SINTEF REPORT

SINTEF Applied Mathematics

Address: P.O.Box 124, Blindern
0314 Oslo NORWAY
Location: Forskningsveien 1
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

TITLE

Genetic Algorithms for the Vehicle Routing Problem with Time Windows

AUTHOR(S)

Olli Bräysy and Michel Gendreau

CLIENT(S)

SINTEF Applied Mathematics, Research Council of Norway

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
STF42 A01021	Open	TOP – 140689/420	
CLASS. THIS PAGE: Open	ISBN: 82-14-01613-4	PROJECT NO.: 42203405	NO. OF PAGES/APPENDICES: 22/0
ELECTRONIC FILE CODE: GA_VRPTW_Report.doc		PROJECT MANAGER (NAME, SIGN.): Geir Hasle	CHECKED BY (NAME, SIGN.): Geir Hasle
FILE CODE	DATE: 2001-12-17	APPROVED BY (NAME, POSITION, SIGN.): Geir Hasle, Research Director	

ABSTRACT

This report surveys the research on the genetic and evolutionary algorithms for the Vehicle Routing Problem with Time Windows (VRPTW). The VRPTW can be described as the problem of designing least cost routes from one depot to a set of geographically scattered points. The routes must be designed in such a way that each point is visited only once by exactly one vehicle within a given time interval; all routes start and end at the depot, and the total demands of all points on one particular route must not exceed the capacity of the vehicle. In addition to describing basic features of each method, experimental results for Solomon's benchmark test problems are presented and analyzed.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Computer Science	Datateknikk
GROUP 2	Optimization	Optimering
SELECTED BY AUTHOR	Genetic Algorithm (GA)	Genetiske Algoritmer
	Vehicle Routing Problem (VRP)	Ruteplanlegging
	Metaheuristic	Metaheuristikk

TABLE OF CONTENTS

1 Introduction

Transportation is an important domain of human activity. It supports and makes possible most other social and economic activities. Whenever we use a telephone, shop at our neighborhood foodstore or mall, read our mail or fly for business or pleasure, we are the beneficiaries of some system that has routed messages, goods or people from one place to another. Freight transportation, in particular, is one of today's most important activities, not only measured by the yardstick of its own share of a nation's gross national product (GNP), but also by the increasing influence that the transportation and distribution of goods have on the performance of virtually all other economic sectors. Let us mention that the annual cost of excess travel in the United States has been estimated at some USD 45 billion (King and Mast, 1997) and the turnover of transportation of goods in Europe is some USD 168 billion per year. In the United Kingdom, France and Denmark, for example, transportation represents some 15%, 9% and 15% of national expenditures respectively (Crainic and Laporte 1997; Larsen 1999). It is estimated that distribution costs account for almost half of the total logistics costs and in some industries, such as in the food and drink business, distribution costs can account for up to 70% of the value added costs of goods (De Backer et al. 1997; Golden and Wasil 1987). Halse (1992) reports that in 1989, 76.5% of all the transportation of goods was done by vehicles, which also underlines the importance of routing and scheduling problems.

The Vehicle Routing Problem with Time Windows (VRPTW) is an important problem occurring in many distribution systems. VRPTW can be described as the problem of designing least cost routes from one depot to a set of geographically scattered points. The routes must be designed in such a way that each point is visited only once by exactly one vehicle within a given time interval, all routes start and end at the depot, and the total demands of all points on one particular route must not exceed the capacity of the vehicle. The VRPTW has multiple objectives in that the goal is to minimize not only the number of vehicles required, but also the total travel time and total travel distance incurred by the fleet of vehicles. Some of the most useful applications of the VRPTW include bank deliveries, postal deliveries, industrial refuse collection, national franchise restaurant services, school bus routing, security patrol services and JIT (just in time) manufacturing.

The VRPTW has been the subject of intensive research efforts for both heuristic and exact optimization approaches. Early surveys of solution techniques for the VRPTW can be found in Golden and Assad (1986), Desrochers et al. (1988), Golden and Assad (1988), and Solomon and Desrosiers (1988). Desrosiers et al. (1995) and Cordeau et al. (2001a) mostly focus on exact techniques. Further details on these exact methods can be found in Larsen (1999) and Cook and Rich (1999). For recent survey on tabu search heuristics for the VRPTW, see Bräysy and Gendreau (2001).

Because of the high complexity level of the VRPTW and its wide applicability to real-life situations, solution techniques capable of producing high-quality solutions in limited time, i.e., heuristics are of prime importance. Genetic algorithms have definitely been among the most suitable approaches for tackling the VRPTW. To our knowledge, these have not been comprehensively surveyed and compared. The purpose of this survey is to fill this gap. The remainder of this paper is organized as follows. In section 2, we recall the formulation of the problem as an integer program. Genetic algorithms are reviewed in section 3 and the results are presented and analyzed in section 4. Section 5 concludes the paper.

2 Problem formulation

The VRPTW is defined on a graph (N, A). The node set N consist of the set of customers, denoted by C, and the nodes 0 and n+1, which represent the depot. The number of customers $|C|$ will be denoted n and the customers will be denoted by 1,2,…,n. The arc set A corresponds to possible connections between the nodes. No arc terminates at node 0 and no arc originates at node n+1. All routes start at 0 and end at n+1. A cost c_{ij} and travel time t_{ij} are associated with each arc $(i, j) \in A$ of the network. The travel time t_{ij} includes a service time at customer i. The set of (identical) vehicles is denoted by V. Each vehicle has a given capacity q and each customer a demand d_i, $i \in C$. At each customer, the start of the service must be within a given time interval, called a time window, $[a_i, b_i]$, $i \in C$. Vehicles must also leave the depot within the time window $[a_0, b_0]$ and return during the time window $[a_{n+1}, b_{n+1}]$. A vehicle is permitted to arrive before the opening of the time window, and wait at no cost until service becomes possible, but it is not permitted to arrive after the latest time window. Since waiting time is permitted at no cost, we may assume without loss of generality that $a_0 = b_0 = 0$; that is, all routes start at time 0.

The model contains two types of decision variables. The decision variable X_{ij}^k (defined $\forall(i, j) \in A, \forall k \in V$) is at equal to 1 if vehicle k drives from node i to node j, and 0 otherwise. The decision variable S_i^k (defined $\forall i \in N, \forall k \in V$) denotes the time vehicle k, $k \in V$, starts service at customer i, $i \in C$. If vehicle k does not service customer i, S_i^k has no meaning. We may assume that $S_0^k = 0, \forall k$, and S_{n+1}^k denotes the arrival time of vehicle k at the depot. The objective is to design a set of minimal cost routes, one for each vehicle, such that all customers are serviced exactly once. Hence, split deliveries are not allowed. The routes must be feasible with respect to the capacity of the vehicles and the time windows of the customers serviced. The VRPTW can be stated mathematically as:

$$\text{minimize} \quad \sum_{k \in V} \sum_{(i,j) \in A} c_{ij} X_{ij}^k \tag{1}$$

subject to :

$$\sum_{k \in V} \sum_{j \in N} X_{ij}^k = 1, \quad \forall i \in C \tag{2}$$

$$\sum_{i \in C} d_i \sum_{j \in N} X_{ij}^k \leq q, \quad \forall k \in V \tag{3}$$

$$\sum_{j \in N} X_{0j}^k = 1, \quad \forall k \in V \tag{4}$$

$$\sum_{i \in N} X_{ih}^k - \sum_{j \in N} X_{hj}^k = 0, \quad \forall h \in C, \ \forall k \in V \tag{5}$$

$$\sum_{i \in N} X_{i,n+1}^k = 1, \quad \forall k \in V \tag{6}$$

$$X_{ij}^k (S_i^k + t_{ij} - S_j^k) \leq 0, \quad \forall (i, j) \in A, \forall k \in V \tag{7}$$

$$a_i \leq S_i^k \leq b_i, \quad \forall i \in N, \forall k \in V \tag{8}$$

$$X_{ij}^k \in \{0,1\}, \quad \forall (i, j) \in A, \forall k \in V \tag{9}$$

The objective function (1) states that costs should be minimized. Constraint set (2) states that each customer must be assigned to exactly one vehicle, and constraint set (3) states that no vehicle can service more customers than its capacity permits. Constraint set (4), (5) and (6) are the flow constraints requiring that each vehicle k leaves node 0 once, leaves node h, $h \in C$ if and only if it enters that node, and returns to node n+1. Note that constraint set (6) is redundant, but is maintained in the model to underline the network structure. The arc (0, n+1) is included in the network, to allow empty tours. More precisely, we permit an unrestricted number of vehicles, but a cost c_v is put on each vehicle used. This is done by setting $c_{0,n+1} = -c_v$. The value of c_v is sufficiently large to primarily minimize the number of vehicles and secondarily minimize travel costs. Nonlinear (easily linearized, see for example Desrosiers et al., 1995) constraint set (7) states that vehicle k cannot arrive at j before $S_i^k + t_{ij}$ if it travels from i to j. Constraint set (8) ensures that all time windows are respected and (9) is the set of integrality constraints.

3 Genetic algorithms

The Genetic Algorithm (GA) is an adaptive heuristic search method based on population genetics. The basic concepts were developed by Holland (1975), while the practicality of using the GA to solve complex problems was demonstrated in De Jong (1975) and Goldberg (1989). Details and references about genetic algorithms can also be found for example in Mühlenbein (1997) and Alander (2000) respectively. GA evolves a population of individuals encoded as chromosomes by creating new generations of offspring through an iterative process until some convergence criteria or conditions are met. Such criteria might, for instance, refer to a maximum number of generations or the convergence to a homogeneous population composed of similar individuals. The best chromosome generated is then decoded, providing the corresponding solution.

The creation of a new generation of individuals involves primarily three major steps or phases: selection, recombination and mutation. The selection phase consist in choosing randomly two parent individuals from the population for the mating purposes. The probability of selecting a population member is generally proportional to its fitness in order to emphasize genetic quality while maintaining genetic diversity. Here fitness refers to measure of profit, utility or goodness to be maximized while exploring the solution space. The recombination or reproduction process makes use of genes of selected parents to produce offspring that will form the next generation. As for mutation, it consists in randomly modifying gene(s) of a single individual at a time to further explore the solution space and ensure, or preserve, genetic diversity. The occurrence of mutation is generally associated with a low probability. A new generation is created by repeating the selection, reproduction and mutation processes until all chromosomes in the new population replace those from the old one. A proper balance between genetic quality and diversity is therefore required within the population in order to support efficient search. Although theoretical results that characterize the behavior of the GA have been obtained for bit-string chromosomes, not all problems lend themselves easily to this representation. This is the case, in particular, for sequencing problems, like vehicle routing problem, where an integer representation is more often appropriate.

Blanton and Wainwright (1993) hybridize a genetic algorithm with a greedy heuristic. Under this scheme, the genetic algorithm searches for a good ordering of customers, while the construction of the feasible solution is handled by the greedy heuristic. Specialized genetic operators use a global precedence relationship among the customers to create new orderings from the old ones. For example, it is generally desirable to insert customer c_i before customer c_j during the greedy insertion phase, if the time window at customer c_i occurs before the time window at customer c_j. Accordingly, this relationship is used by the genetic operators to push customers with early time windows to the front of the orderings. The solutions are coded as sequences of customer indexes. The authors test in addition to time windows also other global precedence relationships, namely distance and capacity. For example in the case of distance, the distances between consecutive genes (customers) in parent solutions determine the ordering of the child. The authors use the so-called Davis encoding method, where a chromosome represents a permutation of n customers to be partitioned into m vehicles. The evaluation function assumes that the first m customers of a chromosome are placed into the m vehicles. The remaining $n-m$ customers are examined individually. As each new customer is selected a possible subtour is evaluated for each vehicle and the best subtour is selected. The mutation operator used randomly exchanges two genes in a chromosome. If some of the customers remain unrouted, the fitness value is the number of these unserviced customers, otherwise the fitness value is based on the total distance of the solution.

Louis et al. (1999) improve the algorithm by Blanton and Wainwright (1993) to achieve better performance in problems with clustered customer locations. This is specifically achieved by improving the Davis encoding scheme. The reason why Davis encoding scheme works poorly on non-random customer location is that the first m customers in a chromosome are assigned to m vehicles randomly. Thus there is a high probability that more than one customer from the same cluster is assigned to different vehicles when only one is needed. To avoid this, the authors build a new initialization procedure that pays attention to the clustering of customers in assigning vehicles. Here, the authors use visual clustering of customers, i.e., determine the clusters manually, and the approach is tested on six problems of Solomon's dataset C1. The authors disregard the most difficult instances C103 and C104, but to other problems they report high quality solutions.

Thangiah (1995a) describes a cluster-first, route second method called GIDEON that assigns customers to vehicles by partitioning the customers into sectors by a genetic algorithm. Customers within each formed sector are routed using the cheapest insertion method of Golden and Stewart (1985). In the next step, the routes are improved using λ-interchanges introduced by Osman (1993). The two processes are run iteratively a finite number of times to improve the solution quality and the proposed search strategy accepts also infeasibilities during the search against certain penalty factors. The search begins by clustering customers either according to their polar coordinate angle or randomly. More precisely, first the customers are divided into K sectors, by planting a set of seed angles in the search space and drawing a ray from the depot to each seed angle. Here K equals the initial number of vehicles required to service the customers according Solomon's insertion heuristic. Then a pseudo polar coordinate angle is calculated for each customer by normalizing the angles between the customers so that the angular difference between any two adjacent customers is equal. Customer c_i is assigned to vehicle r_k if the pseudo polar coordinate angle s_i is greater than the seed angle S_k but is less than or equal to seed angle S_{k+1}. Each seed angle is computed using a fixed angle and an offset from the fixed angle. The Genetic Algorithm is used to decide the offset values that allow the sector to encompass a larger or smaller sector area. In the GIDEON system, each chromosome represents a set of possible clustering schemes and the fitness values are based on corresponding routing costs. The crossover operator exchanges a randomly selected portion of the bit string between the chromosomes and mutation is used with a very low probability to randomly change the bit values.

Thangiah (1995b) develops a similar approach as GIDEON, called GenClust. Instead of sectors, particular geometric shapes are used to cluster customers. In GenSect, each chromosome encodes n_c different circles, one for each cluster, and the GA is then used to search for the set of circles that lead to the best solution. Different heuristic rules are used to associate a customer with a particular cluster, when it is not contained in exactly one circle (i.e., none or many).

Thangiah et al. (1995) examine the same approach as Thangiah (1995a) to solve the vehicle routing problems with time deadlines (VRPTD), i.e., without earliest time window. In addition they created two heuristics based on principles of time oriented sweep and cheapest insertion procedures for solving the VRPTD, followed by λ-interchanges of Osman (1993). Authors conclude that the genetic algorithm based heuristic does well for problems in which the customers are distributed uniformly and/or with short time deadlines. The other two heuristics perform well for problems in which the customers are tightly clustered or have long deadlines.

Thangiah et al. (1994) develop a number of metaheuristics based on a two-phase approach. In the first phase an initial solution is created by either the cheapest insertion heuristic or the sectoring based genetic algorithm GIDEON by Thangiah (1995a). The second phase applies one of the following search procedures which use the λ-interchange mechanism: a local search descent procedure using either the first-accept or the best-accept strategy, a hybrid Simulated Annealing (SA) algorithm (Metropolis et al., 1953) with non-monotonic cooling schedule or a hybrid simulated annealing and tabu search. Tabu search algorithm is combined with the SA based acceptance criterion to decide which moves to accept from the candidate list. The main feature of the local search procedures is that infeasible solutions with penalties are allowed if considered attractive.

Potvin et al. (1996a) use competitive neural network, described in Potvin and Robillard (1995) to select the seed customers for the modification of Solomon's (1987) insertion heuristic by Potvin and Rousseau (1993), where several routes are constructed simultaneously. The algorithm requires a value for three parameters, α_1, α_2 and μ. The first two constants determine the importance of distance and travel time in the cost function for each unrouted customer. The third factor is used to control the savings in distance. A genetic algorithm is used to find values for these three constants. A stochastic selection procedure is applied to the fitness values based on the number of routes and total route time of the best solution produced by the parallel insertion heuristic. A classical 2-point crossover operator is used for recombination. It swaps a segment of consecutive bits between the parents. The mutation changes with very low probability a bit value from 0 to 1 or from 1 to 0. The results are slightly better compared to using the original insertion heuristic without preprocessing.

Benyahia and Potvin (1995) use similar GA approach as Potvin et al. (1996a) to optimize the parameter values of the sequential and parallel versions of Solomon's (1987) insertion heuristic. However, here seed customers are selected as in Solomon (1987) and Potvin and Rousseau (1993) instead of neural networks. Moreover, authors introduce additional cost measures for insertion, involving slack and waiting times, saving of insertion compared to servicing the customer by individual route, and ratio of additional distance to original distance between the pair of consecutive customers.

Potvin and Bengio (1996) propose a genetic algorithm called GENEROUS that directly applies genetic operators to solutions, thus avoiding the coding issues. The initial population is created with the cheapest insertion heuristic of Solomon (1987) and the fitness values of the proposed approach are based on the number of vehicles and total route time. The selection process is stochastic and biased towards the best solutions. For this purpose a linear ranking scheme is used. During the recombination phase, two parent solutions are merged into a single one, so as to guarantee the feasibility of the new solution. Two types of crossover operators are used, namely a sequence-based and a route-based crossover. The sequence-based crossover selects first randomly a link from each parent solution. Then the customers that are serviced before the breakpoint on the route of parent-solution P_1 are linked to the customers that are serviced after the break point on the route of parent solution P_2. Finally, the new route replaces the old one in parent solution P_1. Route-based crossover replaces one route of parent solution P_2 by a route of parent solution P_1. A special repair operator is then applied to the offspring to generate a new feasible solution. Mutation operators are aimed at reducing the number of routes by trying to insert the customers of a randomly selected short route into other routes, either directly or by first removing some customer from the target route and inserting it in some other route to make room for the new customer. Finally, in order to locally optimize the solution, a mutation operator based on Or-opt exchanges (Or, 1976) is included.

Berger et al. (1998) hybridize a genetic algorithm with well-known construction heuristics. The authors omit the coding issues and represent a solution by a set of feasible routes. The initial population is created with a nearest neighbor heuristic inspired from Solomon (1987). The fitness values of the individuals are based on the number of routes and total distance of the corresponding solution and for selection purposes the authors use the so-called “roulette-wheel” scheme. In this scheme the probability of selecting an individual is proportional to its fitness; for details, see Goldberg (1989). The proposed crossover operator combines iteratively various routes r_1 of parent solution P_1 with a subset of customers, formed by r_2 nearest-neighbor routes from parent solution P_2. A removal procedure is first carried out to remove some key customer nodes from r_1. Then an insertion heuristic inspired from Solomon (1987) coupled to a random customer acceptance procedure is locally applied to build a feasible route, considering the partial route r_1 as initial solution. Here only the customer nodes in routes r_2 are considered for insertion. The stochastic customer removal procedure removes either randomly specific customers, customers rather distant from their successors, or customers with waiting times. The mutation operators are aimed at reducing the number of routes of solutions having only a few customers by trying to insert them into other routes or locally reordering routes using the Nearest Neighbor heuristic of Solomon. Berger et al. (1999) used similar approach to tackle VRPTW with itinerary constraints (maximum route time).

Bräysy (1999a and 1999b) extends the work of Berger et al. (1998) by proposing several new crossover and mutation operators, testing different forms of genetic algorithms, selection schemes and scaling schemes, as well as the significance of the initial solutions. The best performing recombination operator removes first a set of customers within randomly generated segments from parent solution P_1. Then the reinsertion of these removed customers is performed by considering only customers in geographically close routes in parent solution P_2 for insertion. The best-performing mutation operator selects randomly one of the shortest routes and tries to eliminate it by inserting the customers into other longer routes. Regarding different forms of genetic algorithms it is concluded that it is important to create many new offspring each generation and it is enough to maintain only one population. Differences between different selection schemes are concluded to be minor. The best results were obtained with so-called tournament selection that performs well-known roulette-wheel scheme twice and selects the better out of the two individuals identified by the roulette-wheel scheme. A new scaling scheme based on a weighted combination of number of routes, total distance and waiting time is found to perform particularly well. Finally to create the initial population, several strategies, such us heuristics of Solomon (1987) and randomly created routes are tried and it is concluded that the best strategy is to create a diverse initial population that also contains some individuals with better fitness scores.

Bräysy et al. (2000) hybridize a genetic algorithm and an evolutionary algorithm consisting of several local search and route construction heuristics inspired from the studies of Solomon (1987) and Taillard et al. (1997). In the first phase a genetic algorithm based on the studies by Berger et al. (1998) and Bräysy (1999a) is used to obtain a feasible solution. The main differences compared to previous studies lie in the usage of a random heuristic to create the initial population, and usage of a Large Neighborhood Search (LNS) based-strategy by Shaw (1998) within the recombination and mutation. The evolutionary algorithm used in the second phase picks every combination of two routes in random order and applies randomly one out of the four local search operators or route construction heuristics. Offspring routes generated by these crossover operators are mutated according to a user-defined probability by selecting randomly one out of two operators. Selecting each possible pair of routes, mating and mutation operators are repeatedly applied for a certain number of generations and finally a feasible solution is returned. To escape from a local minimum, arcs longer than average are penalized if they appear frequently during the search.

Berger et al. (2001) further continue the study by Berger et al. (1998). The proposed genetic algorithm evolves two populations in parallel. The first population is used to minimize the total distance and the second population tries to minimize violation of time window constraints. The initial population is created using a random sequential insertion heuristic combined with λ-exchanges and reinitialization procedure based on insertion procedure by Liu and Shen (1999). The first of the two recombination operators is the same as in Berger et al. (1998). The second extends the first operator by removing also illegally routed customers and by using insertion procedure proposed in Liu and Shen (1999) instead of Solomon's (1987) heuristic in the reinsertion phase. Here cost function is extended to consider temporal constraint violation. Six mutation operators are presented. The first one is a modified version of ant colony optimization approach by Gambardella et al. (1999), where the customers that cannot be inserted to current limited set of routes without violating time windows constraints, are inserted using the insertion procedure by Liu and Shen (1999) with an extended cost function. The second mutation operator is a modification of LNS by Shaw (1998), where the order of reinsertions and the cost function are new. Other mutation operators involve λ-exchanges, exchange of customers served too late in current solution, elimination of shortest route using procedure by Liu and Shen (1999) and within-route reordering using Solomon's (1987) heuristic.

Homberger and Gehring (1999) propose two evolutionary metaheuristics for the VRPTW. The individual representation includes a vector of so-called "strategy parameters" in addition to the solution vector and both components are evolved by means of recombination and mutation operators. In the proposed application for the VRPTW, these strategy parameters refer to how often a randomly selected local search operator is applied and to binary parameter used to alternate the search between minimizing the number of vehicles and total distance. Selection of the parents is done randomly and only one offspring is created through the recombination of parents. This way a number $\lambda > \mu$ offsprings is created, where μ is the population size. At the end, fitness values are used to select μ offsprings to the next population. The local search operators used in the mutation are Or-opt (Or, 1976), 2-opt* (Potvin and Rousseau, 1995) and λ–interchange-move (Osman, 1993) with $\lambda = 1$. In addition a special Or-opt based operator is used to reduce the number of routes. The first out of the two proposed metaheuristics, evolution strategy ES1, skips the recombination phase. The second evolution strategy ES2 uses an uniform order-based crossover to modify the initially randomly created mutation codes. The mutation code is used to control a set of removal and insertion operators performed by Or-opt operator. The fitness values are based on number of routes, total travel distance and on a criterion that determines how easily the shortest route of the solution in terms of the number of customers on the route can be eliminated. The individuals of a starting population are generated by means of a stochastic approach that is based on the savings algorithm of Clarke and Wright (1964). In this approach, the stochastic element consists of the random selection of savings elements from the savings list.

Gehring and Homberger (1999) study a two-phase approach, where the evolution strategy ES1 described above with population size of one is used in the first phase to minimize the number of routes. In the second phase, the total distance is minimized using a tabu search algorithm utilizing the same local search operators. The approach is parallelized using the concept of cooperative autonomy, i.e., several autonomous sequential solution procedures cooperate through the exchange of solutions. The cooperating slave processes are configured in different ways using different seeds for random number generators to create diversity in the search. The authors also develop a new set of larger benchmark problems that are based on the benchmark problems of Solomon (1987).

Gehring and Homberger (2001) introduce some improvements to the parallel method by Gehring and Homberger (1999) described above. First, population size is greater than one. In the evaluation of individuals also capacity related information is used to determine the route for elimination. Additional improvements include new termination criteria for both phases, where the search is stopped once minimum required number of vehicles is obtained.

Zhu (2000) presents a genetic algorithm based on integer representation of solutions and two new crossover operators. The initial population is a combination of solutions created by the insertion heuristic of Solomon (1987), a set of randomly created λ-interchange neighbors of the heuristic solution and totally randomly created solutions. The parents are selected with tournament selection and the recombination is based on selecting randomly a cut-off point in both parents and then selecting the customer right after the cut-off point from either parent. Then this customer is inserted to same position in the other parent solution, and corrections are performed to replace duplicate customer by the replaced one. The next customer to be reinserted is selected based either on distances or latest arrival times with respect to previously chosen customer. Mutation is based on reversing the order of a pair or sequence of nodes. Moreover, a special hill-climbing technique is used, where a randomly selected part of the population is improved by partial λ-exchanges. The author concludes that the obtained results are better than the ones reported by Thangiah (1995a). Further details as well as detailed results can be found in Tan et al. (2000).

Tan et al. (2001) introduce a similar genetic algorithm to Zhu (2000) described above. The representation, strategy for creating the initial population and selection scheme are the same. Well-known PMX crossover operator is used to interchange gene materials between chromosomes and mutation is performed by randomly swapping nodes. The basic idea in PMX crossover is to choose two cut points at random and based on these cut points perform a series of swapping operations in the second parent. Since the authors do not use delimiters to distinguish customers belonging to different routes, one chromosome may be grouped into routes in several different ways. The basic grouping is determined by insertion heuristic of Solomon (1987), and the authors use λinterchanges to create alternative groupings. After the grouping process, a similar hill-climbing method as used by Zhu (2000) is employed to further improve a part of the population.

Wee Kit et al. (2001) propose a hybrid genetic algorithm, where a simple tabu search based on cross, exchange, relocate and 2-opt neighborhoods is applied on individual solutions in the later generations to intensify the search. The genetic algorithm is based on random selection of parent solutions and two new crossover operators. The first operator tries to modify the order of the customers in the first parent by trying to create consecutive pairs of customers according to the second parent. The second crossover operator tries to copy common characteristics of parent solutions to offspring by modifying the seed selection procedure and cost function of insertion heuristic similar to Solomon (1987). The authors do not define used mutation and initial solution procedures. Similar study is reported earlier in Chin et al. (1999).

Rahoual et al. (2001) describe a multicriteria genetic algorithm, where the evaluation of the individuals is based on a weighted sum of objectives related to violated constraints, number of vehicles and total distance. Solutions are represented as integer strings, describing chronological list of customers for each vehicle. The initial population is generated randomly and crossover is based on 2-opt*, where cutting points are determined randomly. Mutation consists on random reinsertions of customers between routes and the selection is done probabilistically based on the ranks of the individuals. Finally, instead of replacing preceding population by the current one, worst individuals in the population are replaced by the best ones found during the search.

Le Bouthillier et al. (2001) present a parallel methodology that combines solutions from different agents in a pool of feasible solutions. The agents consist of simple construction algorithms to initialize the pool and different local search and metaheuristic methods. The local searches include 2-opt, Or-opt and 3-opt operators. Evolutionary algorithm using a probabilistic mutation and merge crossover as well as different tabu searches based on GENIUS neighborhood by Gendreau et al. (1992) and allowance of infeasible solutions during the search are used to perform the global search.

4 Analysis of results

In this section we present and analyze the results obtained with the above described genetic algorithms to the Solomon’s (1987) 56 benchmark problems. These problems have been the most common way to assess and compare the value of the various heuristic approaches, proposed in the literature. These problems have a hundred customers, a central depot, capacity constraints, time windows on the time of delivery, and a total route time constraint. The C1 and C2 classes have customers located in clusters and in the R1 and R2 classes the customers are at random positions. The RC1 and RC2 classes contain a mix of both random and clustered customers. Each class contains between 8 and 12 individual problem instances and all problems in any one class have the same customer locations, and the same vehicle capacities; only the time windows differ. In terms of time window density (the percentage of customers with time windows), the problems have 25%, 50%, 75%, and 100% time windows. The C1, R1 and RC1 problems have a short scheduling horizon, and require 9 to 19 vehicles. Short horizon problems have vehicles that have small capacities and short route times, and cannot service many customers at one time. Classes C2, R2 and RC2 are more representative of “long-haul” delivery with longer scheduling horizons and fewer (2–4) vehicles. Both travel time and distance are given by the Euclidean distance between points.

The results are usually ranked according to a hierarchical objective function, where the number of vehicles is considered as the primary objective, and for the same number of vehicles, the secondary objective is often either total traveled distance or total duration of routes. Therefore a solution requiring fewer routes is always considered better than solution with more routes, regardless of the total traveled distance.

The genetic and evolutionary approaches described above are compared in Table 1 against some of the other recent metaheuristics. All algorithms in Table 1 are stochastic and they are implemented in C, except the one described in Wee Kit et al. (2001) and Bräysy (2001b) that are coded in Java. Tan et al. (2000) do not report the programming language used. A hierarchic objective function is used in every case, except in Tan et al. (2000 and 2001), where the only objective is to minimize total distance. The number of routes is considered as the primary objective and, for the same number of routes, the secondary objective is to minimize the total traveled distance. An exception is found in Potvin and Bengio (1996), where the second objective is to minimize the total duration of routes.

When one considers only evolutionary and genetic algorithms, it seems that the evolutionary algorithms by Homberger and Gehring (1999), hybrids by Gehring and Homberger (1999 and 2001), and genetic algorithm of Berger et al. (2001) produce the best results according to Table 1. The differences between these best methods in terms of solution quality are small, only less than 0.5% in CNV and about 1% in CTD. When it comes to other approaches, the worst results regarding the CNV are produced by Tan et al. (2000 and 2001) that focus only on minimizing total distance. Tan et al. (2001) seems to be better of the two methods, producing results that are competitive even with the best known in terms of distance. If one considers only methods with similar objective function, the one proposed in Wee Kit et al. (2001) appears to perform worst.

Generally the difference in number of vehicles is about 6%, if Tan et al. (2000 and 2001) are not considered. Problem group RC2 seems to be the most problematic regarding the total traveled distance. For example, the difference between Thangiah (1995a) and Gehring and Homberger (2001) is about 25% that can hardly be justified in practical settings.

Table 1: Comparison of evolutionary and genetic algorithms. For each algorithm the average results with respect to Solomon's benchmarks are depicted. Notations CNV and CTD in the rightmost column indicate the cumulative number of vehicles and cumulative total distance over all 56 test problems.

Author	R1	R2	C1	C2	RC1	RC2	CNV/CTD
Thangiah et al. (1994)	**12.33**	**3.00**	**10.00**	**3.00**	**12.00**	**3.38**	**418**
	1227.42	1005.00	830.89	640.86	1391.13	1173.38	58905
Thangiah (1995a)	**12.75**	**3.18**	**10.00**	**3.00**	**12.50**	**3.38**	**429**
	1300.25	1124.28	892.11	749.13	1474.13	1411.13	65074
Potvin et al. (1996)	**12.58**	**3.00**	**10.00**	**3.00**	**12.13**	**3.38**	**422**
	1296.83	1117.64	838.11	590.00	1446.25	1368.13	62634
Berger et al. (1998)	**12.58**	**3.09**	**10.00**	**3.00**	**12.13**	**3.50**	**424**
	1261.58	1030.01	834.61	594.25	1441.35	1284.25	60539
Bräysy (1999b)	**12.58**	**3.09**	**10.00**	**3.00**	**12.13**	**3.38**	**423**
	1272.34	1053.65	857.64	624.31	1417.05	1256.80	60962
Homberger et al.	**11.92**	**2.73**	**10.00**	**3.00**	**11.63**	**3.25**	**406**
(1999)	1228.06	969.95	828.38	589.86	1392.57	1144.43	57876
Homberger et al.	**12.00**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**406**
(1999)	1226.38	1033.58	828.38	589.86	1406.58	1175.98	58921
Gehring et al. (1999)	**12.42**	**2.82**	**10.00**	**3.00**	**11.88**	**3.25**	**415**
	1198	947	829	590	1356	1140	56942
Bräysy et al. (2000)	**12.42**	**3.09**	**10.00**	**3.00**	**12.13**	**3.38**	**421**
	1213.86	978.00	828.75	591.81	1372.20	1170.23	57857
Tan et al. (2000)	**14.42**	**5.64**	**10.11**	**3.25**	**14.63**	**7.00**	**525**
	1314.79	1093.37	860.62	623.47	1512.94	1282.47	62901
Gehring et al. (2001)	**12.08**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**407**
	1208.14	965.46	828.50	589.88	1389.65	1126.22	57420
Gehring et al. (2001)	**12.00**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**406**
	1217.57	961.29	828.63	590.33	1395.13	1139.37	57641
Berger et al. (2001)	**11.92**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**405**
	1221.10	975.43	828.48	589.93	1389.89	1159.37	57952
Tan et al. (2001)	**13.17**	**5.00**	**10.11**	**3.25**	**13.50**	**5.00**	**478**
	1227	980	861	619	1427	1123	58605
Wee Kit et al.(2001)	**12.58**	**3.18**	**10.00**	**3.00**	**12.75**	**3.75**	**432**
	1203.32	951.17	833.32	593.00	1382.06	1132.79	57265
Rahoual et al. (2001)	**12.92**	____	**10.00**	____	**12.63**	____	____
	1362		887		1487		
Gambardella et al.	**12.00**	**2.73**	**10.00**	**3.00**	**11.63**	**3.25**	**407**
(1999)	1217.73	967.75	828.38	589.86	1382.42	1129.19	57525
Bräysy (2001b)	**11.92**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**405**
	1222.12	975.12	828.38	589.86	1389.58	1128.38	57710
Cordeau et al. (2001b)	**12.08**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**407**
	1210.14	969.57	828.38	589.86	1389.78	1134.52	57556

The following list provides information concerning author, computer used, number of runs and average time consumption in minutes to solve the problems.

Thangiah et al. (1994), NeXT 25 MHz, number of runs not reported, 30 min.,
Thangiah (1995a), Solbourne 5/802, number of runs not reported, 2.1 min.,
Potvin and Bengio (1996), Sun Sparc 10, number of runs not reported, 25 min.,
Berger et al. (1998), Sun Sparc 10, number of runs not reported, 1–10 min.,
Bräysy (1999b), Sun Ultra Enterprise 450 (300 MHz), 5 runs, 17 min.,
Homberger and Gehring (1999, method ES1), Pentium 200 MHz, 10 runs, 13 min.,
Homberger and Gehring (1999, method ES2), Pentium 200 MHz, 10 runs, 19 min.,
Gehring and Homberger (1999), 4×Pentium 200 MHz, 1 run, 5 min.,
Bräysy et al. (2000), Pentium Celeron 366 MHz, 5 runs, 15 min.,
Tan et al. (2000), Pentium II 266 MHz, number of runs and time consumption not reported,
Gehring and Homberger (2001, method HM4), 4×Pentium 400 MHz, 5 runs, 13.5 min.,
Gehring and Homberger (2001, method HM4C), 4×Pentium 400 MHz, 5 runs, 15.1 min.,
Berger et al. (2001), Pentium 400 MHz, 3 runs, 30 minutes.,
Tan et al. (2001), Pentium II 330 MHz, number of runs not reported, 25 min.,
Wee Kit et al. (2001), Digital Workstation 433a, number of runs not reported, 147.4 min.,
Rahoual et al. (2001), computational effort not reported.,
Gambardella et al. (1999), Sun Ultra Sparc 1 167 MHz, number of runs and the time not reported,
Bräysy (2001b), Pentium 200 MHz, 1 run, 82.5 min.,
Cordeau et al. (2001b), Sun Ultra 2 300 MHz, number of runs and time not reported.

When comparing the results of different evolutionary and genetic algorithms with the results produced with other recent approaches, described in the lower part of Table 1, one can conclude that the performance of recent evolutionary and genetic algorithms is excellent. The recent genetic algorithm by Berger et al. (2001) is able to obtain the lowest known cumulative number of vehicles, and the total distance values of Gehring and Homberger (2001) are even better than the ones obtained with other recent approaches showing the best performance. Since only Bräysy (1999b, 2000 and 2001b), Homberger and Gehring (1999), Gehring and Homberger (2001) and Berger et al. (2001) report the number of runs required to obtain the reported results, it is impossible to draw any final conclusions regarding which method performs best. Another comparison is provided in Table 2, where only results obtained with limited computational results are considered.

We consider in Table 2 only results for which the time consumption and the number of runs are described. To facilitate the comparison, the effect of different hardware is normalized to equal Sun Sparc 10 using Dongarra’s (1998) factors. In addition, if the reported results are the best ones over multiple experiments, we multiplied the computation times by this number to see the real computational effort. The computer, number of independent runs, and the CPU time used to obtain the reported results are described in the lower part of Table 2. The number of runs is greater than one only if the reported result is best over multiple executions of the given algorithm and the CPU time is reported only for a single run. Two CPU time values are described: the one reported by the authors and in the parenthesis the modified CPU time. Details for calculating these modified CPU times can be found in Bräysy (2001a).

Table 2: Comparison of results obtained with limited computational effort for Solomon's benchmark problems.

Author	R1	R2	C1	C2	RC1	RC2	CNV/CTD
(1) Homberger et al.	**11.92**	**2.73**	**10.00**	**3.00**	**11.63**	**3.25**	**406**
(1999)	1228.06	969.95	828.38	589.86	1392.57	1144.43	57876
(2) Homberger et al.	**12.00**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**406**
(1999)	1226.38	1033.58	828.38	589.86	1406.58	1175.98	58921
(3) Gehring et al. (1999)	**12.42**	**2.82**	**10.00**	**3.00**	**11.88**	**3.25**	**415**
	1198	947	829	590	1356	1144	56946
(4) Bräysy (1999b)	**12.58**	**3.09**	**10.00**	**3.00**	**12.13**	**3.38**	**423**
	1272.34	1053.65	857.64	624.31	1417.05	1256.80	60962
(5) Bräysy et al. (2000)	**12.42**	**3.09**	**10.00**	**3.00**	**12.13**	**3.38**	**421**
	1213.86	978.00	828.75	591.81	1372.20	1170.23	57857
(6) Gehring et al. (2001)	**12.00**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**406**
	1217.57	961.29	828.63	590.33	1395.13	1139.37	57641
(7) Gehring et al. (2001)	**12.08**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**407**
	1208.14	965.46	828.50	589.88	1389.65	1126.22	57420
(8) Berger et al. (2001)	**12.17**	**2.73**	**10.00**	**3.00**	**11.88**	**3.25**	**411**
	1251.40	1056.59	828.50	590.06	1414.86	1258.15	60200
(9) Gambardella et al.	**12.38**	**3.00**	**10.00**	**3.00**	**11.92**	**3.33**	**418**
(1999)	1210.83	960.31	828.38	591.85	1388.13	1149.28	57583
(10) Liu et al (1999)	**12.17**	**2.82**	**10.00**	**3.00**	**11.88**	**3.25**	**412**
	1249.57	1016.58	830.06	591.03	1412.87	1204.87	59318
(11) Bräysy (2001b)	**11.92**	**2.73**	**10.00**	**3.00**	**11.50**	**3.25**	**405**
	1222.12	975.12	828.38	589.86	1389.58	1128.38	57710

(1) Pentium 200 MHz, 10 runs, 13 (312) min., (2) Pentium 200 MHz, 10 runs, 19 (456) min., (3) 4×Pentium 200 MHz, 1 run, 5 (48) min., (4) Sun Ultra Enterprise 450 (300 MHz), 5 runs, 17 (374) min., (5) Pentium Celeron 366 MHz, 5 runs, 15 (360) min., (6) 4×Pentium 400 MHz, 5 runs, 13.5 (1458) min., (7) 4×Pentium 400 MHz, 5 runs, 15.1 (1631) min., (8) Pentium 400 MHz, 1 run, 30 (162) min., (9) Sun Ultrasparc 1, 1 run 30 (210) min., (10) HP 9000/720, 3 runs, 20 (102) min., (11) Pentium 200 MHz, 1 run, 82.5 (198) min.

According to Table 2, the algorithms by Homberger and Gehring (1999), Gehring and Homberger (2001), Bräysy (2001b) and Berger et al. (2001) show the best overall performance. Regarding individual problem groups, it seems that practically all approaches yield excellent results to problem groups C1 and C2, having customers located in geographical clusters. Bräysy (2001b) gives the best output in R1 and RC1 on average, though the difference to the best competing approaches is very small, less than 0.5%. Gehring and Homberger (2001) report the best results to R2 and RC2 on average, though again the differences to the best competing approaches by

Homberger and Gehring (1999) and Bräysy (2001b) respectively, are very small. Also in general, the differences between recent results, reported in year 2001, are small, varying within 2%. The only exception is problem group RC2, where the difference between Gehring et al. (2001) and Berger et al. (2001) is about 12%. For detailed best-known results to Solomon's benchmarks, we refer to http://www.top.sintef.no/.

5 Conclusions

NP-hardness of the vehicle routing problem requires heuristic solution strategies for most real-life instances. In the previous sections, we have comprehensively surveyed the genetic and evolutionary algorithms for the vehicle routing problem with time windows. Currently algorithms by Homberger and Gehring (1999), Gehring and Homberger (2001) and Berger et al. (2001) seem to achieve best performance. These evolutionary and genetic algorithms show better performance than any other heuristic approach. Only the recent variable neighborhood search approach by Bräysy (2001b) is able to perform better. Though, here one must note that the difference in performance to other recent metaheuristics, such as tabu search of Cordeau et al. (2001b) and ant colony system of Gambardella et al. (1999) remain small. It is our hope that this paper has provided valuable insights for the pursuit of solutions to many current and future challenging problems.

Acknowledgements

This work was partially supported by the Emil Aaltonen Foundation, the Canadian Natural Science and Engineering Research Council and the TOP programme funded by the Research Council of Norway. This support is gratefully acknowledged.

6 References

Alander, J. T. (2000), "An Indexed Bibliography of Genetic Algorithms in Operations Research", Technical report series No. 94-1-OR, University of Vaasa, Finland. Available via anonymous `ftp:` site `ftp.uwasa.fi` directory `cs/report94-1` file `gaORbib.ps.Z.`

Benyahia, I. and J.-Y. Potvin (1995), "Generalization and Refinement of Route Construction Heuristics using Genetic Algorithms", in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, 39–43, IEEE Sercive Center, Piscataway, U.S.A.

Berger, J., M. Salois and R. Begin (1998), "A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows", Lecture Notes in Artificial Intelligence 1418, AI'98 Advances in Artificial Intelligence, 114–127, Vancouver.

Berger, J., M. Sassi and M. Salois (1999), "A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows and Itinerary Constraints", in *Proceedings of the Genetic and*

Evolutionary Computation Conference, W. Banzhaff (ed), 44–51, Morgan Kaufmann Publishing, San Francisco.

Berger, J., M. Barkaoui and O. Bräysy (2001), "A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows", Working Paper, Defence Research Establishment Valcartier, Canada.

Blanton, J.L. and R.L. Wainwright (1993), "Multiple Vehicle Routing with Time and Capacity Constraints using Genetic Algorithms", in *Proceedings of the Fifth International Conference on Genetic Algorithms*, S. Forrest (ed), 452–459, Morgan Kaufmann Publishing, San Francisco.

Bräysy, O. (1999a), "A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows", Licentiate thesis, University of Vaasa, Finland.

Bräysy, O. (1999b), "A New Algorithm for the Vehicle Routing Problem with Time Windows Based on the Hybridization of a Genetic Algorithm and Route Construction Heuristics", *Proceedings of the University of Vaasa*, *Research papers* 227.

Bräysy, O., J. Berger and M. Barkaoui (2000), "A New Hybrid Evolutionary Algorithm for the Vehicle Routing Problem with Time Windows", Presented at the Route 2000 workshop, Skodsborg, Denmark.

Bräysy, O. (2001a), "Local Search and Variable Neighborhood Search Algorithms for the Vehicle Routing Problem with Time Windows", Doctoral Dissertation, University of Vaasa, Finland.

Bräysy, O. (2001b), "A Reactive Variable Neighborhood Search for the Vehicle Routing Problem with Time Windows", to be published in INFORMS Journal on Computing.

Bräysy, O. and M. Gendreau (2001), "Tabu Search Heuristics for the Vehicle Routing Problem with Time Windows", Internal Report STF42 A01022, SINTEF Applied Mathematics, Department of Optimisation, Norway.

Chin, A., H. Wee Kit and A. Lim (1999), "A New GA Approach for the Vehicle Routing Problem" in *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, 307–310, IEEE Computer Society, Los Alamitos, California.

Clarke, G. and J.W. Wright (1964), "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research* 12, 568–581.

Cook, W. and J.L. Rich (1999), "A Parallel Cutting-Plane Algorithm for the Vehicle Routing Problems with Time Windows", Working Paper, Department of Computational and Applied Mathematics, Rice University, Houston.

Cordeau, J.F., G. Desaulniers, J. Desrosiers, M. M. Solomon and F. Soumis (2001a), "The VRP with Time Windows", To appear in The Vehicle Routing Problem, Chapter 7, Paolo Toth and Daniele Vigo (eds), SIAM Monographs on Discrete Mathematics and Applications.

Cordeau, J.-F., G. Laporte, A. Mercier (2001b), "A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows", *Journal of the Operational Research Society* 52, 928–936.

Crainic, T. G. and G. Laporte (1997), "Planning Models for Freight Transportation", *European Journal of Operational Research* 97, 409–438.

De Backer, B. and V. Furnon (1997), "Meta-heuristics in Constraint Programming Experiments with Tabu Search on the Vehicle Routing Problem", presented at the Second International Conference on Metaheuristics (MIC´97), Sophia Antipolis, France.

Desrochers, M., J.K. Lenstra, M.W.P. Savelsbergh and F. Soumis (1988), "Vehicle Routing with Time Windows: Optimization and Approximation", in *Vehicle Routing*: *Methods and Studies*, B. Golden and A. Assad (eds), 65–84, Elsevier Science Publishers, Amsterdam.

Desrosiers, J, Y. Dumas, M.M. Solomon and F. Soumis (1995), "Time Constrained Routing and Scheduling", in *Handbooks in Operations Research and Management Science 8*: *Network Routing*, M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser (eds), 35–139, Elsevier Science Publishers, Amsterdam.

Dongarra, J. (1998), "Performance of Various Computers Using Standard Linear Equations Software", Report CS-89-85, Department of Computer Science, University of Tennessee, U.S.A.

Gambardella, L.M., E. Taillard and G. Agazzi (1999), "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows", in *New Ideas in Optimization*, D. Corne, M. Dorigo and F. Glover (eds), 63–76, McGraw-Hill, London.

Gehring, H. and J. Homberger (1999), "A Parallel Hybrid Evolutionary Metaheuristic for the Vehicle Routing Problem with Time Windows", in *Proceedings of EUROGEN99 - Short Course on Evolutionary Algorithms in Engineering and Computer Science*, *Reports of the Department of Mathematical Information Technology*, Series A. Collections, No. A 2/1999, K. Miettinen, M. Mäkelä and J. Toivanen (eds), 57–64, University of Jyväskylä, Finland.

Gehring, H. and J. Homberger (2001), "Parallelization of a Two-Phase Metaheuristic for Routing Problems with Time Windows", *Asia-Pacific Journal of Operational Research* 18, 35–47.

Gendreau, M., A. Hertz and G. Laporte (1992), "A New Insertion and Postoptimization Procedures for the Traveling Salesman Problem", *Operations Research* 40, 1086–1093.

Goldberg, D. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Publishing Company Inc, New York.

Golden, B. L. and W. R. Stewart (1985), "Empirical Analysis of Heuristics", in *The Traveling Salesman Problem*, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (eds), 207–249, John Wiley & Sons, Chichester.

Golden, B.L. and A.A. Assad (1986), "Perspectives on Vehicle Routing: Exciting New Developments", *Operations Research* 34, 803–809.

Golden, B.L. and E.A. Wasil (1987), "Computerized Vehicle Routing in the Soft Drink Industry", *Operations Research* 35, 6–17.

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

Halse, K. (1992), "Modeling and Solving Complex Vehicle Routing Problems", Ph. D. thesis, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

Homberger, J. and H. Gehring (1999), "Two Evolutionary Meta-heuristics for the Vehicle Routing Problem with Time Windows", *INFOR* 37, 297–318.

Jong De, K.A. (1975), "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", Ph.D. thesis, University of Michigan, U.S.A.

King, G. F. and C. F. Mast (1997), "Excess Travel: Causes, Extent and Consequences", *Transportation Research Record* 1111, 126–134.

Larsen, J. (1999), "Parallelization of the Vehicle Routing Problem with Time Windows", Ph.D. thesis, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Le Bouthillier, A., T. G. Crainic and R . K. Keller (2001), "Co-operative Parallel Method for Vehicle Routing Problems with Time Windows", presented at MIC'2001, 4th Metaheuristics International Conference, July 16-20th, Porto, Portugal.

Liu, F.-H. and S.-Y. Shen (1999), "A Route-neighborhood-based Metaheuristic for Vehicle Routing Problem with Time Windows", *European Journal of Operations Research* 118, 485–504.

Louis, S.J., Y. Xiangying and Y.Y. Zhen (1999), "Multiple Vehicle Routing with Time Windows using Genetic Algorithms", in *Proceedings of the 1999 Congress on Evolutionary Computation*, 1804–1808, IEEE, Pistacaway.

Metropolis, W., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller (1953), "Equation of the State Calculations by Fast Computing Machines", *Journal of Chemical Physics* 21, 1087–1092.

Mühlenbein, H. (1997), "Genetic Algorithms", in *Local Search in Combinatorial Optimization*, E. Aarts and J. K. Lenstra (eds), 137–172, John Wiley & Sons, Chichester.

Or, I. (1976), "Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Regional Blood Banking", Ph.D. thesis, Northwestern University, Evanston, Illinois.

Osman, I.H. (1993), "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problems", *Annals of Operations Research* 41, 421–452.

Potvin, J.-Y. and J.-M. Rousseau (1993), "A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows", *European Journal of Operational Research* 66, 331–340.

Potvin, J.-Y. and J.-M. Rousseau (1995), "An Exchange Heuristic for Routeing Problems with Time Windows", *Journal of the Operational Research Society* 46, 1433–1446.

Potvin, J.-Y. and C. Robillard (1995), "Clustering for Vehicle Routing with a Competitive Neural Network", *Neurocomputing* 8, 125–139.

Potvin, J.-Y. and S. Bengio (1996), "The Vehicle Routing Problem with Time Windows Part II: Genetic Search", *INFORMS Journal on Computing* 8, 165–172.

Potvin, J.-Y., D. Dubé and C. Robillard (1996), "A Hybrid Approach to Vehicle Routing Using Neural Networks and Genetic Algorithms", *Applied Intelligence* 6, 241-252.

Rahoual, M., B. Kitoun, M.-H. Mabed, V. Bachelet and F. Benameur (2001), “Multicriteria Genetic Algorithms for the Vehicle Routing Problem with Time Windows”, Presented at MIC’2001, 4th Metaheuristic International Conference, July 16-20th, Porto, Portugal.

Shaw, P. (1998), “Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems”, in *Principles and Practice of Constraint Programming – CP98, Lecture Notes in Computer Science*, M. Maher and J.-F. Puget (eds), 417–431, Springer-Verlag, New York.

Solomon, M.M. (1987), “Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints”, *Operations Research* 35, 254–265.

Solomon, M.M. and J. Desrosiers (1988), “Time Window Constrained Routing and Scheduling Problems”, *Transportation Science* 22, 1–13.

Tan, K. C., L. H. Lee and K. Q. Zhu (2000), “Heuristic Methods for Vehicle Routing Problem with Time Windows”, in *Proceedings of the 6th International Symposium on Artificial Intelligence & Mathematics*, Ft. Lauderdale, Florida.

Tan, K. C., L. H. Lee and K. Ou (2001), “Hybrid Genetic Algorithms in Solving Vehicle Routing Problems with Time Window Constraints”, *Asia-Pacific Journal of Operational Research* 18, 121–130.

Thangiah, S., I. Osman and T. Sun (1994), “Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows”, Working Paper UKC/IMS/OR94/4, Institute of Mathematics and Statistics, University of Kent, Canterbury.

Thangiah, S. (1995a), “Vehicle Routing with Time Windows Using Genetic Algorithms”, in *Application Handbook of Genetic Algorithms*: *New Frontiers*, *Volume II*, L. Chambers (ed), 253–277, CRC Press, Boca Raton.

Thangiah, S.R. (1995b), “An Adaptive Clustering Method Using a Geometric Shape for Vehicle Routing Problems with Time Windows”, in *Proceedings of the 6th International Conference on Genetic Algorithms*, L.J. Eshelman (ed), 536–543, Morgan Kaufmann, San Francisco.

Thangiah, S.R., I.H. Osman, R. Vinayagamoorthy and T. Sun (1995), “Algorithms for the Vehicle Routing Problems with Time Deadlines”, *American Journal of Mathematical and Management Sciences* 13, 323–355.

Wee Kit, H., J. Chin and A. Lim (2001), “A Hybrid Search Algorithm for the Vehicle Routing Problem with Time Windows”, *International Journal on Artificial Intelligence Tools* (to appear).

Zhu, K. Q. (2000), “A New Genetic Algorithm for VRPTW”, presented at IC-AI 2000, Las Vegas, U.S.A.