ELSEVIER

European Journal of Operational Research 93 (1996) 121–134

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

A genetic algorithm for service level based vehicle scheduling

Charles J. Malmborg

Department of Decision Sciences and Engineering Systems, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA

Received March 1995; revised June 1995

Abstract

In many practical applications, vehicle scheduling problems involve more complex evaluation criteria than simple distance or travel time minimization. Scheduling to minimize delays between the accumulation and delivery of correspondence represents a class of vehicle scheduling problems, where: the evaluation of candidate solutions is costly, there are no efficient schemes for evaluation of partial solutions or perturbations to existing solutions, and dimensionality is limiting even for problems with relatively few locations. Several features of genetic algorithms (GA's) suggest that they may have advantages relative to alternative heuristic solution algorithms for such problems. These include ease of implementation through efficient coding of solution alternatives, simultaneous emphasis on global as well as local search, and the use of randomization in the search process. In addition, a GA may realize advantages usually associated with interactive methods by replicating the positive attributes of existing solutions in the search process, without explicitly defining or measuring these attributes. This study investigates these potential advantages through application of a GA to a service level based vehicle scheduling problem. The procedure is demonstrated for a vehicle scheduling problem with 15 locations where the objective is to minimize the time between the accumulation of correspondence at each location and delivery to destination locations. The results suggest that genetic algorithms can be effective for finding good quality scheduling solutions with only limited search of the solution space.

Keywords: Vehicle scheduling; Routing; Traveling salesman problem; Unit periods of waiting; Heuristic solution procedure; Genetic algorithm

1. Introduction

Considerable research has been devoted to solving vehicle routing and scheduling problems based on deterministic distance and/or travel time criteria. Recent surveys of this work are presented in Bodin et al. (1983), Christofides (1985), Golden and Assad (1988), Lawler et al. (1985), and Laporte and Nobert (1987). Considerably less work has been devoted to vehicle scheduling problems with stochastic demand and more complex evaluation criteria (Jaillet and Odoni, 1988; Bertsimas, 1992; Bertsimas and Van Ryzin, 1993). In some vehicle scheduling applications, complex evaluation criteria preclude the development of a closed form problem representation and limit the effectiveness of solution techniques based on structural insights. In some cases, such problems have been successfully addressed using interactive decision support systems (Bell et al. 1983; Brown and Graves, 1981; Malmborg and Lutley, 1989) to generate 'good quality' solutions. Alternatively, local search techniques such as multi-start descent meth-

SSDI 0377-2217(95)00185-9

ods using simulated annealing, tabu search, and other neighborhood exploration methods have been successfully applied in some difficult applications (Thompson and Psaraftis, 1993). Despite their flexibility and relative efficiency, decision support systems and local search techniques tend to require extensive manual intervention and/or reevaluation of candidate solutions. Although these disadvantages are sometimes offset by efficient schemes for evaluating perturbations of existing solutions, this is not practical in all cases.

One area where candidate solutions are costly to evaluate, the potential for user insights is limited, and there are no efficient schemes for evaluating perturbations to existing solutions involves constrained vehicle scheduling to maximize service level. A specific example of this problem involves the scheduling of vehicles to minimize the unit periods of waiting associated with material transfers. In this case, a vehicle or other resource serves the material flow requirements between a collection of workcenters over a fixed operating period. The objective is to minimize the delay between the time that material accumulates at each workcenter, and the time that it is delivered to its destination workcenter. In some cases, the problem has the added feature that the delivery of all accumulated material must be completed during or between operating periods. Practical examples include scheduling automated guided vehicles for mail delivery and scheduling bank couriers for check retrieval and processing (Malmborg and Simons, 1989). For even a small number of workcenters, these problems have the complicating features described above. An effective solution strategy requires a computationally efficient procedure that is not prone to trapping at local minima, does not rely extensively on user intervention, and is relatively easy to implement.

The purpose of this study is to evaluate the potential of genetic algorithms (see Goldberg, 1988; Davis, 1987) for solving vehicle scheduling problems based on a unit periods of waiting criterion. The obvious advantages of genetic algorithms (GA's) for this problem include ease of implementation, an emphasis on global as well as local search, and the use of randomization in the search process. In addition, GA's feature some of the advantages of interactive methods in that they seek to replicate the positive attributes of existing solutions in the search for improved solutions without explicitly defining or measuring these attributes. To investigate these potential advantages, a difficult vehicle scheduling problem based on maximizing the level of service provided by a vehicle in performing material transfers between workcenters is introduced. Based on the special structure of this problem, a two phase GA is designed and implemented for the test problem.

The next section presents background information on the vehicle scheduling problem and discusses related literature. Section 3 formally defines the problem and describes the information base associated with it. This includes the formulation of the unit periods of waiting criterion relative to the definition of a schedule alternative. In Section 4, a genetic algorithm based on the structure of the vehicle scheduling problem is described. The algorithm includes a two phase procedure for finding the terminal routing to assure delivery of all accumulated material at the end of the operating period and a schedule building phase prior to the end of the operating period. Computational experience with the algorithm for a test problem with 15 workcenters is described. The final section offers a summary and conclusions.

2. Background information

Vehicle routing and scheduling represents an important class of problems on which a vast and growing literature is focused. Recent surveys of this research presented in Bodin et al. (1983), Christofides (1985), Golden and Assad (1988) and Lawler et al., (1985) provide several hundreds of references. The majority of these studies deal with vehicle routing problems where the objective criterion is the minimization of the distance or time traveled by vehicles in executing a schedule. Other variations of the problem that have received considerable attention by researchers include vehicle scheduling for fleet size minimization, (Li and Simchi-Levi, 1990, 1992), and distance or capacity constrained problems, (Haimovich and Rinnooy Kan, 1985; Altinkemer and Gavish, 1987, 1990; Assad, 1988). Vehicle scheduling techniques based on distance minimization have many practical applications. However, no rigorous mathematical analysis is known for even the simplest variation of the problem despite the many optimization algorithms reported in the literature (see Laporte, Nobert and Taillefer, 1988, for a survey of optimal procedures). Examples of specific optimization algorithms for distance based problems are presented in Christofides and Eilon (1969), Christofides et al. (1981a,b), Laporte et al. (1985) and many other studies. Although computational results reported in such studies generally focus on problems with fewer than 50 destinations, Fisher (1994) cites rapidly decreasing computation costs, improved accuracy of data through road network databases, and the growing body of vehicle scheduling methodologies as hopeful signs for the future utility of optimal vehicle scheduling algorithms. As a demonstration of this potential, he presents an efficient optimal procedure for capacitated, multi-vehicle problems based on the distance minimization criterion. The procedure is tested on fifteen problems from the literature, (each having approximately 100 destinations), with the results suggesting that a large problem can be solved in just a few hundred minutes on a workstation.

Optimal procedures have also been suggested for somewhat more complex vehicle scheduling problems such as those with time windows. Time windows represent a variation of the vehicle scheduling problem where the service time for a customer is restricted to a fixed time window. Surveys of advances in exact procedures for solving problems with time windows are presented by Solomon and Desrosiers (1988) and Dumas et al. (1991). More recently, Desrochers et al. (1992) present a new optimization procedure based on linear relaxations of the set partitioning formulation of the time windows problem. The linear relaxations are solved by column generation and used as lower bounds in a branch and bound procedure. Previously, this procedure was used by Haouari et al. (1990) to optimally solve two other variations of the vehicle routing problem including fleet size minimization with time windows, and the multidepot problem with time windows.

Apart from optimal strategies, there is a vast literature on heuristic methods for solving vehicle scheduling problems (see Haimovich et al., 1988, for a recent survey). Most practical applications have relied on the use of heuristic methods. An important variation of these methods is neighborhood search procedures. These procedures have been successfully applied in many large scale single vehicle and multiple vehicle applications. Most single vehicle scheduling procedures have focused on constrained or unconstrained single traveling salesman problems (see for example Stewart, 1987, Psaraftis, 1983, and Savelsbergh, 1985). More recently, Thompson and Psaraftis (1993) have extended neighborhood search procedures to the multivehicle case. These procedures, known as cyclic transfer algorithms, first assign demands to vehicles and then route each vehicle to its assigned stops. Thompson and Psaraftis present computational results on the use of neighborhood search algorithms for three diverse vehicle routing and scheduling problems including distance minimization, time windows, and precedence constrained problems (i.e., each demand requires pickup and delivery such as in freight shipping or dial-a-ride). Their results show that cyclic transfer methods are either comparable or better than the best published heuristic methods for several important variations of the problem.

Recently, several heuristic approaches to vehicle routing and scheduling have focused on vehicle scheduling problems with stochastic demands (Jaillet and Odoni, 1988). Stewart and Golden (1983) and Dror and Trudeau (1986) use techniques from stochastic programming to optimally solve small problems based on the use of bounding procedures. The idea of using an apriori sequence for the solution of TSP's when instances are modified probabilistically was introduced in Jaillet (1988). More recently, Bertsimas (1992) reports on a heuristic procedure to solve a variation of the problem where a single vehicle of limited capacity must meet demands at n fixed locations, returning periodically to the depot to empty its contents. The objective is to minimize the total distance traveled. The demand at each location is unknown at the time that the tour is designed but is assumed to follow a known probability distribution. Other examples of the same problem reported in the literature are presented in the context of a bank messenger problem (Malmborg and Simons, 1989), a 'hot meals' delivery system (Bartholdi et al., 1983) and routing of forklifts in a warehouse (Bertsimas, 1992).

Despite the considerable research devoted to vehicle routing problems with time windows and stochastic demands, generally much less research has focused on more difficult variations of vehicle scheduling problems. In many practical applications the objective of minimizing travel distance is not necessarily paramount. Rather, the delivery time (wait for service) is a more appropriate objective. Consider the following utility repair example as described in Bertsimas and Van Ryzin (1993):

> "A utility firm is responsible for the maintenance of a large, geographically dispersed facilities network that is subject to random failures. A fleet of repair vehicles is dispatched from a depot to respond to failures. Vehicle crews spend a random amount of time servicing each call before moving on to the next one. The objective is to minimize the average downtime due to failures".

Bertsimas and Van Ryzin (1993) suggest several other similar applications where the objective is to minimize a combination of delivery cost and average wait for delivery, including home heating oil deliveries, mail order firms, automobile distribution by truck, freight consolidation, and parcel post systems. Other researchers have proposed models that incorporate stochastic or dynamic demands and congestion/waiting time measures. Batta, Larson and Odoni (1988) and Berman et al. (1990) consider congestion effects in the context of location problems. Bertsimas and Van Ryzin (1991) present a more detailed review of such models and introduce a model for dynamic vehicle scheduling where an uncapacitated vehicle traveling in an Euclidean region must service demands whose time of arrival, location and on-site service are stochastic. Brown and Graves (1991) present a strategy for dynamically revising vehicle schedules using a rolling planning horizon. Malmborg and Simons (1989) presents a formulation for measuring the 'lateness cost' of a fixed multivehicle bank courier schedule where demands are stochastic.

In the present study, a single vehicle generalization of the courier scheduling problem described in Malmborg and Lutley (1989) is considered where the objective is to minimize the expected unit periods of waiting (UPW). An application of this problem involves scheduling a vehicle for delivery of mail between destinations where the objective is to minimize the expected UPW between the accumulation and delivery of all mail arriving during a fixed operating period. It is assumed that there is a single uncapacitated vehicle, individual destinations have correspondence with all other destinations, there are no layovers (where the vehicle remains idle at a specific location), and that any combination of vehicle routings is possible as long as all arriving mail is picked up and delivered. As with the model described in Malmborg and Lutley (1989), the need to completely process mail over an operating period requires the vehicle to function beyond the operating period to complete a final pickup and delivery at each destination. The essential difference between the model described in Malmborg and Lutley and the one described in this study, is that all destinations in the current problem are both pick up and delivery locations whereas destinations were either pick up *or* delivery locations in the previous study. In the next section, the definition of this problem is formalized and a model for measuring the expected UPW is presented.

3. Model description

The information base for the vehicle scheduling problem consists of expected accumulation volumes at each destination, the probability distribution of correspondence for each destination, and travel/processing times for the delivery vehicle. Notation for describing this data and vehicle scheduling alternatives is summarized below:

n = The number of destinations served by the vehicle.

T = The length of the operating period in time units.

v_{it} = The expected volume of correspondence accumulating at workcenter i during time period t, for $i = 1,\dots,N$ and $t = 1,\dots,T$.

p_{ij} = The probability that a unit of correspondence accumulating at workcenter i is destined for workcenter j, for $i, j = 1,\dots,n$.

$t(i, j)$ = The vehicle travel time between workcenters i and j, for $i, j = 1,\dots,n$ (and including loading and unloading).

X = A vehicle scheduling alternative comprised of a sequence of s stops, $x_1, x_2,\dots,x_s$, where x_j denotes the destination of the j-th stop on schedule X with $1 \leqslant x_j \leqslant n$, and for $j = 1,\dots,s$.

k_i = The number of departures from destination i (under the current scheduling alternative).

d_{ik} = The rank ordered time of the k-th stop at destination i, for $i = 1, \dots, n$ and $k = 1, \dots, k_i$, where $d_{i0} \equiv 0$ and $d_{ih} < d_{ij}$ for $h < j$.

For a given scheduling alternative, the assumption of no layovers makes it possible to compute the d_{ik} values directly from X using the $t(i, j)$ travel times. Using these values, it is possible to compute the expected volume of correspondence for destination j that is retrieved on the k-th stop at destination i as

$$\sum_{t=d_{i,k-1}}^{d_{ik}} v_{it} p_{ij}.$$

It follows that the expected UPW for this correspondence *prior* to pickup is given by

$$\sum_{t=d_{i,k-1}}^{d_{ik}} v_{it}(d_{ik} - t).$$

The travel time before this correspondence is delivered to destination j can be obtained by the difference between d_{ik} and the first departure time from destination j following d_{ik}:

$$\min_m \{ d_{jm} - d_{ik} > 0 \}.$$

Prior to pickup at destination i, the UPW accumulated for this same correspondence can be estimated using

$$\sum_{t=d_{i,k-1}}^{d_{ik}} v_{it} p_{ij}(d_{ik} - t).$$

Since it is assumed that all correspondence accumulating during the operating period must be picked up and delivered, it is necessary to include a final pick up and delivery at each destination after the end of the operating period. This requires that

$$d_{i,ki-1} \geqslant T$$

for $i = 1, \dots, n$ and

$$d_{i,ki-1} < d_{j,kj}$$

for $i \neq j$ and $i, j = 1, \dots, n$.

Using the above definitions, minimizing the total UPW associated with scheduling the vehicle is equivalent to solving:

(Problem P1)

Minimize $f(X)$

$$= \sum_{i=1}^{n} \sum_{k=1}^{d_i - 1} \sum_{j=1}^{n} \left(\sum_{t=d_{i,k-1}}^{d_{ik}} v_{it} p_{ij} \right) \min_m \{ d_{jm} - d_{ik} > 0 \}$$
$$+ \sum_{t=d_{i,k-1}}^{d_{ik}} v_{it} p_{ij}(d_{ik} - t)$$
$$+ \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\sum_{t=d_{i,ki-2}}^{d_{i,ki-1}} v_{it} p_{ij} \right) (d_{j,kj} - d_{i,ki-1})$$

subject to

$d_{i,ki-1} \geqslant T$ for $i = 1, \dots, n$,

$d_{i,ki-1} < d_{j,kj}$ for $i \neq j$ and $i, j = 1, \dots, n$.

The next section describes a solution strategy for P1 based on the use of a genetic algorithm.

4. Solution strategy

Depending on the operating period length, the objective function of problem P1 can represent a costly calculation for even small to moderate sized problems. Further, since changes in the vehicle routing impose revision of downstream departure times, schemes for reevaluating adjusted solutions are only efficient for perturbations occurring late in a schedule. Neighborhood and sequential search procedures are limited by the fact that partial solutions cannot be evaluated for unit periods of waiting. These characteristics suggest that procedures based primarily on a neighborhood search concept are unlikely to yield an efficient solution strategy. Genetic algorithms provide an alternative strategy that would not be affected by these limitations. Therefore, this technique was selected for the purpose of solving the vehicle scheduling problem. The specific algorithm developed for this purpose proceeds in two phases. In the first phase, a sequence n 'tours' denoted $Q_1, Q_2, \dots, Q_n$ are generated where

$Q_i = \{ q_{i1}, q_{i2}, \dots, q_{in} \}$,

$q_{i1} = 1$,

and

$1 \leqslant q_{ij} \leqslant n,$

for $i, j = 1, \dots, n$. These tours are obtained by generating 'good quality' solutions to the travel time minimizing TSP for each starting destination. Once these tours are generated, an initial population of M schedules, $\{X_1, \dots, X_M\}$, is generated by executing Procedure 1 in the Appendix M times.

As detailed in the Appendix, Procedure 1 randomly generates an initial population of M schedules using 'splices' of tours, $Q_1, \dots, Q_n$, until no more stops can be added within the operating period. That is, random splices of TSP optimal tours are laid in an end-to-end fashion until the vehicle is fully scheduled over the operating period. After this, the final pickup and delivery are added using the TSP optimal sequence of destinations. Procedure 1 guarantees that the M initial solutions will be feasible with respect to problem P1.

The second phase of the solution procedure embodies the use of a GA to mate solutions to produce subsequent generations of improved solutions. To apply this procedure, the objective function of problem P1 is computed to obtain the values, $f(X_1), \dots, f(X_M)$, and Procedure 2 (described in the Appendix) is executed to obtain G subsequent generations of improved solutions. The logic of Procedure 2 is to select vehicle schedules for mating and then to combine sequential splices of random length from the two mating schedules. Through the G_1 and G_2 parameters, the user specifies the maximum number of destinations that can be included in each random splice. The procedure alternates between the two mating schedules in selecting the random splices of destinations to add to the offspring vehicle schedule. As the process continues, the time required to visit the destinations included on the partial schedule constructed from random splices of the two mating destinations approaches the length of the operating period. When this occurs, the random schedule building process terminates and the resultant partial schedule is appended with two TSP optimal sequences of the n destinations in order to complete the final pickup and delivery at each destination. The TSP optimal sequences used for the final pickup and delivery are conditional on the last destination visited during the operating period.

Procedure 2 utilizes the techniques of natural selection, reproduction and mutation that characterize genetic algorithms. The process of natural selection is executed through the computation of mating probabilities within each generation based on the UPW associated with schedules in the current population. Step 1 assures that schedules with lower UPW values will have higher mating probabilities. Once mating schedules are selected through the sampling process in Step 2, the reproductive process described in Steps 4–8 is executed. In effect, the reproductive or 'crossover' step successively joins crossections of each schedule with the size of these sections varying between the limits of G_1 and G_2. In general, the value of G_2 will be less than $2s$ in order to assure that both mating schedules are reflected in the offspring of the process. In effect, the smaller the values of G_1 and G_2, the greater the 'randomness' of the crossover between the mating schedules. If G_1 and G_2 are small, the process will alternate frequently between the two mating schedules in generating and combining small crossections of each. This will result in an offspring schedule that is quite different from the parent schedules. If G_1 and G_2 are large, the process will only combine a few large crossections of each mating schedule and result in an offspring schedule that retains more characteristics of the parent schedules. Mutations can occur as each stop is added to the offspring schedule. As shown in Steps 5 and 7 of Procedure 2 (and as suggested for best results in Goldberg, 1988), the likelihood of such mutations is on the order of one in ten thousand.

In summary, the two phase algorithm proceeds by assuming that each individual destination is the starting point and generating n 'good quality' solutions to the corresponding travel time minimizing TSP's. An initial population of M schedules is then generated by combining random length crossections of the TSP solutions until the vehicle is scheduled over the full operating period. Each schedule in the initial population is appended with the final pick up and delivery based on the TSP optimal sequence to assure distribution of all accumulated material. Subsequent generations of schedules are obtained by randomly combining pairs of schedules using Procedure 2. In the next section, the two phase algorithm is illustrated for a sample problem.

5. Illustration through a sample problem

To illustrate the two phase algorithm, a sample problem with $n = 15$ workcenters, $T = 240$ minutes, and the travel times, item volumes and correspondence probabilities illustrated in Table 1(a)–(c) is used. Using this data for an average vehicle schedule, each reevaluation of the of the unit periods of waiting consumed approximately 55 seconds of CPU on a Pentium 90 based computer. To execute the algorithm, it was necessary to generate good quality solutions to the travel minimizing TSP's where each of the 15 destinations is assumed to be the starting point. These solutions were obtained by executing the 'N opt' procedure for $N = 1, 2, 3, 4, 5$, and then selecting the solution that yielded the lowest travel time for a given starting destination. The results of this procedure for 15 TSP's associated with the sample problem are illustrated in Table 1(d). To illustrate the potential effects of computational parameters G_1 and G_2 using only a Pentium 90 based computer, a population size of $M = 10$ was selected. (As discussed later in this section, much larger values of M would be needed to justify broad general-

Table 1
Sample problem parameters
(a) Travel/service times between destinations in minutes (symmetric)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	–	7.3	5.4	6.3	3.9	1.0	11.2	6.1	0.6	16.6	3.2	2.0	8.1	1.7	2.1
2		–	6.7	13.8	4.2	7.2	6.2	6.0	7.1	10.1	10.1	6.9	7.8	5.8	7.4
3			–	4.9	4.2	5.6	4.2	3.2	5.1	6.6	8.8	3.9	2.2	5.5	6.8
4				–	10.9	4.7	9.0	11.0	7.2	12.5	8.2	7.4	4.2	7.4	6.8
5					–	4.4	10.5	4.3	3.5	16.5	6.3	3.2	7.1	2.7	4.3
6						–	10.5	6.2	1.5	14.7	1.9	2.7	7.9	2.4	1.8
7							–	8.2	11.2	1.5	16.2	9.8	3.8	10.8	12.3
8								–	5.7	14.1	11.7	3.9	4.8	6.2	8.1
9									–	16.9	4.1	1.5	7.9	1.3	2.8
10										–	24.9	15.0	5.4	15.3	17.4
11											–	5.8	12.0	2.0	0.4
12												–	6.7	2.5	4.1
13													–	8.0	9.4
14														–	1.3
15															–

Table 1 (continued)
(b) Vectors of p_{ij} correspondence probabilities (listed in sequential order for $i = 1, \ldots, 15$)

$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$	$j=9$	$j=10$	$j=11$	$j=12$	$j=13$	$j=14$	$j=15$
0.063	0.027	0.044	0.118	0.117	0.092	0.067	0.003	0.063	0.054	0.096	0.057	0.082	0.062	0.055
0.053	0.089	0.086	0.113	0.072	0.069	0.072	0.004	0.073	0.130	0.010	0.136	0.008	0.046	0.039
0.033	0.040	0.045	0.068	0.031	0.095	0.007	0.078	0.093	0.033	0.098	0.121	0.042	0.115	0.101
0.075	0.045	0.139	0.031	0.057	0.036	0.083	0.017	0.054	0.061	0.149	0.058	0.038	0.139	0.019
0.095	0.021	0.005	0.017	0.118	0.059	0.046	0.061	0.117	0.001	0.146	0.018	0.100	0.089	0.106
0.018	0.034	0.099	0.046	0.010	0.139	0.134	0.046	0.017	0.129	0.022	0.139	0.082	0.061	0.025
0.100	0.021	0.090	0.035	0.055	0.014	0.069	0.000	0.101	0.082	0.038	0.116	0.101	0.076	0.102
0.085	0.020	0.096	0.020	0.072	0.067	0.077	0.079	0.027	0.062	0.088	0.091	0.112	0.036	0.066
0.032	0.088	0.125	0.030	0.081	0.053	0.070	0.137	0.112	0.084	0.077	0.000	0.046	0.027	0.038
0.098	0.126	0.155	0.035	0.043	0.030	0.146	0.033	0.016	0.085	0.098	0.029	0.034	0.049	0.023
0.102	0.042	0.085	0.050	0.001	0.095	0.025	0.099	0.011	0.037	0.051	0.096	0.099	0.105	0.102
0.026	0.130	0.017	0.064	0.024	0.077	0.106	0.044	0.072	0.001	0.026	0.121	0.088	0.084	0.118
0.039	0.075	0.153	0.078	0.041	0.104	0.004	0.025	0.098	0.100	0.105	0.046	0.022	0.046	0.063

Table 1 (continued)
(c) Average correspondence volumes in items per hour [a]

Destination	Hour 1	2	3	4
1	89	50	79	22
2	50	37	16	75
3	15	36	51	41
4	52	87	23	42
5	12	48	48	10
6	62	35	30	74
7	54	34	20	79
8	53	78	53	72
9	20	59	38	29
10	43	49	63	44
11	21	47	76	36
12	22	56	50	22
13	84	14	97	56
14	54	75	15	50
15	43	51	88	38

[a] Items per minutes are obtained for each minute as the product of the corresponding hourly accumulation and 1/60.

Table 1 (continued)
(d) Optimal TSP sequence for each starting solution

Starting destination	
1	1-9-14-15-11-6-12-3-8-5-2-7-10-13-4
2	2-5-14-15-11-6-1-9-12-3-7-10-13-4-8
3	3-12-9-14-15-11-6-1-5-2-7-10-13-4-8
4	4-13-3-8-12-9-1-6-15-11-14-5-2-7-10
5	5-14-9-1-12-15-11-6-3-8-13-4-10-7-2
6	6-1-9-12-3-8-2-5-14-11-15-4-7-10-13
7	7-3-12-9-1-6-11-15-14-5-2-8-4-13-10
8	8-12-9-1-6-11-15-14-5-3-4-13-10-7-2
9	9-1-6-11-15-14-12-3-8-5-2-7-10-13-4
10	10-7-3-12-9-1-6-11-15-14-5-2-8-13-4
11	11-15-14-6-1-9-12-3-8-5-2-7-10-13-4
12	12-9-1-6-11-15-14-5-3-8-13-4-10-7-2
13	13-3-8-12-9-1-6-15-11-14-5-2-7-10-4
14	14-9-1-12-6-11-15-5-3-8-13-4-10-7-2
15	15-11-6-1-9-14-12-3-8-5-2-7-10-13-4

izations concerning the parameters of the algorithm.) Procedure 1 was executed to generate the first generation of schedules. The UPW values associated with the resultant ten schedules are summarized in Table 2 where the average UPW is 185 115 and the minimum equal to 169 276.

In order to gain a perspective on the quality of the initial population of solutions generated through Procedure 1, a total of 5000 random schedules were generated and evaluated. Random schedules were generated by selecting destinations at random until the end of the operating period and then appending each schedule with the final pickup and delivery using the TSP optimal sequence based on the final destination during the operating period. The 5000 schedules generated through this procedure had an average UPW value of 911 216 with a minimum value of 662 192 and a maximum value of 1 300 370. Fig. 1 shows the frequency histogram of UPW values obtained through this procedure which required approximately 80 hours of CPU on a Pentium 90 based computer. As would be expected, Procedure 1 generates schedules that are significantly better than arbitrarily generated schedules.

The next step was to compare the quality of results generated through Procedure 2 to those obtained using Procedure 1. This comparison was based on generating a population of 100 schedules using Procedure 1. These 100 schedules were compared to 100 schedules that included the initial population of 10 schedules obtained via Procedure 1 and nine subsequent generations of schedules obtained through Procedure 2. The average UPW values from 100 schedules generated through Procedure 1 was 192 642 with a minimum value of 161 514 and a maximum value of 264 115. Fig. 2 summarizes the frequency histogram of UPW values describing the 100 Procedure 1 schedules.

Using the ten schedules defining the initial population, nine subsequent generations of schedules were generated using Procedure 2. This process was re-

Table 2
Expected unit periods of waiting for ten schedules generated using procedure 1

Schedule #	Expected unit periods of waiting
1	212619
2	171421
3	169276
4	190004
5	200338
6	181078
7	194726
8	172561
9	185134
10	182915

Fig. 1. Frequency histogram for a sample of 5000 randomly generated schedules.

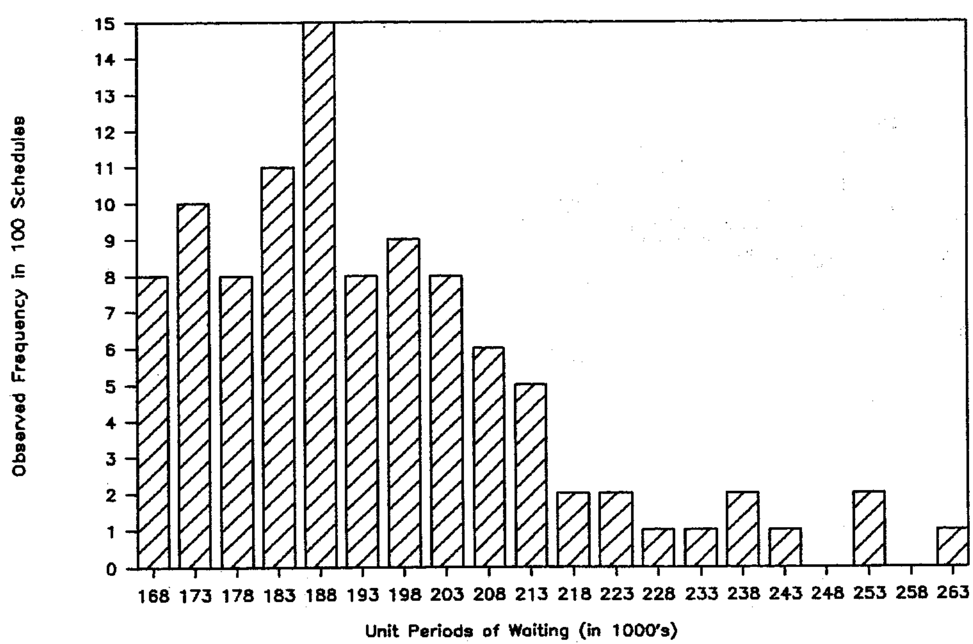

Fig. 2. Frequency histogram of 100 schedules obtained from Procedure 1.

Table 3
Summary of Procedure 2 performance for alternative combinations of crossover parameters G_1 and G_2. In all combinations, $G_1 = 15$

Generation	$G_2 = 30$		$G_2 = 45$		$G_2 = 60$		$G_2 = 75$		$G_2 = 90$		$G_2 = 105$	
	Inc. opt. val. [a]	Av. val. [b]	Inc. opt. val.	Av. val.	Inc. opt. val.	Av. val.	Inc. opt. val.	Av. val.	Inc. opt. val.	Av. val.	Inc. opt. val.	Av. val.
1	169276	185115	169276	185115	169276	185115	169276	185115	169276	185115	169276	185115
2	160696	184553	167532	185095	157642	195567	169276	188373	153071	181459	146624	185992
3	164750	189644	157871	199017	157642	190098	168971	186807	153071	194880	146624	189884
4	149136	173883	153634	190451	157642	192625	168971	191727	148230	172671	146624	173503
5	149136	169153	153634	206330	157642	185992	168971	186465	148230	162820	146624	172860
6	149136	171954	153634	183336	157642	182752	168971	189567	145075	161909	146624	171745
7	149136	166015	153634	187337	157642	181919	168971	179016	145075	167497	146624	166856
8	149136	169551	153634	180941	157642	182100	168971	180142	144238	164487	146624	170474
9	149136	168687	153634	183760	157642	183695	168971	181340	144238	159866	146624	172673
10	149136	169861	153634	175350	157642	181453	168971	181644	144238	158547	146624	172900

[a] Incumbent optimal value.
[b] Average value.

peated for the six combinations of crossover parameters, G_1 and G_2, summarized in Table 3 where G_1 is held constant at 15 and G_2 is increased by increments of 15 from 30 to 105. (The value of G_1 is held constant at $G_1 = n$ in order to capture travel time efficiencies inherent in the initial population of schedules as new schedules are generated from crossections of previous generation schedules.) Table 3 also summarizes the average and incumbent optimal UPW values obtained in each generation for each crossover parameter combination. As Table 3 shows, the highest quality schedule generated using

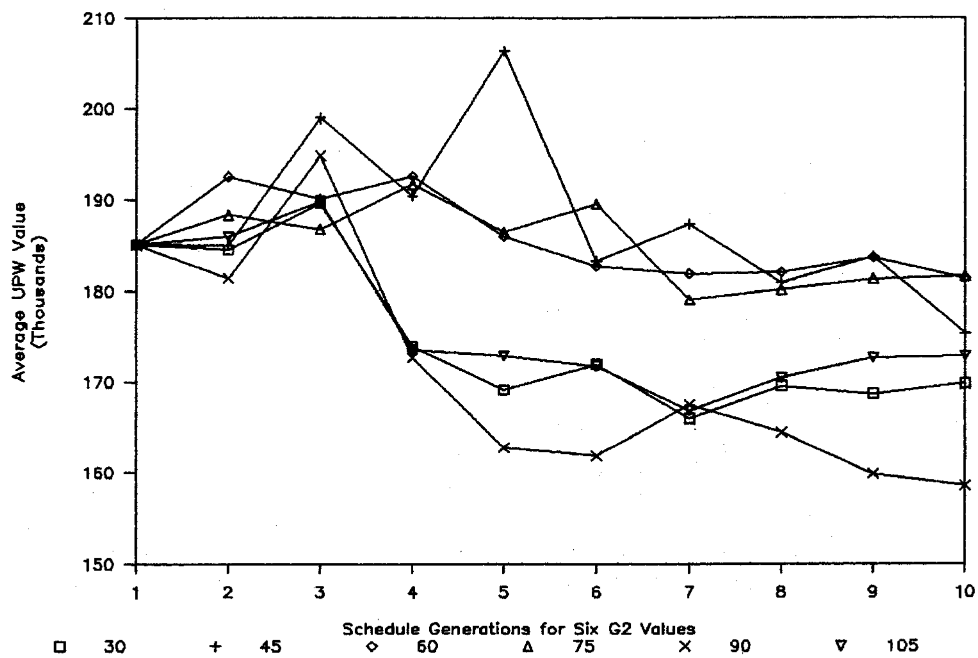

Fig. 3. Average UPW each generation, for G_2 values from 30 to 105.

Procedure 2 had an expected UPW of 144238. The actual sequence of locations in this schedule is given by the 120 stops defined by the destination sequence: 8-12-9-14-5-1-15-4-13-3-8-5-14-15-11-6-1-9-12-2-5-14-15-11-6-4-13-3-8-5-14-15-11-1-9-14-15-11-6-12-3-13-10-7-2-8-12-9-14-5-1-15-4-13-3-8-5-14-15-11-6-1-9-12-2-5-14-15-11-6-4-13-3-8-5-14-15-11-1-9-14-15-11-6-12-3-13-10-7-2-8-12-9-14-5-1-15-11-6-4-3-2-7-10-13-8-12-9-14-5-1-15-11-6-4-3-2-7-10-13.

As Table 3 suggests, convergence to the local optimal occurred within just four generations for all values except $G_2 = 90$. In terms of the quality of the optimal solution, the procedure performed best for the extreme values of G_2. That is, the highest quality optimal solutions were obtained using the minimum value of $G_2 = 30$ (UPW = 149136), the maximum value of $G_2 = 105$ (UPW = 146624), and the second largest value of $G_2 = 90$ (UPW = 144238). The performance of Procedure 2 as measured by the quality of the optimal solution deteriorates as the value of G_2 is increased from 30 to 75 where the lowest quality optimal solution is observed (UPW = 168971). Performance then improves as G_2 is increased from 75. In all cases, Procedure 2 produces higher quality optimal solutions than those obtained from 100 iterations of Procedure 1 alone. Both approaches consumed approximately 90 minutes of CPU on a Pentium 90 based computer in order to evaluate 100 candidate schedules.

To gain further insight to the performance of Procedure 2, a summary the performance of Procedure 2 relative to the value of G_2 is shown in Fig. 3 which plots the *average* UPW value in each generation for the six values of G_2. The results in Fig. 3 generally reinforce the results in Table 3. However, being based on average values, the Fig. 3 results are less influenced by the random nature of the search process. The combined results from Fig. 3 and Table 3 suggest that the performance of Procedure 2 for values of G_2 other than $G_2 = 90$ does not vary significantly. Average values for UPW range between 169000 and 182000 for all other values of G_2 while the average UPW value after 10 generations with $G_2 = 90$ is 158547. Furthermore, Fig. 3 suggests that Procedure 2 converges more uniformly for $G_2 = 90$ than for other values of G_2. A possible explanation for this result is that larger crossections of mating schedules are more effective at preserving positive performance attributes in offspring schedules while smaller crossections fail to retain the efficiencies inherent in the mating schedules. Crossections that are too large may fail to fully utilize the positive attributes of both schedules in the mating process. However, it should also be noted that such a small population is known to converge rapidly and not necessarily optimally. Therefore, even though the results do suggest that the user must carefully weigh the issue of crossover parameter size relative to the application problem when calibrating the algorithm, it is also clear that further studies based on larger population sizes are needed to resolve this issue.

6. Summary and conclusions

A service level based vehicle scheduling problem associated with the minimization of the unit periods of waiting between the accumulation and delivery of correspondence between workcenters has been described. Since alternative schedules are costly to evaluate and there are no efficient schemes for evaluation of partial solutions, techniques requiring extensive reevaluations of complete and/or partial schedules do not provide a practical solution strategy. Based on the unique structure of the unit periods of waiting criterion and the comparative advantages of GA's relative to this problem, a two phase algorithm has been proposed for efficiently generating 'good quality' solutions. The results from the study are based on a limited population size of $M = 10$ and are therefore tentative in nature. Nonetheless, they do suggest that good quality solutions can be obtained with minimal search of the solution space using the procedure and that performance of the algorithm is probably sensitive to the specification of crossover parameters. These findings suggest that investigation into the relationship between the performance of the procedure, crossover parameters, population size, mutation probability, and other parameters of the process is a fruitful area for continuing study using larger population sizes. In addition, analytical research into the development of 'tight' lower bounds on UPW values could yield more definitive assessment of the performance of GA's and other heuristic procedures for the service level based vehicle scheduling problem. Finally, development of customized GA's for the multivehicle case and other important variations of service level based vehicle scheduling problems would be a fruitful area for continued research. These areas are the focus of continuing research.

Appendix

Procedure 1. Generation of the initial population of vehicle schedules (all uniform random numbers between 1 and n are constrained to be integer).

Step 0. Set $h = 0$ and go to Step 1.

Step 1. Generate a $U(1, n)$ integer,
set $h = h + 1$,
set $x_k = U(1, n)$,
set $g = 0$ and
set $k = 1$.
Go to Step 2.

Step 2. Generate $z = U(1, n)$ and set
$x_{k+1} = q_{xk,1}$,
$j = 1$, and
$t = t + t(x_k, x_k + 1)$.
If $t \geqslant T$, then STOP,
otherwise proceed to Step 3.

Step 3. Set
$j = j + 1$,
$x_{k+j} = q_{xk,j}$, and
$t = t + t(x_k, x_k + 1)$.
If $t \geqslant T$, then set $k = k + j$ and go to Step 5,
otherwise, proceed to Step 4.

Step 4. If $j < z$, then go to Step 3,
otherwise set $k = k + z$ and go to Step 2.

Step 5. Add the final pick up and delivery to the schedule by setting
$x_{k+1} = q_{xk,2}$,
$x_{k+2} = q_{xk,3}, \dots, x_{k+n-1} = q_{xk,n}$,
$x_{k+n} = q_{xk,1}$,
$x_{k+n+1} = q_{xk,2}, \dots, x_{k+2n} = q_{xk,n}$,
and set $s = k + 2n$.
Go to Step 6.

Step 6. Fix schedule
$x_h = \{x_{h1}, \dots, x_{hsh}\} = \{x_1, \dots, x_s\}$
and set $h = h + 1$.
If $h > M$, then STOP,
otherwise go to Step 1.

Procedure 2. Computation of G generations of improved vehicle schedules.

Step 0. Set $g = 1$, $m = 1$ and go to Step 1.

Step 1. Generate the vector of mating probabilities,
$P = \{p_1, \dots, p_M\}$,
for the current generation of schedules, where

$$p_i = \frac{(f(X_i))^{-1}}{\Sigma_{j=1}^{M}\left\{(f(X_i))^{-1}\right\}} \quad \text{for } i = 1, \dots, M.$$

Set the upper and lower limits of the sampling interval for schedule i equal to
$L_i = \Sigma_{k=1}^{i-1} p_k$ and
$U_i = \Sigma_{k=1}^{i} p_k$
for $i = 1, \dots, M$, and go to Step 2.

Step 2. Generate $z_1 = U(0, 1)$ and select schedule y for mating such that
$L_y < z_1 \leqslant U_y$.
Generate $z_2 = U(0, 1)$ and select schedule w for mating such that
$L_w < z_2 \leqslant U_w$.
If $y = w$, then continue to regenerate z_2 and w until $y \neq w$.
Go to Step 3.

Step 3. Set
$k = 1$, $\phi_1 = 1$, and
$\phi_2 = 1$, $c = y$, $j = 1$, $t = 0$.

Step 4. Generate $z = U(G_1, G_2)$. [1]
If $c = y$, then go to Step 5.
If $c = w$, then go to Step 7.

Step 5. Generate $z_1 = U(0, 1)$.
If $z_1 < 0.0001$, then set
$x'_k = x_{w,\phi_1}$,
$k = k + 1$,
$j = j + 1$ and
$\phi_1 = \phi_1 + 1$.
If $z_1 \geqslant 0.0001$, then set
$x'_k = x_{y,\phi_1}$,
$k = k + 1$,
$j = j + 1$ and
$\phi_1 = \phi_1 + 1$.
If $\phi_1 > S_y$, then set $\phi_1 = 1$.
If $k > 1$, then set
$t = t + t(x'_{k-1}, x'_k)$.

[1] Uniform random numbers between G_1 and G_2 are constrained to be integer.

If $t \geqslant T$, then go to Step 9,
otherwise proceed to Step 6.

Step 6. If $j \leqslant z$, then go to Step 5.
Otherwise, set $c = w$, $j = 1$, and go to Step 4.

Step 7. Generate $z_1 = U(0, 1)$.
If $z_1 < 0.0001$, then set
$x'_k = x_{y,\phi_2}$,
$k = k + 1$,
$j = j + 1$ and
$\phi_2 = \phi_2 + 1$.
If $z_1 \geqslant 0.0001$, then set
$x'_k = x_{w,\phi_2}$,
$k = k + 1$,
$j = j + 1$ and
$\phi_2 = \phi_2 + 1$.
If $\phi_2 > s_w$, then set $\phi_2 = 1$.
If $k > 1$, then set
$t = t + t\,(x'_{k-1}, x'_k)$.
If $t \geqslant T$, then go to Step 9,
otherwise proceed to Step 8.

Step 8. If $j \leqslant z$, then go to Step 7.
Otherwise, set $c = y$, $j = 1$, and go to Step 4.

Step 9. Add the final pick up and delivery to the schedule by setting
$x'_{k+1} = q_{xk,2}$,
$x'_{k+2} = q_{xk,3}, \dots, x'_{k+n-1} = q_{xk,n}$,
$x'_{k+n} = q_{xk,1}$,
$x'_{k+n+1} = q_{xk,2}, \dots, x'_{k+2n} = q_{xk,n}$,
and set $s = k + 2n$.
Go to Step 10.

Step 10. Save the schedule just generated,
$X'_m = \{x'_{m1}, \dots, x'_{msm}\} = \{x'_1, \dots, x'_s\}_2$
and set $m = m + 1$.
If $m < M$, then go to Step 2,
otherwise go to Step 11.

Step 11. Record the current generation of schedules and objective function values. Install the next generation of schedules as:
$X_h = \{x_{h1}, \dots, x_{hsh}\} = \{x'_{h1}, \dots, x'_{hs}\}$,
for $h = 1, \dots, M$.
Recompute $f(X_1), \dots, f(X_n)$,
set $m = 1$ and $g = g + 1$.
If $g < G$, then go to Step 1,
otherwise STOP.

Acknowledgement

The author gratefully acknowledges the inputs of the reviewers which have improved the paper.

References

Altinkemer, K., and Gavish, B. (1987), "Heuristics for unequal weight delivery problems with a fixed error guarantee", *Operations Research Letters* 6, 149–158.

Altinkemer, K., and Gavish, B. (1990), "Heuristics for equal weight delivery problems with constant error guarantees", *Transportation Science* 24, 294–297.

Assad, A.A. (1988), "Modeling and implementation issues in vehicle routing", in: B.L. Golden and A.A. Assad (eds.), *Vehicle Routing: Methods and Studies*, Elsevier Science Publishers, Amsterdam, 745.

Bartholdi, J.J., Platzman, L.K., Lee, C.R., and Warden, W.W. (1983), "A minimal technology routing system for meals on wheels", *Interfaces* 13, 1–8.

Batta, R., Larson, R.C., and Odoni, A.R. (1988), "A single server priority queuing location model", *Networks* 18, 87–103.

Bell, W., Dalberto, L., Fisher, M., Greenfield, A., Jaikumar, R., Kedia, P., Mack, R., and Prutzman, P. (1983), "Improving the distribution of industrial gases with an online computerized routing and scheduling optimizer", *Interfaces* 13, 9–23.

Berman, O., Chu, S.S., Larson, R.C., Odoni, A.R., and Batta, R. (1990), "Location of mobile units in a stochastic environment", in: P.B. Mirchandani and R.L. Francis (eds.), *Discrete Location Theory*, Wiley, New York.

Bertsimas, D.J. (1992), "A Vehicle Routing Problem with stochastic demand", *Operations Research* 40/3, 574–585.

Bertsimas, D.J., and Van Ryzin, G. (1991), "A stochastic and dynamic vehicle routing problem in the Euclidean plane", *Operations Research* 39, 601–615.

Bertsimas, D.J., and Van Ryzin, G. (1993), "Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles", *Operations Research* 41/1, 60–76.

Bodin, L.D., Golden, B.L., Assad, A., and Ball, M. (1983), "Routing and scheduling of vehicles and crews: The state of the art", *Computers & Operations Research* 10, 69–211.

Brown, G.G., and Graves, G.W. (1981), "Real time dispatching of petroleum tank trucks", *Management Science* 27, 19–32.

Ceder, A., and Stern, H.I. (1981), "Deficit function bus scheduling with deadheading trip insertion for fleet size reduction", *Transportation Science* 15, 338–363.

Christofides, N., and Eilon S. (1969), "An algorithm for the Vehicle Dispatching Problem", *Operations Research* 20.

Christofides, N., Mingozzi, A., and Toth, P. (1981a), "Exact algorithms for the Vehicle Routing Problem based on spanning tree and shortest path relaxations", *Mathematical Programming* 20, 255–282.

Christofides, N., Mingozzi, A., and Toth, P. (1981b), "Space state relaxation procedures for the computation of bounds to routing problems", *Networks* 11, 145–164.

Christofides, N. (1985), "Vehicle routing", in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (eds.), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, New York.

Cullen, F., Jarvis, J., and Ratliff, H. (1981), "Set partitioning based heuristics for interactive routing", *Networks*, 125–143.

Davis, L. (ed.) (1987), *Genetic Algorithms and Simulated Annealing*, Pitman, London.

Desrochers, M., Desrosiers, J., and Solomon, M. (1992), "A new optimization algorithm for the Vehicle Routing Problem with time windows", *Operations Research* 40/2, 342–354.

Dror, M., and Trudeau, P. (1986), "Stochastic vehicle routing with modified savings algorithm", *European Journal of Operational Research* 23, 228–235.

Dumas, Y., Desrosiers, J., and Soumis, F. (1991), "The pickup and delivery problem with time windows", *European Journal of Operational Research* 54, 7–22.

El-Azm, A. (1985), "The Minimum Fleet Size Problem and its application to bus scheduling", in: J.M. Rosseau (ed.), *Computer Scheduling of Public Transport*, North-Holland, Amsterdam, 493–512.

Fisher, M.L. (1994), "Optimal solution of Vehicle Routing Problems using minimum *k*-trees", *Operations Research* 42/4, 626–642.

Gallego, G., and Simchi-Levi, D. (1990), "On the effectiveness of the direct shipping strategy for one warehouse multiretailer *R*-systems", *Management Science*, 240–243.

Goldberg, D.E. (1988), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York.

Golden, B.L., and Assad, A.A. (1988), *Vehicle Routing: Methods and Studies*, Elsevier Science Publishers, Amsterdam.

Haimovich, M., and Rinnooy Kan, A.H.G. (1985), "Bounds and heuristics for Capacitated Routing Problems", *Mathematics of Operations Research* 10, 527–542.

Haimovich, M., Rinnooy Kan, A.H.G., and Stougie, L. (1988), "Analysis of heuristics for Vehicle Routing Problems", in: B.L. Golden and A.A. Assad (eds.), *Vehicle Routing: Methods and Studies*, Elsevier Science Publishers, Amsterdam, 47–61.

Haouari, M., Dejax, P., and Desrochers, M. (1990), "Modelling and solving complex Vehicle Routing Problems using column generation", Working Paper, LEIS, École Centrale, Paris.

Jaillet, P. (1988), "A priori solution of a Traveling Salesman Problem in which a random subset of customers are visited", *Operations Research 36*, 929–936.

Jaillet, P., and Odoni, A. (1988), "The stochastic vehicle routing problem", in: B.L. Golden and A.A. Assad (eds.), *Vehicle Routing: Methods and Studies*, Elsevier Science Publishers, Amsterdam.

Laporte, G., and Nobert, Y. (1987), "Exact algorithms for the Vehicle Routing Problem", *Annals of Discrete Mathematics* 31, 147–184.

Laporte, G., Nobert, Y., and Desrochers, M. (1985), "Optimal routing under capacity and distance restrictions", *Operations Research* 33, 498–516.

Laporte, G., Nobert, Y., and Taillefer, S. (1988), "Solving a family of multidepot vehicle routing and location routing problems", *Transportation Science* 22, 161–172.

Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., and Shmoys, D.B. (1985), *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, New York.

Li, C.L., and Simchi-Levi, D. (1990), "Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems", *ORSA Journal on Computing* 2, 64–73.

Li, C.L., Simchi-Levi, D., and Desrochers, M. (1992), "On the distance constrained vehicle routing problem", *Operations Research* 40/4, 790–799.

Malmborg, C.J., and Lutley, R. (1989), "A PC based system for financial transit retrieval operations", *Industrial Engineering* 21/12.

Malmborg, C.J., and Simons, G.R. (1989), "Integrating logistical and processing functions through mathematical modelling", *Applied Mathematical Modelling* 13/6, 357–364.

Osman, I.H. (1993), "Metastrategy simulated annealing and tabu search algorithms for the Vehicle Routing Problem", *Annals of Operations Research* 41, 421–451.

Powell, W. (1988), "A comparative review of alternative algorithms for the Dynamic Vehicle Allocation Problem", in: B.L. Golden and A.A. Assad (eds.), *Vehicle Routing: Methods and Studies*, Elsevier Science Publishers, Amsterdam.

Psaraftis, H.N. (1983), "*K*-interchange procedures for local search in a precedence constrained routing problem", *European Journal of Operational Research* 13, 391–402.

Savelsbergh, M.W.P. (1985), "Local search in routing problems with time windows", *Annals of Operations Research* 4, 285–305.

Smith, B. and Wren, A. (1981), "VAMPIRES and TASC: Two successfully applied bus scheduling programs", in: A. Wren (ed.), *Computer Scheduling of Public Transport*, North-Holland, Amsterdam, 92–124.

Solomon, M.M., and Desrosiers, J. (1988), "Time window constrained routing and scheduling problems", *Transportation Science* 22, 1–13.

Stewart, W.R. (1987), "Accelerated branch exchange heuristics for symmetric traveling salesman problems", *Networks* 17, 423–437.

Stewart, W.R., and Golden, B. (1983), "Stochastic vehicle routing: A comprehensive approach", *European Journal of Operational Research* 14, 371–385.

Thompson, P.M., and Psaraftis, H.N. (1993), "Cyclic transfer algorithms for multivehicle routing and scheduling algorithms", *Operations Research* 41/5, 935–946.