only two exact algorithms have been proposed for the MDVRP. The first, by Laporte *et al.* [10] formulates symmetric problems as integer linear programs containing degree constraints, subtour elimination constraints, chain barring constraints (to prevent chains of cities linking two different depots), and integrality constraints. The problems are solved by a branch and bound algorithm that initially relaxes the last three types of constraints. Optimal solutions are reported for problem sizes in the range $n \leqslant 50$ and $p \leqslant 8$. Another exact algorithm was proposed by Laporte *et al.* [11] for asymmetric MDVRPs. It first transforms the problem into an equivalent constrained assignment problem. Optimal solutions are then found by a branch and bound algorithm in which the subproblems are assignment problems. Results are reported for instances containing up to 80 cities and 3 depots.

Most existing algorithms for the MDVRP are heuristics. One of the first methods is due to Tillman [12] and uses the Clarke and Wright savings criterion [13]. The algorithm first assigns each customer to its nearest depot and constructs back and forth routes between depots and the customers. These are gradually merged into larger routes using a savings criterion that takes into account the presence of several depots. Tillman and Hering [14] use the same idea within a look-ahead procedure that considers the effect of each potential assignment decision on the next r iterations. Finally, in Tillman and Cain [15] the original Tillman [12] procedure is embedded within a partial enumerative scheme that maximizes a savings criterion. All the above heuristics are constructive and do not use a post-optimization phase. We also note that initial assignments of customers to depots are not reconsidered at later stages.

The next two heuristics construct solutions by means of a sweep procedure. Wren and Holliday [16] present an algorithm that can be applied to single- or multi-depot problems. They first temporarily assign each city to its nearest depot and, using a reference axis, compute its polar angle with respect to this depot. Then all cities are sorted by ascending order of their polar angle. These are then iteratively assigned to new or existing routes, whichever involve the least additional distance. Reallocations to different depots are then made. These authors also present a different way to calculate the polar angle which take into account the configuration of the points around each depot. This new ordering is used to generate four different initial solutions by assigning cities starting from four different positions in the ordered list. The best of these four solutions is chosen as an input to an improvement phase. This latter phase uses seven procedures repeatedly until no improvement can be done. Results are reported on two problems having two depots and up to 176 cities. A second algorithm belonging to this category is due to Gillett and Johnson [17]. Here customers are first assigned to depots to form compact and disjoint clusters. Independent VRPs are solved over the customers assigned to each depot using the sweep algorithm of Gillett and Miller [18]. To improve the solution, the algorithm selects for possible reassignments customers that lie in the region between two depots. Each depot takes turn as an "attracting center" for the customers. VRPs are again solved and the algorithm stops after all depots have served as "attracting centers". The authors present results for the Christofides and Eilon [19] problems and introduce four new 249-city problems having between 2 and 5 depots.

Golden *et al.* [20] propose two approaches for the MDVRP. The first one is based on the method proposed by Yellow [21] for computing savings. In order to limit data storage requirements, they superimpose a grid structure over the problem and only consider joining vertices in adjacent cells of the grid. In the second approach, the MDVRP is solved in two phases. First, customers are assigned to depots, and a separate VRP is then solved for each depot. The following procedure is used for assigning customers to depots. For each $v_i \in V_c$, let $v_{k'_i}$ and $v_{k''_i}$ be closest and second closest depot to v_i, respectively; let $r_i = c_{ik'_i}/c_{ik''_i}$. If r_i does not exceed a certain value λ, then v_i is assigned to depot $v_{k'_i}$; otherwise it is declared a border customer. All border customers are assigned to depots using the first algorithm. A VRP is then solved for each depot and this is followed by an improvement phase. Results are reported for instances containing up to 100 customers and 4 depots.

Raft [22] presents a solution technique that can handle objectives other than route length minimization. This heuristic is a modular algorithm that decomposes the problem into smaller subproblems. The algorithm starts with a route assignment phase. After having estimated the number of vehicles needed, the algorithm constructs clusters of customers, each assigned to one vehicle. These clusters are not assigned to any depot and are constructed to provide a small expected length. In the next phase, each route is assigned to a depot. Lin's [23] 2-opt exchange procedure is then applied to each route. Test results on six 249-city problems indicate that this algorithm is comparable to that of Gillett and Johnson [1].

Finally, Chao *et al.* [24] describe a multi-phase heuristic. Customers are first assigned to their closest depot. A VRP is then solved for each depot using the modified savings algorithm of Golden *et al.* [20]. This solution is then improved by moving customers to different depots. The authors use the "record-to-record" improvement approach of Dueck [25] which allows deteriorations of the current solution. Two reinitialization procedures are also used to diversify the search. Results are presented for the eleven benchmark problems of Gillett and Johnson [17] and for twelve new instances containing up to 360 cities and 9 depots.

We note in closing this section the contribution of Li and Simchi-Levi [26] who present a worst-case analysis of tour partitioning heuristics for the capacitated MDVRP, and that of Min *et al.* [27] who describe a three-phase heuristic for the MDVRP with backhauls.

3. ALGORITHM

The algorithm we have developed is based on tabu search. This global optimization metaheuristic was initially proposed by Glover [28]. It consists of exploring the search space by moving from a solution to its best neighbour, even if this results in a deterioration of the objective function value. This way, the likelihood of moving out of local optima is increased. To avoid cycling, solutions that were recently examined are forbidden or declared *tabu* for a certain number of iterations. *Intensification* strategies can be applied to accentuate the search in a promising region of the solution space. In contrast, *diversification* can be used to broaden the search to less explored region. Our algorithm contains two parts: (1) construction of an initial solution and (2) tabu search.

3.1. Construction of an initial solution

Initially each vertex is assigned to its nearest depot and a VRP heuristic is applied to the vertex set of each depot. For this, we use the *Improved Petal* heuristic of Renaud *et al.* [29]: in this algorithm, a set of routes that can either be supplied by one or two vehicles are first generated, and the best route selection is made by solving a set partitioning problem. As the generation of two-vehicle routes can sometimes be time consuming, it is only used if the problem at hand is quite large and thus more likely to be difficult, i.e. if $n/m \geqslant 50$. Otherwise, only one-vehicle routes are generated.

3.2. Tabu search

Tabu search makes up the core of our algorithm called FIND. It consists of three phases: **F**ast improvement, **IN**tensification, and **D**iversification. Each of these phases uses some or all of the three basic procedures 1-route, 2-route and 3-route which we now describe.

1-route. This procedure is used as a post-optimizer on single vehicle routes. It consists of applying the 4-opt* improvement mechanism developed by the authors for the Traveling Salesman Problem [30]. This algorithm attempts to improve a Hamiltonian circuit by using eight of all potential 4-opt moves [23]. Compared with the standard 3-opt exchange procedure, 4-opt* is only 0.95% worse, but uses only 1% of the execution time.

2-route. Solution improvements can often be obtained by moving vertices belonging to two different routes, assigned to one or two depots. Let $(v_{i_h}, v_{j_h}, v_{k_h}, v_{l_h})$ be two sequences of four consecutive vertices (possibly including a depot) in each of two routes $h = 1$ and 2. Then the following six moves are attempted, as long as a depot is not moved and feasibility is maintained: (1) insert v_{j_1} between v_{i_2} and v_{j_2}; (2) insert v_{j_2} between v_{i_1} and v_{j_1}; (3) swap v_{j_1} and v_{j_2}; (4) insert (v_{j_1}, v_{k_1}) between v_{i_2} and v_{j_2}; (5) insert (v_{j_2}, v_{k_2}) between v_{i_1} and v_{j_1}; (6) swap (v_{j_1}, v_{k_1}) and (v_{j_2}, v_{k_2}). These six moves are a subset of the larger family of moves considered within the λ-interchange procedure proposed by Osman [31]. We found experimentally that very little quality is lost but much time is gained by concentrating on this restricted subset of six moves.

3-route. When routes are near full capacity, it may not be possible to improve a solution using the 2-route mechanism. Instead, we propose an exchange scheme involving three routes. For routes $h = 1$ and 2, consider a vertex $v_{i_h} \in V_c$; for routes $h = 2$ and 3, consider the sequences of two vertices (v_{r_h}, v_{s_h}) where $v_{r_2} \neq v_{i_2}$ and $v_{s_2} \neq v_{i_2}$. Then the following combination of moves is attempted as long as feasibility is maintained: insert v_{i_1} between v_{r_2} and v_{s_2}; insert v_{i_2} between v_{r_3} and v_{s_3}.

Throughout FIND, the incumbent and its value are recorded, but as we allow deteriorations of the objective function, the current solution is not necessarily the best known. Whenever a vertex is moved from its current route, moving this vertex back into the same route is declared tabu for θ iterations, where θ is randomly chosen in [4, 10]. Random tabu durations help avoid cycling. Their use was first proposed by Taillard [32] in the context of the Quadratic Assignment Problem and they have since been applied to the Vehicle Routing Problem [33]. As is common in tabu search, we use an *aspiration criterion*, namely, a tabu status may be overridden if implementing the corresponding move yields a better incumbent. The three phases of FIND are now described.

3.2.1. Fast improvement. In this phase, the algorithm attempts to improve upon the incumbent by repeatedly applying the following three steps:

— inter-depot: apply 2-route exchanges between routes of two different depots;
— intra-depot: apply 2-route exchanges between routes of the same depot;
— 3-route: exchange vertices between three routes.

This sequence is repeated until the incumbent does not improve for θ_1 consecutive iterations. The value of θ_1 is taken as 75. It was determined experimentally as a good compromise after testing a range of values.

For each of the three steps, any move that yields an improvement is immediately implemented. Otherwise, the best non-tabu deteriorating move is implemented. Whenever a move is implemented, the 1-route procedure is applied to all routes involved in the move.

The selection of routes to which 2-route and 3-route are applied is done as follows. Denote by m_i the number of routes associated with depot v_i. To define the distance between a route and a depot or between two routes, each route is represented by its center of gravity. In inter-depot, we consider exchanges between each depot v_i and the $\lfloor p/2 \rfloor + 1$ depots closest to it. For each pair of depots v_i and v_j, we consider exchanges between the $\lceil m_i/2 \rceil$ routes of v_i closest to v_j and the $\lceil m_j/2 \rceil$ routes of v_j closest to v_i. In intra-depot, we consider all pairs of routes for each depot. In 3-route, the three routes h_1, h_2 and h_3 are selected as follows: all routes are considered for h_1; h_2 is the closest neighbour of route h_1 and h_3 is the closest neighbour of route h_2, with $h_3 \neq h_1$.

3.2.2. Intensification. This phase intensifies the search for better routes, starting with the best known solution and working on one depot at the time. It applies the intra-depot step to each depot in turn until no improvement to the incumbent has been produced for θ_2 consecutive iterations. The value of θ_2 is taken as 300.

3.2.3. Diversification. In this phase, the following two steps are repeated 20 times. First, we seek the best reinsertion of a vertex from its current route into a route belonging to a different depot. Choosing the same vertex for reinsertion is prohibited for the next ten applications of this step. Second, the inter-depot and intra-depot steps of the fast improvement phase are applied for θ_3 consecutive iterations without improvement to the solution values obtained in the first step. We use $\theta_3 = 50$. Here, the length of the interval during which a move is tabu is randomly chosen in [15, 20] and no aspiration criterion is used. The effect of the diversification phase is to perform a broader exploration of the solution space.

4. COMPUTATIONAL RESULTS

The algorithm was tested on the 11 classical problems described by Christofides and Eilon [19] and by Gillett and Johnson [17], and on the 12 new problems of Chao *et al.* [24]. The main characteristics of

Table 1. Characteristics of test problems

Problem	Source	p	n	Q	L
1	CE	4	50	80	∞
2	CE	4	50	160	∞
3	CE	5	75	140	∞
4	CE	2	100	100	∞
5	CE	2	100	200	∞
6	CE	3	100	100	∞
7	CE	4	100	100	∞
8	GJ	2	249	500	310
9	GJ	3	249	500	310
10	GJ	4	249	500	310
11	GJ	5	249	500	310
12	CGW	2	80	60	∞
13	CGW	2	80	60	200
14	CGW	2	80	60	180
15	CGW	4	160	60	∞
16	CGW	4	160	60	200
17	CGW	4	160	60	180
18	CGW	6	240	60	∞
19	CGW	6	240	60	200
20	CGW	6	240	60	180
21	CGW	9	360	60	∞
22	CGW	9	360	60	200
23	CGW	9	360	60	180

CE, Christofides and Eilon [19]; GJ, Gillett and Johnson [17]; CGW, Chao *et al.* [24].

Table 2. Depot coordinates for problems 1–7

Problem	Depot coordinates
1–2	(20,20), (30,40), (50,30), (60,50)
3	(40,40), (50,22), (55,55), (25,45), (20,20)
4	(35,20), (35,50)
5	(15,35), (55,35)
6	(15,20), (50,20), (35,55)
7	(15,35), (55,35), (35,20), (35,50)

the test problems are summarized in Table 1. For each problem, the travel time matrix coincides with the distance matrix and there are no service times at the vertices. All computations were performed with full real precision and reported solution values were rounded up or down after the second decimal. Since depot locations for problems 1 to 7 are not provided in the original article, we give them in Table 2. For problem 8, the location of customer 245 must be changed to (70, −80) [24].

We present in Table 3 the solution values and computational times obtained by Gillett and Johnson [17], Chao *et al.* [24] and the FIND algorithm. In the latter case, we provide the solution cost and the cumulative running time after each of the three phases of the tabu search: fast improvement, intensification and diversification. Best solution values are indicated by **bold** numbers.

Results show that the Gillett and Johnson algorithm is clearly dominated by that of Chao, Golden and Wasil (CGW) and by FIND. We will therefore focus on the last two algorithms. Over the 23 instances, FIND obtains a best solution value 20 times, and the same solution as CGW in one case. Substantial improvements are obtained on some of the larger instances (10, 11, 15 and 23). In 16 cases (1, 4, and 10 to 23), the fast improvement phase alone is sufficient to produce a best solution value and this number goes up to 19 after the intensification phase. Computing times after these two phases are very modest and would seem comparable to those obtained by CGW. The diversification phase is rather time consuming, but it improves the incumbent 12 times out of 23. Exact time comparisons between CGW and FIND are difficult to make because of the different computers involved, but it is very likely that the full FIND heuristic is slower than CGW.

Results presented in Table 3 correspond to "standard" versions of CGW and FIND run with a prefixed set of parameters. However, each of these two algorithms did on occasions produce even better solutions, using different parameters. The best solution values produced by CGW and FIND are given in Table 4. For each of the problems, the best known solution is obtained with FIND. These solutions are fully described in [34].

Table 3. Summary of computational results

	Gillett and Johnson	Chao, Golden and Wasil (CGW)		FIND algorithm					
				Fast improvement		Intensification		Diversification	
	Cost	Cost	Time*	Cost	Time†	Cost	Time†	Cost	Time†
1	593.2	582.4	1.1	**576.86**	0.5	**576.86**	0.6	**576.86**	3.2
2	486.2	**476.6**	1.2	484.30	1.4	484.30	1.5	476.66	4.8
3	652.4	**641.2**	1.8	650.42	0.6	650.42	0.8	645.14	5.8
4	1066.7	1026.9	2.2	**1025.09**	1.3	**1022.98**	2.6	**1016.13**	11.4
5	778.9	756.6	2.4	784.76	6.2	775.70	7.1	**754.20**	12.8
6	912.2	883.6	2.1	886.29	0.6	**878.31**	1.7	**876.50**	8.4
7	939.5	898.5	4.8	905.77	1.2	905.75	1.6	**897.86**	6.8
8	4832.0	4511.6	24.1	4576.61	6.4	**4500.48**	27.5	**4500.48**	69.4
9	4219.7	**3950.9**	20.9	4122.70	5.4	4071.01	13.1	3669.31	41.2
10	3822.0	3815.6	7.2	**3804.92**	8.6	**3754.52**	13.5	**3720.88**	43.0
11	3754.1	3733.0	16.7	**3727.70**	7.1	**3696.37**	10.3	3670.25	36.4
12		1327.3	2.8	**1318.95**	1.0	**1318.95**	1.4	**1318.95**	5.4
13		1345.9	0.7	**1318.95**	1.1	**1318.95**	1.4	**1318.95**	4.8
14		1372.5	1.3	**1365.68**	1.1	**1365.68**	1.1	**1365.68**	2.6
15		2610.3	2.3	**2551.45**	4.1	**2551.45**	5.0	**2551.45**	15.5
16		2605.3	6.1	**2572.23**	2.9	**2572.23**	3.6	**2572.23**	11.1
17		2816.6	6.5	**2731.37**	2.3	**2731.37**	2.9	**2731.37**	5.8
18		3877.4	8.6	**3814.62**	3.9	**3814.62**	5.2	**3789.96**	23.2
19		3863.9	22.3	**3828.61**	4.5	**3828.61**	5.6	**3827.06**	22.0
20		4272.0	14.6	**4097.05**	3.6	**4097.05**	4.5	**4097.05**	10.0
21		5791.5	78.5	**5678.50**	17.8	**5678.50**	19.9	**5678.50**	48.7
22		5857.4	132.4	**5718.00**	6.0	**5718.00**	7.7	**5718.00**	33.5
23		6494.6	24.4	**6145.58**	5.7	**6145.58**	7.2	**6145.58**	17.3

*Minutes on a Sun 4/370.
†Minutes on a Sun Sparcstation 10 (cumulative time).

Table 4. Best solutions values obtained by CGW and FIND

Problem	CGW	FIND
1	**576.9**	**576.86**
2	474.6	**473.53**
3	**641.2**	**641.18**
4	1012.0	**1003.86**
5	756.5	**750.26**
6	879.1	**876.50**
7	893.8	**892.58**
8	4511.6	**4485.08**
9	3950.9	**3937.81**
10	3727.1	**3669.38**
11	3670.2	**3648.94**
12	1327.3	**1318.95**
13	1345.9	**1318.95**
14	1372.5*	**1365.68**
15	2610.3	**2551.45**
16	2605.3	**2572.23**
17	2816.6*	**2731.37**
18	3877.4	**3781.03**
19	3863.9	**3827.06**
20	4272.0*	**4097.06**
21	5791.5	**5656.46**
22	5857.4	**5718.00**
23	6494.6*	**6145.58**

*For problems 14, 17, 20 and 23, Chao, Golden and Wasil obtained better "manual solution values" of 1365.7, 2731.4, 4097.1 and 6145.6, respectively

5. CONCLUSION

We have described a new tabu search heuristic for the multi-depot vehicle routing problem. On 23 benchmark problems, the standard version of the proposed heuristic produces a best solution 20 times. Moreover, using various parameters, the algorithm generates "best solutions" 23 times out of 23. Finally, it is interesting to observe that the truncated version of the algorithm (obtained by stopping after the intensification phase) is very fast and highly competitive: it generates a best solution 19 times out of 23.

Acknowledgements—This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under grants OGP0036509 and OGP0039862. This support is gratefully acknowledged. Thanks are also due to two referees for their valuable comments.

REFERENCES

1. P. J. Cassidy and H. S. Bennett, TRAMP—a multi-depot vehicle scheduling system. *Opl Res. Q.* **23,** 151–162 (1972).
2. M. O. Ball, B. L. Golden, A. A. Assad and L. D. Bodin, Planning for truck fleet size in the presence of a common-carrier option. *Decis. Sci.* **14,** 103–120 (1983).
3. B. L. Golden and E. Wasil, Computerized vehicle routing in the soft drink industry. *Opns Res.* **35,** 6–17 (1987).
4. M. L. Fisher, R. Greenfield, R. Jaikumar and J. Lester, A computerized vehicle routing application. *Interfaces* **12** (4), 42–52 (1982).
5. W. Bell, L. Dalberto, M. L. Fisher, A. Greenfield, R. Jaikumar, P. Kedia, R. Mack and P. Prutzman, Improving the distribution of industrial gases with an on-line computerized routing and scheduling system. *Interfaces* **13,** 4–22 (1983).
6. G. Brown, C. Ellis, G. Graves and D. Ronen, Real-time wide area dispatch of mobil tank trucks. *Interfaces* **17** (1)107–120 (1987).
7. J. Pooley, Integrated production and distribution facility planning at Ault Foods. *Interfaces* **24,** 113–121 (1994).
8. M. Dror and L. Levy, A vehicle routing improvement algorithm-comparison of a "greedy" and a "matching" implementation for inventory routing. *Computers Opns Res.* **23,** 33–45 (1986).
9. G. Laporte, The vehicle routing problem: an overview of exact and approximate algorithms. *Eur. J. Opl Res.* **59,** 345–358 (1992).
10. G. Laporte, Y. Nobert and D. Arpin, Optimal solutions to capacitated multidepot vehicle routing problems. *Congressus Numerantium* **44,** 283–292 (1984).
11. G. Laporte, Y. Nobert and S. Taillefer, Solving a family of multi-depot vehicle routing and location-routing problems. *Transp. Sci.* **22,** 161–172 (1988).
12. F. A. Tillman, The multiple terminal delivery problem with probabilistic demands. *Transp. Sci.* **3,** 192–204 (1969).
13. G. Clarke and J. W. Wright, Scheduling of vehicles from a central depot to a number of delivery points. *Opns Res.* **46,** 93–100 (1964).
14. F. A. Tillman and R. W. Hering, A study of a look-ahead procedure for solving the multiterminal delivery problem. *Transp. Res.* **5,** 225–229 (1971).
15. F. A. Tillman and T. M. Cain, An upperbound algorithm for the single and multiple terminal delivery problem. *Mgmt Sci.* **18,** 664–682 (1972).
16. A. Wren and A. Holliday, Computer scheduling of vehicles from one or more depots to a number of delivery points. *Opns Res. Q.* **23,** 333–344 (1972).
17. B. E. Gillett and J. G. Johnson, Multi-terminal vehicle-dispatch algorithm. *Omega* **4,** 711–718.
18. B. E. Gillett and L. E. Miller, A heuristic algorithm for the vehicle-dispatch problem *Opns Res.* **22,** 340–349 (1974).
19. N. Christofides and S. Eilon, An algorithm for one vehicle-dispatching problem. *Opl Res. Q.* **20,** 309–318 (1969).
20. B. L. Golden, T. L. Magnanti and H. Q. Nguyen, Implementing vehicle routing algorithms. *Networks* **7,** 113–148 (1977).
21. P. Yellow, A Computational modification to the savings method of vehicle scheduling. *Opl Res. Q.* **21,** 281–283 (1970).
22. O. M. Raft, A modular algorithm for an extended vehicle scheduling problem. *Eur. J. Opl Res.* **11,** 67–76 (1982).
23. S. Lin, Computer solution of the traveling salesman problem. *Bell Syst. Comput. J.* **44,** 2245–2269 (1965).
24. I. M. Chao, B. L. Golden and E. Wasil, A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. *Am. J. Math. Mgmt Sci.* **13,** 371–406 (1993).
25. G. Dueck, New optimization heuristics, the great deluge algorithm and the record-to-record travel. Scientific center technical report, IBM Germany, Heidelberg Scientific Center (1990).
26. C.-L. Li and D. Simchi-Levi, Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems. *ORSA J. Comput.* **2,** 64–73 (1990).
27. H. Min, J. Current and D. Schilling, The multiple depot vehicle routing problem with backhauling. *J. Business Logist.* **13,** 259–288 (1992).
28. F. Glover, Heuristic for integer programming using surrogate constraints. *Decision Sci.* **8,** 156–166 (1977).
29. J. Renaud, F. F. Boctor and G. Laporte, An improved petal heuristic for the vehicle routing problem. *J. Opl Res. Soc.* In press.
30. J. Renaud, F. F. Boctor and G. Laporte, A fast composite heuristic for the symmetric traveling salesman problem. Working paper CRT-981, Centre for research on transportation, Montreal (1994).
31. I. H. Osman, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annls Opns Res.* **41,** 421–451 (1993).
32. É. Taillard, Robust taboo search for the quadratic assignment problem. *Parallel Comput.* **17,** 433–445 (1991).
33. M. Gendreau, A. Hertz and G. Laporte, A tabu search heuristic for the vehicle routing problem. *Mgmt Sci.* **40,** 1276–1290 (1994).
34. J. Renaud, G. Laporte and F. F. Boctor, A tabu search heuristic for the multi-depot vehicle routing problem. Working paper CRT-94-44, Centre for research on transportation, Montreal (1994).