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

Route Construction and Local Search 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 A01024	Open	TOP – 140689/420	
CLASS. THIS PAGE	**ISBN**	**PROJECT NO.**	**NO. OF PAGES/APPENDICES**
Open	82-14-01614-2	42203405	28/0
ELECTRONIC FILE CODE		**PROJECT MANAGER (NAME, SIGN.)**	**CHECKED BY (NAME, SIGN.)**
Local_VRPTW_Report.doc		Geir Hasle	Geir Hasle
FILE CODE	**DATE**	**APPROVED BY (NAME, POSITION, SIGN.)**	
	2001-12-18	Geir Hasle, Research Director	

ABSTRACT

This report presents a survey of research on 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. Both traditional heuristic route construction methods and recent local search algorithms are examined. The basic features of each method are described, and experimental results for Solomon’s benchmark test problems are presented and analyzed. Moreover, we discuss how heuristic methods should be evaluated and propose using the concept of Pareto optimality in the comparison of different heuristic approaches. The metaheuristic methods are described in Bräysy and Gendreau (2001).

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Computer Science	Datateknikk
GROUP 2	Optimization	Optimering
SELECTED BY AUTHOR	Vehicle Routing Problem (VRP)	Ruteplanlegging
	Construction Heuristics	Konstruksjonsheuristikk
	Local Search	Lokalsøk

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 food store 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. 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. 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). 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. Over the last few years, many authors have proposed new heuristic approaches, mostly metaheuristics, 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. In this report, we examine traditional heuristic approaches, that is, route construction and route improvement (local search) methods. These are of interest by themselves since they can provide good solutions with a low computational effort, but also because they are a major component of all metaheuristics for the VRPTW. Metaheuristics are discussed in Bräysy and Gendreau (2001).

The remainder of this paper is organized as follows. In section 2, we recall the formulation of the problem as an integer program. Section 3 is devoted to a discussion of how heuristics are to be evaluated. Route construction techniques are reviewed in section 4 and route improvement (local search) methods in section 5. Section 6 concludes the paper.

2 Problem formulation

The VRPTW is defined on a graph (N, A). The node set N consists 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 $1,2,\ldots,n$. 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 deadline. 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 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:

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

subject to :

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

$$\sum_{i\in C} d_i \sum_{j\in N} X_{ij}^k \le q, \qquad \forall k \in V \qquad (3)$$

$$\sum_{j\in N} X_{0j}^k = 1, \qquad \forall k \in V \qquad (4)$$

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

$$\sum_{i\in N} X_{i,n+1}^k = 1, \qquad \forall k \in V \qquad (6)$$

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

$$a_i \le S_i^k \le b_i, \qquad \forall i \in N, \forall k \in V \qquad (8)$$

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

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 Evaluation of heuristics

Evaluation of any heuristic method is subject to the comparison of a number of criteria that relate to various aspects of algorithm performance. Examples of such criteria are running time, quality of solution, ease of implementation, robustness and flexibility (Barr et al., 1995; Cordeau et al., 2001b). Since heuristic methods are ultimately designed to solve real world problems, flexibility is an important consideration. An algorithm should be able to easily handle changes in the model, the constraints and the objective function. As for robustness, an algorithm should still able to produce results under difficult circumstances such as when a problem instance is pathological. Extensive discussions on these subjects can be found in Cordeau et al. (2001b).

The time a heuristic takes to produce good quality solutions can be crucial when choosing between different techniques. Similarly the quality of the final solution, as measured by the objective function, is important. How close the solution is to the optimal solution is a standard measure of quality or, if the heuristic is designed to simply produce feasible solutions, then the ability of the heuristic to provide such solutions is important.

There is generally a trade-off between run time and solution quality – the longer a heuristic is run the better the quality of the final solution. A compromise is needed so that good quality solutions are produced in a reasonable amount of time. Basically this trade-off between run time and solution quality can be viewed in terms of a multiobjective optimization in which the two objectives are balanced. Performance measures for heuristics can be plotted in two-dimensional space, with the first dimension corresponding to run time and the second to solution quality. In that space, points such that there exist no other points with better values on both dimensions are said to be Pareto optimal; they define effective compromises between the objectives. This is illustrated in Figure 5-7 of section 5, where points Antes et al. (1995), Russell (1995) and Bräysy (2001a) are the Pareto optimal ones. The choice between different heuristic approaches yielding Pareto optimal results depends on the preferences of the decision-maker and the situation at hand.

By far the most common method of evaluating the solution quality of a heuristic algorithm is empirical analysis. In general, empirical analysis involves testing the heuristic across a wide range of problem instances to get an idea of overall performance. To arrive at conclusions that have meaning in a statistical sense, experimental design should ideally be used on different levels of the various algorithm parameters and the results compared by appropriate techniques.

In the actual comparison of heuristics one often faces a number of difficulties. The most obvious difficulty is making the competition fair. Differences between machines first come to mind. In this paper, we address this issue by adjusting reported running times by the factors given by Dongarra (1998). Even more difficult issues to face are differences in coding skill, tuning and effort invested (Hooker, 1995)

Other difficulties faced especially in the VRPTW context are that often only the best results obtained during the whole computational study are reported. Moreover, in some cases the authors do not report the number of runs or CPU time required to get the reported results. In these cases it is impossible to conclude anything about the efficiency of the methods, or compare these methods with other approaches. The only adequate basis for comparison of these methods would be optimal solutions, since if enough time is available, it is always preferable to solve the problems to optimality using exact methods. To be able to reach proper conclusions, in addition to the number of runs and time consumption one should answer questions such as what are the limits of the given algorithm, i.e., how good are the best results that can be obtained using the particular approach, how good a solution can be obtained in a given amount of time. One should, in other words, report results obtained using different computation times, and observe how much time is needed to obtain results of a given quality. Moreover, in our opinion, figures describing the relationship between solution quality and computation time would greatly facilitate the analysis. Taillard (2001) discusses extensively this issue and proposes reporting an absolute computational effort, such as number of iterations instead of computation time and using probability diagram based on repeated Mann-Whitney statistical tests. Obviously, such an approach is not possible when one relies on previously published results as we do here.

In the VRPTW context, the most common way to compare heuristics is the results obtained for Solomon's (1987) 56 benchmark problems. 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.

4 Route construction heuristics

Route construction heuristics select nodes (or arcs) sequentially until a feasible solution has been created. Nodes are chosen based on some cost minimization criterion subject to the restriction that the selection does not create a violation of vehicle capacity or time window constraints. Sequential methods construct one route at a time, while parallel methods build several routes simultaneously.

Pullen and Webb (1967) describe a system developed for duty scheduling of van drivers in a heavily time-constrained real-life environment, where part of the customers are considered in ad hoc manner after creating the initial schedule. Additional constraints over basic VRPTW include consideration of employment regulations. A step by step heuristic approach is used to match each service in turn with the schedule of the driver best able to perform it. This matching is carried out on the basis of the lowest cost of idle time and empty running time. In the end, the results of the scheduling process are sorted so that customers are in time order within each route.

Knight and Hofer (1968) present a case study involving a contract transport company. The examined problems involve also some connected calls (pickup and delivery) in addition to independent calls. A two-phase approach is selected, consisting of initial allocation of calls to vehicles prior to a sequential routing of the calls. The allocation of a call to a vehicle is determined by the pivot to which it is nearest. Because of the time constraints and grouping around pivots, only a small number of stops had to be considered at a time, so that the best feasible sequence could easily be determined by inspection. The goal of the developed system was to increase the utilization of vehicles as measured by the average number of calls per vehicle and the total number of vehicles used.

Madsen (1976) develops two algorithms for routing problems with tight due dates faced by a large newspaper and magazine distribution company. The examined approach does not consider vehicle capacities and earliest time windows. Two heuristics are proposed. The first is a parallel insertion heuristic based on due times of the customers and distances of the vehicles from the customers. The second method is based on Monte Carlo simulation. It allocates calls arbitrarily to routes and orders customers in each route according to their due times. Finally the created solution is improved by simple between-route customer reinsertions.

Solomon (1986) uses a so-called route-first cluster-second scheme using a Giant-Tour Heuristic. First the customers are scheduled into one giant route and then this route is divided into a number of routes. The initial giant tour could, for example, be generated as a traveling salesman tour without considering capacity and time constraints. No computational results are given in the paper for the heuristic.

Solomon (1987) describes several heuristics for VRPTW. One of the methods is an extension to the savings heuristic of Clarke and Wright (1964). The savings method, originally developed for classical VRP, is probably the best-known route construction heuristic. It begins with a solution in which every customer is supplied individually by a separate route. Combining any two of these single customer routes results in a cost savings of $S_{ij} = d_{i0} + d_{0j} - d_{ij}$. Clarke and Wright (1964) select the arc (i, j) linking customers c_i and c_j with maximum S_{ij} subject to the requirement that the combined route is feasible. With this convention, the route combining operation can be applied iteratively. In combining routes, we can simultaneously form partial routes for all vehicles or sequentially add customers to a given route until the vehicle is fully loaded. To account for both the spatial and temporal closeness of customers Solomon sets a limit to the waiting time. The savings method is illustrated in Figure 4-1.

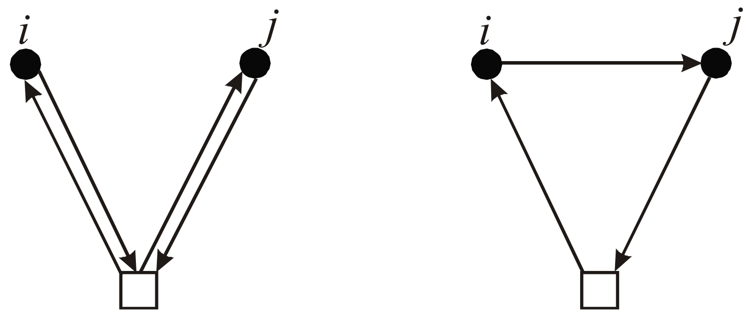

Figure 4-1: The savings heuristic. In the left, customers *i* and *j* are served by separate routes and in the right the routes are combined by inserting customer *j* after *i*.

The second heuristic, a time oriented nearest-neighbor method, starts every route by finding an unrouted customer closest to the depot. At every subsequent iteration, the heuristic searches for the customer closest to the last customer added into the route and adds it at the end of the route. A new route is started any time the search fails to find a feasible insertion place, unless there are no more unrouted customers left. The metric used to measure the closeness of any pair of customers tries to account for both geographical and temporal closeness of customers.

The most successful of the proposed three sequential insertion heuristics is called I1. A route is first initialized with a "seed" customer and the remaining unrouted customers are added into this route until it is full with respect to the scheduling horizon and/or capacity constraint. If unrouted customers remain, the initializations and insertion procedures are then repeated until all customers are serviced. The seed customers are selected by finding either the geographically farthest unrouted customer in relation to the depot or the unrouted customer with the lowest starting time for service. After initializing the current route with a seed customer, the method uses two subsequently defined criteria $c_1(i,u,j)$ and $c_2(i,u,j)$ to select customer u for insertion between adjacent customers i and j in the current partial route. Let $(i_0, i_1, i_2, \ldots, i_m)$ be the current route with i_0 and i_m representing the depot. For each unrouted customer u, we first compute its best feasible insertion cost on the route as

$$c_1(i(u),u,j(u)) = \underset{\rho=1,\ldots,m}{optimum} \quad c_1(i_{\rho-1},u,i_\rho)\,, \tag{10}$$

Next, the best customer u^* to be inserted in the route is the one for which

$$c_2(i(u^*),u^*,j(u^*)) = \underset{u}{optimum} \quad c_2(i(u),u,j(u))\,, \tag{11}$$

u unrouted and feasible.

Client u^* is then inserted into the route between $i(u^*)$ and $j(u^*)$. When no more customers with feasible insertions can be found, the method starts a new route, unless it has already routed all customers. More precisely $c_1(i,u,j)$ is calculated as

$$
\begin{aligned}
&c_1(i,u,j) = \alpha_1 c_{11}(i,u,j) + \alpha_2 c_{12}(i,u,j), && (12)\\
&\text{where } \alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0, &&\\
&c_{11}(i,u,j) = d_{iu} + d_{uj} - \mu d_{ij,}\ \mu \geq 0, && (13)\\
&c_{12}(i,u,j) = b_{ju} - b_j, && (14)
\end{aligned}
$$

and d_{iu}, d_{uj} and d_{ij} are distances between customers i and u, u and j and i and j respectively. Parameter μ controls the savings in distance and b_{ju} denotes the new time for service to begin at customer j, given that u is inserted on the route and b_j is the beginning of service before insertion. The criterion $c_2(i,u,j)$ is calculated as follows

$$c_2(i,u,j) = \lambda d_{0u} - c_1(i,u,j), \lambda \geq 0. \qquad (15)$$

Parameter λ is used to define how much the best insertion place for an unrouted customer depends on its distance from the depot and on the other hand how much the best place depends on the extra distance and extra time required to visit the customer by the current vehicle. The second type of the proposed insertion heuristics (I2) aims to select customers whose insertion costs minimize a measure of total route distance and time and the third approach (I3) accounts for the urgency of servicing a customer.

Dullaert (2000a and 2000b) argues that Solomon's time insertion criterion $c_{12}(i,u,j)$ underestimates the additional time needed to insert a new customer u between the depot and the first customer in the partially constructed route. This can cause the insertion criterion to select suboptimal insertion places for unrouted customers. Thus, a route with a relatively small number of customers can have a larger schedule time than necessary. The author introduces new time insertion criteria to solve the problem and concludes that the new criteria offer significant cost savings starting from more than 50%. These cost savings are however concluded to decrease as the number of customers per route increases.

The time oriented sweep heuristic of Solomon (1987) is based on the idea of decomposing the problem into a clustering stage and a scheduling stage. In the first phase, customers are assigned to vehicles as in the original sweep heuristic (Gillett and Miller 1974). Here a "center of gravity" is computed and the customers are partitioned according to their polar angle. In the second phase customers assigned to a vehicle are scheduled using an insertion heuristic of type I1.

Potvin and Rousseau (1993) introduce a parallel version of Solomon's insertion heuristic I1, where the set of *m* routes is initialized at once. The authors use Solomon's sequential insertion heuristic to determine the initial number of routes and the set of seed customers. The selection of the next customer to be inserted is based on a generalized regret measure over all routes. Large regret measure means that there is a large gap between the best and second best insertion places for a customer.

Foisy and Potvin (1993) implemented the above-described parallel version of Solomon's insertion heuristic on parallel hardware consisting of 2–6 Sun 3 workstation transputers. The parallelism is exploited in calculation of insertion cost for each customer. The selection of the best customer for insertion is then run only on half of the available processors. To reduce the unequal workload among the processors, unrouted customers are reassigned among the processors so as to reduce the average processor's idle time. The authors conclude that the overall reduction in computation time is linear with the number of processors for the distributed part of the heuristic algorithm.

Ioannou et al. (2001) use the generic sequential insertion framework proposed by Solomon (1987) to solve a number of theoretical benchmark problems and an industrial example from food industry. The proposed approach is based on new criteria for customer selection and insertion that are motivated by the minimization function of the greedy look-ahead approach solution approach of Atkinson (1994). The basic idea behind the criteria is that a customer *u* selected for insertion into a route should minimize the impact of the insertion on the customers already on the route under construction, on all non-routed customers, and on the time window of customer *u*.

Balakrishnan (1993) describes three heuristics for the vehicle routing problem with soft time windows (VRPSTW). The heuristics are based on nearest-neighbor and Clarke-Wright savings rules and they differ only in the way used to determine the first customer on a route and in the criteria used to identify the next customer for insertion. The motivation behind VRPSTW is that by allowing limited time window violations for some customers, it may be possible to obtain significant reductions in the number of vehicles required and/or the total distance or time of all routes. Among the soft time window problem instances, dial-a-ride problems play a central role.

Bramel and Simchi-Levi (1996) propose an asymptotically optimal heuristic based on the idea of solving the capacitated location problem with time windows (CLPTW). In CLPTW the objective is to select a subset of possible sites, to locate one vehicle to each site and to assign customers to the vehicles. In the VRPTW context, this selection of locations for vehicles refers to selecting a set of seed customers that initialize the routes. The authors use a Lagrangian relaxation based technique to solve the CLPTW and the other customers are inserted in greedy order into simple tours by favoring customers that least increase the distance traveled. The authors conclude that their heuristic provides a better solution than Solomon's heuristic for 25 of the 56 problems using reasonable running times.

Table 1 compares some of the described route construction algorithms. The first column to the left describes the authors. Columns R1, R2, C1, C2, RC1 and RC2 present the average number of vehicles and average total distance with respect to six problem groups of Solomon (1987). Finally, the rightmost column indicates the cumulative number of vehicles and cumulative total distance over all 56 test problems. In the lower part of the Table we report information regarding the computer used, number of runs and average time consumption of a single run in minutes as reported by the authors. We could not include all described algorithms in the Table due to lack of information (not all authors report results properly or use Solomon's problem set). In Table 1, the number of vehicles is the primary minimization objective and the secondary objective is total duration of routes in Solomon (1987) and Potvin and Rousseau (1993), and total distance in Ioannou et al. (2001). The methods by Solomon (1987) and Potvin and Rousseau (1993) are coded in Fortran and Ioannou et al. (2001) does not report the used programming language. Finally, since we used rounded distance measures reported by other authors to calculate the Cumulative Total Distance (CTD), we rounded the values to integers in Tables 1 and 2.

Table 1: Route construction heuristics. For all algorithms the average results for Solomon's benchmarks are described. The 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
(1) Solomon (1987)	**13.58**	**3.27**	**10.00**	**3.13**	**13.50**	**3.88**	**453**
	1436.7	1402.4	951.9	692.7	1596.5	1682.1	73004
(2) Potvin et al. (1993)	**13.33**	**3.09**	**10.67**	**3.38**	**13.38**	**3.63**	**453**
	1509.04	1386.67	1343.69	797.59	1723.72	1651.05	78834
(3) Ioannou et al.	**12.67**	**3.09**	**10.00**	**3.13**	**12.50**	**3.50**	**429**
(2001)	1370	1310	865	662	1512	1483	67891
(1) DEC 10, 1 run, 0.6 min., (2) IBM PC, 1 run, 19.6 min., (3) Intel Pentium 133 MHz, 1 run, 4.0 min.							

It seems that Ioannou et al. (2001) produces the best results, though at the cost of higher computation times. As for the other two methods, Solomon (1987) seems to be better than Potvin and Rousseau (1993) only in clustered problem groups C1 and C2, while the opposite is true for the other problem groups. These heuristics are very fast and there are not significant differences in the computational burden, if one takes into account the differences in the hardware used. Compared to local search approaches, these construction heuristics are considerably faster, as one can see from Figure 5-7. However, these simple procedures lack in solution quality compared to more sophisticated approaches.

5 Solution improvement methods

Classical local search methods form a general class of approximate heuristics based on the concept of iteratively improving the solution to a problem by exploring neighboring ones. In order to design a local search algorithm, one typically needs to specify the following choices: How an initial feasible solution is generated, what move-generation mechanism to use, the acceptance criterion and stopping test. The move-generation mechanism creates the neighboring solutions by changing one attribute or a combination of attributes of an given instance. Here attribute could refer for example to an arc connecting a pair of customers. Once a neighboring solution is identified, it is compared against the current solution. If the neighboring solution is better, it replaces the current solution, and the search continues. Two acceptance strategies are common in the VRPTW context, namely a first-accept (FA) and best-accept (BA). The first-accept strategy selects the first neighbor that satisfies the pre-defined acceptance criterion. The best-accept strategy examines all neighbors satisfying the criteria and selects the best among them.

The local optimum produced by any local search procedure can be very far from the optimal solution. Local search methods perform a myopic search since they only accept sequentially solutions that produce reductions in the objective function value. Thus the outcome depends heavily on initial solutions and the neighborhood generation mechanism used. Most iterative improvement methods that have been applied to vehicle routing and scheduling problems are edge-exchange algorithms.

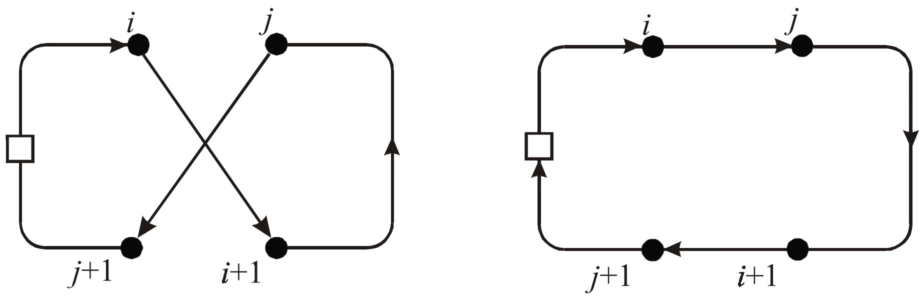

Figure 5-1: 2-opt exchange operator. The edges (*i*, *i*+1) and (*j*, *j*+1) are replaced by edges (*i*, *j*) and (*i*+1, *j*+1), thus reversing the direction of customers between *i*+1 and *j*.

The edge-exchange neighborhoods for a single route are set of tours that can be obtained from an initial tour by replacing a set of k of its edges by another set of k edges. Such replacements are called k-exchanges, and a tour that cannot be improved by a k-exchange is said to be k-optimal. Verifying k-optimality requires $O(n^k)$ time. 2-exchange or 2-opt is illustrated in Figure 5-1. It tries to improve the tour by replacing two of its edges by two other edges and iterates until no further improvement is possible.

Russell (1977) reports early work on the VRPTW for a k-optimal improvement heuristic. The so-called M-Tour approach was effective in solving an actual problem with a few time constrained customers. A solution for a 163-customer problem with 15% time constrained customers was generated in less than 90 seconds of IBM 370/168 CPU time.

Efficient implementations for speeding up the screening of infeasible solutions and the evaluation of the objective function are reported in Savelsbergh (1986), Solomon and Desrosiers (1988), Solomon, Baker and Schaffer (1988), Savelsbergh (1990) and Savelsbergh (1992). The techniques used involve preprocessing, tailored updating mechanisms and lexicographic search strategies.

Baker and Schaffer (1986) report a computational study of route improvement procedures, which are applied to heuristically generated initial solutions. Time-oriented nearest neighbor and three different cheapest insertion algorithms with differing cost functions are used for solution construction purposes. The cost functions consider one or more out of the following components: distance, increase in arrival time and waiting time. The improvement methods considered are extensions to the VRPTW of the 2-opt and 3-opt edge exchange procedures of Lin (1965). Both within-route and between-route exchanges are tested. The authors conclude that the best overall solutions are usually obtained from the best starting solutions and that generally the cheapest insertion procedures outperformed the nearest neighbor ones. The authors also conclude that only less than 10% of the solution improvements involve the reversal of the orientation of a sequence of two or more customers.

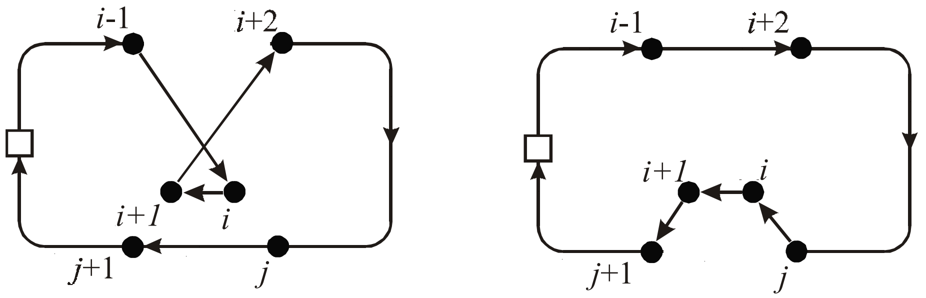

Figure 5-2: The Or-opt operator. Customers *i* and *i*+1 are relocated to be served between two customers *j* and *j*+1 instead of customers *i*-1 and *i*+2. This is performed by replacing 3 edges (*i*-1, *i*), (*i*+*1*, *i*+2) and (*j*, j+1) by the edges (*i*-1, *i*+2), (*j*, *i*) and (*i*+*1*, *j*+1), preserving the orientation of the route.

Van Landeghem (1988) presents a bi-criteria heuristic based on the savings method of Clarke and Wright (1964). More precisely, the author proposes combining original savings measure in terms of timing with so called "loss of flexibility". The flexibility is defined as the difference between customer time window length and route time window length after combining. Route time window refers to the difference between time slots inside which a vehicle can start servicing the first and last customers on the route. In the end, the results are improved using simple customer reinsertions. A closely related operator is the Or-opt introduced by Or (1976) for the traveling salesman problem. The basic idea is to relocate a chain of *l* consecutive vertices (customers). This is achieved by replacing three edges in the original tour by three new ones without modifying the orientation of the route as illustrated in Figure 5-2.

Potvin and Rousseau (1995) compare different edge exchange heuristics for VRPTW (2-opt, 3-opt and Or-opt) and introduce a new 2-opt* exchange heuristic. The basic idea in 2-opt* is to combine two routes so that the last customers of a given route are introduced at the end of first customers of another route, thus preserving the orientation of the routes. The operator is illustrated in Figure 5-3, where the edges (i, i+1) and (j, j+1) are replaced by (i, j+1) and (j, i+1), i.e., the end portions of two routes are exchanged. As a special case, it can combine two routes into one if edge (i, i+1) is the first one on its route and edge (j, j+1) the last one on its route or vice versa. A hybrid approach based on Or-opt and 2-opt* exchanges is found to be particularly powerful. This approach oscillates between the two neighborhoods by changing the operator each time local minimum is found. The authors test also an implementation where the two operators were merged together. The initial solutions are produced with Solomon's I1 heuristic.

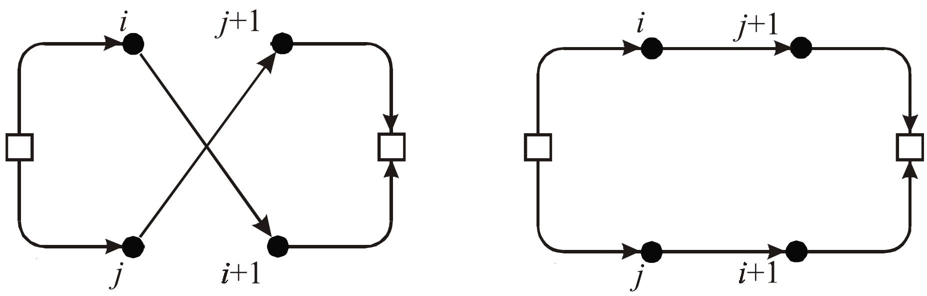

Figure 5-3: 2-opt* operator. The customers served after customer *i* on the upper route are reinserted to be served after customer *j* on the lower route and customers after *j* on the lower route are moved to be served on the upper route after customer *i*. This is performed by replacing edges (*i*, *i*+1) and (*j*, *j*+1) with edges (*i*, *j*+1) and (*j*, *i*+1).

Prosser and Shaw (1996) compare within-route 2-opt by Lin (1965) and inter-route operators relocate, exchange and cross, originally proposed by Savelsbergh (1992) for classical VRP. The 2-opt works by reversing part of a single route (see Figure 5-1). The relocate operator simply moves a customer visit from one route to another. It is illustrated in Figure 5-4. The exchange heuristic swaps two visits in different routes. This is pictured in Figure 5-5. Finally, cross is similar to 2-opt* proposed by Potvin and Rousseau (1995) for VRPTW. It is illustrated in Figure 5-3. Initially, a virtual vehicle exists that performs the visits not carried out by the real vehicles. The virtual vehicle is different from the real ones in two respects. First, the virtual vehicle can make an unlimited number of customer visits. Second, the cost incurred by the virtual vehicle when it performs a customer visit is typically higher than that incurred by a real vehicle.

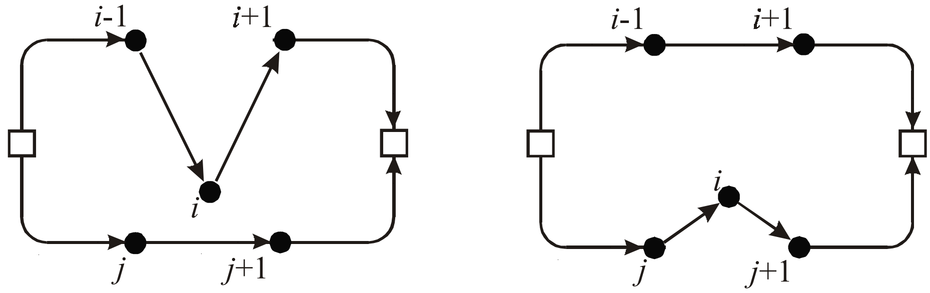

Figure 5-4: Relocate operator. The edges (*i*-1, *i*), (*i*, *i*+1) and (*j*, *j*+1) are replaced by (*i*-1, *i*+1), (*j*, *i*) and (*i*, *j*+1), i.e., customer *i* from the origin route is placed into the destination route.

De Backer et al. (1997) report similar research as Prosser and Shaw (1996) in the Constraint Programming (CP) context. In CP, the computation is driven by constraints, thus giving them an active role. Looking locally at a particular constraint, the algorithm attempts to remove, from the domain of each variable involved in that constraint, values that cannot be part of any solution. For more details on Constraint Programming, see, for example, Jaffar and Lassez (1986) and Jaffar and Maher (1994).

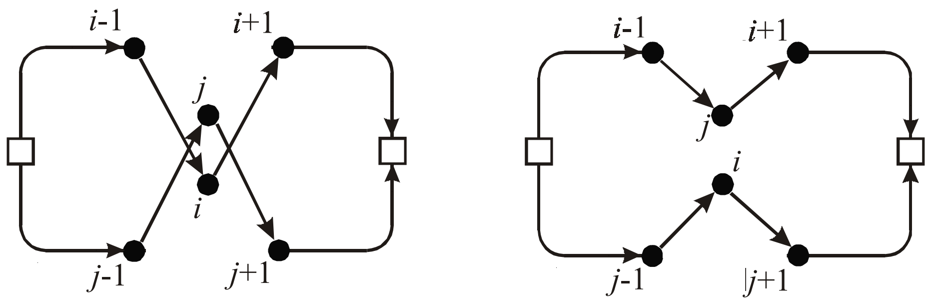

Figure 5-5: The exchange operator. The edges (*i*-1, *i*), (*i*, *i*+1), (*j*-1, *j*) and (*j*, *j*+1) are replaced by (*i*-1, *j*), (*j*, *i*+1), (*j*-1, *i*) and (*i*, *j*+1), i.e., two customers from different routes are simultaneously placed into the other routes.

Thompson and Psaraftis (1993) propose a method based on the concept of cyclic *k*-transfers that involves transferring simultaneously *k* demands from route I^j to route $I^{\delta(j)}$ for each *j* and fixed integer *k*. The set of routes $\{I^r\}$, *r*=1,…,*m* constitutes a feasible solution and δ is a cyclic permutation of a subset of {1,…,*m*}. In particular, when δ has fixed cardinality *C*, we obtain a *C*-cyclic *k*-transfer. By allowing *k* dummy demands on each route, demand transfers can be performed among permutations rather than cyclic permutations of routes. Due to the complexity of the cyclic transfer neighborhood search, it is performed heuristically. A general methodology developed by Thompson and Orlin (1989) is used for searching cyclic transfer neighborhoods. They transform the search for negative cost cyclic transfers into a search for negative cost cycles in an auxiliary graph. Savelsbergh's 2-opt (1986) procedure is used to maintain local optimality of the routes at all times and the initial solutions are constructed using the I1 heuristic of Solomon. The 3-cyclic 2-transfer operator is illustrated in Figure 5-6.

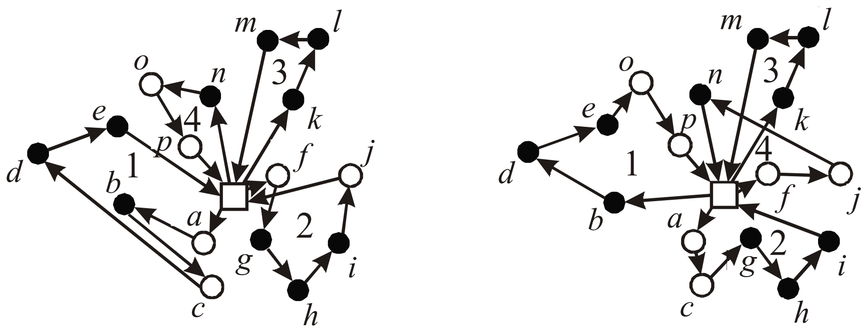

Figure 5-6: The cyclic transfer operator. The basic idea is to transfer simultaneously the customers denoted by white circles in cyclical manner between the routes. More precisely here customers *a* and *c* in route 1, *f* and *j* in route 2 and *o* and *p* in route 4 are simultaneously transferred to routes 2, 4, and 1 respectively and route 3 remains untouched.

Antes and Derigs (1995) propose a parallel construction approach that constructs and improves several tours simultaneously. The approach is based on the concept of negotiation between customers and tours. First, each unrouted customer requests a service cost from every tour and sends a proposal to the tour that offered the lowest price, and second, each tour selects the most efficient proposal. The prices are calculated according to Solomon's evaluation measures for insertion (heuristic I1). Once a feasible solution is constructed, the number of tours is reduced by one and the problem is resolved. The authors propose also a post-optimization approach, where some of the most inefficient customers are first removed from the tours and then reinserted using the negotiation procedure described above.

Russell (1995) embeds global tour improvement procedures within the tour construction process. The construction procedure used is similar to that in Potvin and Rousseau (1993). *N* seed points representing fictious customers are first selected using the seed point generation procedure of Fisher and Jaikumar (1981), originally proposed for the classical VRP. The basic idea is to use vehicle capacity information to create sectors and decide the distance of the seeds from the depot within each sector. Three ordering rules are used to select next customer for insertion, namely earliest time window, farthest distance from depot and width of the time window augmented by distance from the depot. The local search method employs a scheme developed by Christofides and Beasley (1984). In this scheme a move is performed by deleting and reinserting four customer points close to each other. For each customer, the best two routes are first determined according to insertion cost of Solomon (1987) since it would be computationally intractable to evaluate all route assignments. This interchange procedure is applied after every *k* customers have been routed. This approach is compared to the *k*-opt multiple tour branch exchange heuristic of Russell (1977). The author concludes that the hybrid approach of embedding improvement into the construction procedure is superior compared to the traditional two-phase approach, i.e., route construction followed by solution improvement.

Thangiah et al. (1995) examines the vehicle routing problems with time deadlines (VRPTD), i.e., without earliest time window. 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 proposed two heuristics perform well for problems in which the customers are tightly clustered or have long deadlines.

Hamacher and Moll (1996) describe a heuristic for real life VRP's with narrow time windows in the context of delivery of groceries to restaurants. The algorithm is divided into two parts. In the clustering step, the customers are partitioned into regionally bounded sets using the structure of the Minimal Spanning Tree (MST). The MST is divided into subtrees, where nodes of each subtree represent the customers belonging to one tour. Several weight functions based on number of customers, distance, total demand and time window types are used to determine whether a subtree leads to a cluster. Then customers within these sets are routed using a simple cheapest insertion algorithm followed by a local improvement, which cuts out pieces of the tour and inserts them back at another feasible location within the same tour. If a feasible solution is not found, the remaining unrouted customers are scheduled manually.

Shaw (1997) describes a Large Neighborhood Search (LNS) based upon rescheduling selected customer visits using Constraint Programming techniques. LNS is analogous to the shuffling technique used in job-shop scheduling (see for example Applegate and Cook, 1991), which is itself inspired from the shifting bottleneck procedure by Adams et al. (1988). The search operates by choosing in a randomized fashion a set of customer visits. The selected customers are removed from the schedule, and then reinserted at optimal cost. To create opportunity for interchange of customer visits between routes, the removed visits are chosen so that they are related. Here the term related refers to customers that are geographically close to each other, served by the same vehicle, have similar demand for goods and similar starting times for service. Then a branch and bound method coupled with Constraint Programming is used to reschedule removed visits. In the initial solution, each customer is served by a separate vehicle. Due to high computational requirements, this approach can be applied only to problems where the number of customers per route is relatively low.

Shaw (1998) uses an LNS approach similar to Shaw's (1997) above for solving vehicle routing problems. The basic difference is the usage of constraint based Limited Discrepancy Search (LDS) in the reinsertion of customers within the branch and bound procedure. For more details about LDS, see Harwey and Ginsberg (1995). The number of visits to be removed is increased during the search each time a number of consecutive attempted moves have not resulted in an improvement of the cost. LDS is used to explore the search tree in order of an increasing number of discrepancies, a discrepancy being a branch against the best insertion places. For instance, a single discrepancy would consist in inserting a customer at its second cheapest position.

Cordone and Wolfler-Calvo (1998) propose a deterministic heuristic based on classical *k*-opt exchanges combined with a procedure to reduce the number of routes. The special feature of the algorithm is that it alternates between minimization of total distance and total route duration to escape from local minima. The algorithm builds a set of initial solutions using Solomon's insertion heuristic I1, applies a local search procedure (exchanges 2 or 3 arcs) to each of them, and chooses the best one. The route reduction procedure tries to insert each customer of one route at a time into another route. If simple insertion fails, a simple ejection chain is tried, where a customer c_j is first removed from the target route r_n and inserted into some other route r_m, before inserting the current customer c_i into r_n. The authors use special implementation techniques to reduce the computation time. The first technique is based on so called macronodes. The macronode is a collapse of whatever sequence of nodes into a single one easier to handle. (see Cordone and Wolfler-Calvo, 1997). The other techniques are exploring the *k*-neighborhood in lexicographic order (for details see Savelsbergh 1986) and keeping in memory the best exchange for each route, each pair and each triplet of routes.

Caseau and Laburthe (1999) describe a heuristic specifically designed for large routing problems. The authors introduce an LDS variation to the parallel cheapest insertion heuristic that branches between the best and second best alternative routes for each customer if the differences in insertion costs are small. During solution construction, three moves are considered after each insertion, namely 2-opt*, reinsertion of a chain of consecutive customers from a route r into another route r', as well as a simple customer transfer move. When no feasible insertion place can be found, three different types of moves are considered to make room for the unrouted customer. The first move, swap, removes a chain of consecutive customers from r and inserts it into another route r'. The second move, relocate, removes a vertex from r, and inserts it into another route r', which may recursively require that another vertex is removed from r', etc., followed by reoptimization of each route concerned by the move. The last move, flush and relocate, first removes from r all nodes that can be directly relocated into another route, before trying to insert customer c_i. In cases where the number of customers on a route is less than 30, the order of the customers within the route is optimized using the exact constraint-based branch-and-bound algorithm by Caseau and Laburthe (1997). Otherwise, in case of longer routes, 3-opt is used to modify routes after each insertion. The authors also try to restrict the customers included in each route to a particular geometric zone.

Hong and Park (1999) propose a two-phase heuristic algorithm that consists of a parallel insertion method for clustering and a sequential linear goal programming procedure for routing. The primary criterion for the algorithm is the minimization of total traveled distance instead of number of vehicles and the second criterion is minimization of total customer waiting time. The seed customers are selected by identifying customers that cannot be served on the same route due to time or vehicle constraints. The remaining customers are inserted into these initialized tours so that the increase in route distance and waiting time is minimal. Similar to Potvin and Rousseau (1993), customers with a small number of feasible insertion locations and a large difference between the best and next best insertion places are considered for clustering first (regret measure). At the end of the clustering stage, groups are reformed using Or-opt and 2-opt improvement procedures. In the routing stage, the goal-programming model is decomposed into two linear programming subproblems, where either total distance or waiting time is minimized first. The authors report slightly better results than Potvin and Rousseau (1993), though using longer computation time.

Bräysy (2001a) describes several local search heuristics using a new three-phase approach for the VRPTW. In the first phase, several initial solutions are created using route construction heuristics with different combinations of parameter values. In the second phase, an effort is put to reduce the number of routes using a new ejection chain-based approach (Glover, 1991 and 1992) that considers also reordering of the routes. In the third phase, Or-opt exchanges are used to minimize total traveled distance. One of the construction heuristics borrows its basic ideas from the studies of Solomon (1987) and Russell (1995). Routes are built one at a time in sequential fashion and after k customers have been inserted into the route, the route is reordered using Or-opt exchanges. In addition, new seed selection schemes are introduced. The other heuristic draws its basic concepts from the Savings heuristic of Clarke and Wright (1964). Here a parallel version of the Savings heuristic is implemented, and the original measure of savings is modified to consider also changes in waiting times. Moreover, the customers in the combined route are reordered before evaluating the saving incurred by uniting the two routes.

Bräysy (2001b) suggests two heuristics specially designed for the clustered vehicle routing problems with time windows. The first approach is similar to Bräysy (2001a). The basic difference is the usage of four new local search operators in the third phase instead of Or-opt exchanges. These local search operators are based on modifications to CROSS-exchanges of Taillard et al. (1997) and cheapest insertion heuristics. The second approach is based on identifying customers within the same cluster by forming boxes around the selected seed customers. The customers within the boxes are then ordered using their time windows, cheapest insertion heuristic and Or-opt-exchanges. In addition, if some customers are not located in any box, an attempt is made to insert them into closest route using cheapest insertion heuristics. In the end, between-route customer relocations are also attempted. Experimental results indicate that the proposed procedures are efficient. The optimal solution is obtained for all 17 clustered test problems of Solomon (1987) in only about one second of computation time.

Table 2 summarizes some of the results obtained by described local search algorithms. We could not include all described algorithms in the table due to lack of information (not all authors report results properly or use Solomon's problem set). In Table 2, most of the algorithms are deterministic in nature. The only stochastic approaches are Russell (1995) and Shaw (1997 and 1998). Russell (1995) and Cordone and Wolfler-Calvo (1998) implemented their algorithm in Fortran, and Potvin and Rousseau (1995), Antes and Derigs (1995), Shaw (1998) and Caseau and Laburthe (1999) used C. Thompson and Psaraftis (1993), Prosser and Shaw (1996) and Shaw (1997) do not report the software used. The number of vehicles is considered as a primary optimization criterion by all authors except Prosser and Shaw (1996) where only the total distance of the routes is minimized. The secondary objective is total distance in Antes and Derigs (1995), Shaw (1997), Shaw (1998), Cordone and Wolfler-Calvo and Caseau and Laburthe (1999), while Thompson and Psaraftis (1993), Potvin and Rousseau (1995) and Russell (1995) optimize the total duration of routes.

Table 2: Local search algorithms. For each method two average results for Solomon's benchmarks are presented. The rightmost CNV and CTD indicate the cumulative number of vehicles and cumulative total distance over all test problems.

Author	R1	R2	C1	C2	RC1	RC2	CNV/CTD
(1) Thompson et al.	**13.00**	**3.18**	**10.00**	**3.00**	**13.00**	**3.71**	**438**
(1993)	1356.92	1276.00	916.67	644.63	1514.29	1634.43	68916
(2) Potvin et al. (1995)	**13.33**	**3.27**	**10.00**	**3.13**	**13.25**	**3.88**	**448**
	1381.9	1293.4	902.9	653.2	1545.3	1595.1	69285
(3) Russell (1995)	**12.66**	**2.91**	**10.00**	**3.00**	**12.38**	**3.38**	**424**
	1317	1167	930	681	1523	1398	65827
(4) Antes et al. (1995)	**12.83**	**3.09**	**10.00**	**3.00**	**12.50**	**3.38**	**429**
	1386.46	1366.48	955.39	717.31	1545.92	1598.06	71158
(5) Prosser et al. (1996)	**13.50**	**4.09**	**10.00**	**3.13**	**13.50**	**5.13**	**471**
	1242.40	977.12	843.84	607.58	1408.76	1111.37	58273
(6) Shaw (1997)	**12.31**	____	**10.00**	____	**12.00**	____	____
	1205.06		828.38		1360.40		
(7) Shaw (1998)	**12.33**	____	**10.00**	____	**11.95**	____	____
	1201.79		828.38		1364.17		
(8) Cordone et al. (1998)	**12.50**	**2.91**	**10.00**	**3.00**	**12.38**	**3.38**	**422**
	1241.89	995.39	834.05	591.78	1408.87	1139.70	58481
(9) Caseau et al. (1999)	**12.42**	**3.09**	**10.00**	**3.00**	**12.00**	**3.38**	**420**
	1233.34	990.99	828.38	596.63	1403.74	1220.99	58927
(10) Bräysy (2001a)	**12.17**	**2.82**	**10.00**	**3.00**	**11.88**	**3.25**	**412**
	1253.24	1039.56	832.88	593.49	1408.44	1244.96	59945

(1) PC/AT 12 MHz, 4 runs, 1.8 min., (2) Sparc workstation, number of runs not reported, 3.0 min., (3) PC/486/DX2 66 MHz, 3 runs, 1.4 min., (4) RS6000/530, 4 runs, 3.6 min., (5) Computational effort not reported , (6) DEC Alpha, 3 runs, 2 hours, (7) Sun Ultra Sparc 143 MHz, 6 runs, 1 hour, (8) Pentium 166 MHz, 1 run, 15.7 min., (9) Pentium 300 MHz, 1 run, 5 min., (10) Pentium 200 MHz, 1 run, 4.6 min.

According to Table 2, method Bräysy (2001a) is the best one with respect to solution quality. Moreover, one must note that in the paper by Bräysy (2001a) even better results are reported from testing other parameter combinations. The difference in the cumulative number of vehicles is about 14% compared to the worst method by Prosser and Shaw (1996). The reason for this can be found in the optimization criteria used: in Prosser and Shaw (1996), only the total distance of the routes is considered. Bräysy (2001a) dominates all other methods for all problem groups, except for the easy clustered problem groups C1 and C2, for which Shaw (1997), Shaw (1998), Caseau and Laburthe (1999) and Cordone and Wolfler-Calvo (1998) yield slightly better output respectively. However, these competing approaches require more computational resources. It should also be noted that, due to poor performance, Shaw (1997) and Shaw (1998) do not report the results for the problem groups R2, C2 and RC2. These two procedures are thus not comparable with other approaches in terms of robustness.

The differences in solution quality between the best three approaches by Bräysy (2001a), Caseau and Laburthe (1999) and Cordone and Wolfler-Calvo (1998) are quite small in most of the cases. For the CNV, the approach described in Bräysy (2001a) is about 2% better than the one in Cordone and Wolfler-Calvo (1998) or Caseau and Laburthe (1999). On the other hand, in terms of CTD, Caseau and Laburthe (1999) and Cordone and Wolfler-Calvo (1998) report results about 1.7% and 2.5% better than Bräysy (2001a) respectively, which illustrates the conflicting nature of the two objectives.

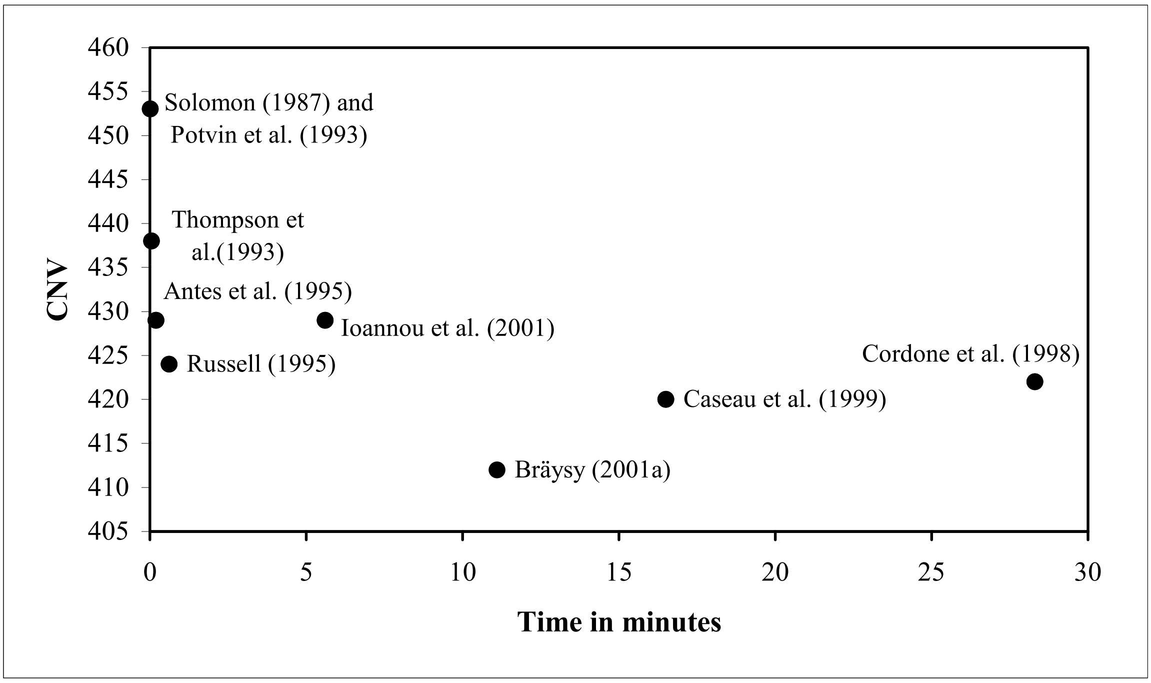

Figure 5-7: The efficiency of the described methods. The notation CNV refers to the cumulative number of vehicles required to solve all 56 test problems. Note that the time consumption of each method is normalized to Sun Sparc 10 using Dongarra's (1998) factors to facilitate the analysis.

The efficiency of the described methods is illustrated in Figure 5-7. We included in Figure 5-7 only approaches where a sufficient amount of information is provided by the authors. At least the computer, number of computational runs as well as the time consumption and number of vehicles must be reported. From Figure 5-7 one can see, that difference in time consumption between Solomon (1987), Potvin and Rousseau (1993), Thompson and Psaraftis (1993) and Antes and Derigs (1995) is quite small. Therefore only Antes and Derigs (1995), Russell (1995) and Bräysy (2001a) can be considered as Pareto optimal in terms of solution quality and time consumption. There is no clear rule to determine which Pareto optimal approach is the best. The choice depends on the preferences of the decision maker. The methods by Antes and Derigs (1995) and Russell (1995) are a lot faster than the one in Bräysy (2001a), but they fall behind in solution quality.

6 Conclusions

The vehicle routing problem with time windows is one of the classical research areas in operations research with considerable economic significance. The high complexity of the VRPTW requires heuristic solution strategies for most real-life instances. The research on approximation methods has, over the years, produced a wide variety of heuristic approaches for the VRPTW. In this paper, methods based on classical solution construction and improvement techniques were comprehensively reviewed. For comprehensive survey on metaheuristics for the VRPTW, see Bräysy and Gendreau (2001).

VRPTW heuristics are usually measured against two criteria: solution quality in terms of objective function value and speed. In our opinion simplicity of implementation, flexibility and robustness are also essential attributes of good heuristics. By flexibility we mean the ability to accommodate the various side constraints encountered in a majority of real-life applications. As for robustness, an algorithm should still able to produce results under difficult circumstances such as when a problem instance is pathological. These issues, as well as the question, how to evaluate heuristics, are discussed in Section 3.

Recent composite heuristics were found to perform best in terms of solution quality, the most efficient being those of Russell (1995) and Bräysy (2001a). These methods provide better results than earlier simple heuristics, while being still quite fast. As heuristics need to be especially effective for very large-scale problems, we expect work on these to intensify.

Acknowledgements

This work was partially supported by the Emil Aaltonen Foundation, the Canadian Natural Science and Engineering Research Council and TOP programme funded by the Research Council of

Norway. This support is gratefully acknowledged. We would also like to thank Dr. Geir Hasle (SINTEF Applied Mathematics, Norway) for his valuable comments.

7 References

Adams, J., E. Balas and D. Zawack (1988), "The Shifting Bottleneck Procedure for Job Shop Scheduling", *Management Science* 34, 391–401.

Antes, J. and U. Derigs (1995), "A New Parallel Tour Construction Algorithm for the Vehicle Routing Problem with Time Windows", Working Paper, Department of Economics and Computer Science, University of Köln, Germany.

Applegate, D and W. Cook (1991), "A Computational Study of the Job-Shop Scheduling Problem", *ORSA Journal On Computing* 3, 149–156.

Atkinson, J. B. (1994), "A Greedy Look-Ahead Heuristic for Combinatorial Optimisation: An Application to Vehicle Scheduling with Time Windows", *Journal of the Operational Research Society* 45, 673–684.

Backer de, B., V. Furnon, P. Prosser, P. Kilby and P. Shaw (1997), "Local Search in Constraint Programming: Application to the Vehicle Routing Problem", presented at the CP-97 Workshop on Industrial Constraint-based Scheduling, Schloss Hagenberg, Austria.

Baker, E.K. and J.R. Schaffer (1986), "Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints", *American Journal of Mathematical and Management Sciences* 6, 261–300.

Balakrishnan, N. (1993), "Simple Heuristics for the Vehicle Routeing Problem with Soft Time Windows", *Journal of the Operational Research Society* 44, 279–287.

Barr, R. S., B. L. Golden, J. P. Kelly, M. G. C. Resende and W. R. Stewart (1995), "Designing and Reporting on Computational Experiments with Heuristic Methods", *Journal of Heuristics* 1, 9–32.

Bramel, J. and D. Simchi-Levi (1996), "Probabilistic Analyses and Practical Algorithms for the Vehicle Routing Problem with Time Windows", *Operations Research* 44, 501–509.

Bräysy, O. (2001a), "Five Local Search Algorithms for the Vehicle Routing Problem with Time Windows", Working Paper, SINTEF Applied Mathematics, Department of Optimisation, Norway.

Bräysy, O. (2001b), "Two Heuristics for the Clustered Vehicle Routing Problems with Time Windows", *Proceedings of the University of Vaasa, Discussion papers* 280.

Bräysy, O. and M. Gendreau (2001), "Metaheuristics for the Vehicle Routing Problem with Time Windows", Internal Report STF42 A01025, SINTEF Applied Mathematics, Department of Optimisation, Norway.

Caseau, Y. and F. Laburthe (1997), "Solving Small TSPs with Constraints", in *Proceedings of the 14th International Conference on Logic Programming*, L. Naish (ed), 316–330, MIT Press, Cambridge.

Caseau, Y. and F. Laburthe (1999), “Heuristics for Large Constrained Vehicle Routing Problems”, *Journal of Heuristics* 5, 281–303.

Christofides, N. and J. Beasley (1984), “The Period Routing Problem”, *Networks* 14, 237–246.

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., M. Gendreau, G. Laporte, J.-Y. Potvin and F. Semet (2001b), “A Guide to Vehicle Routing Heuristics”, Publication CRT-2001-23, University of Montreal, Canada.

Cordone, R. and R. Wolfler-Calvo (1997), “A Note on Time Windows Constraints in Routing Problems”, Internal Report, Department of Electronics and Information, Polytechnic of Milan, Milan, Italy.

Cordone, R. and R. Wolfler-Calvo (1998), “A Heuristic for the Vehicle Routing Problem with Time Windows”, Internal Report, Department of Electronics and Information, Polytechnic of Milan, Milan, Italy. To appear in Journal of Heuristics.

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

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.

Dullaert, W. (2000a), “Impact of Relative Route Length on the Choice of Time Insertion Criteria for Insertion Heuristics for the Vehicle Routing Problem with Time Windows”, in *Proceedings of the Rome Jubilee 2000 Conference Improving Knowledge and Tools for Transportation and Logistics Development: 8th Meeting of the Euro Working Group Transportation,* Faculty of Engineering, “La Sapienza”, University of Rome, Italy, B. Maurizio (ed.), 153–156, Rome.

Dullaert, W. (2000b), “A Sequential Insertion Heuristic for the Vehicle Routing Problem with Time Windows with Relatively Few Customers Per Route”, Research paper, Faculty of Applied Economics UFSIA-RUCA; 2000:014, Antwerpen, Belgium.

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

Foisy, C. and J.-Y. Potvin. (1993), “Implementing an Insertion Heuristic for Vehicle Routing on Parallel Hardware. *Computers and Operations Research* 20, 737–745.

Gillett, B. and L.R. Miller (1974), “A Heuristic Algorithm for the Vehicle Dispatch Problem”, *Operations Research* 22, 340–349.

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.

Hamacher, A. and C. Moll (1996), “A New Heuristic for Vehicle Routing with Narrow Time Windows”, in *Operations Research Proceedings* 1996, *Selected papers of the symposium* (*SOR*`96), Braunschweig, Germany, September 3-6, U. Derigs, W. Gaul, R H. Möhring and K.-P. Schuster (eds), 301–306, Springer Verlag, New York.

Harwey, W. and M. Ginsberg (1995), “Limited Discrepancy Search”, in *Proceedings of the* 14th *IJCAI*, T. Dean (ed), 607–615, Morgan Kaufmann, San Francisco.

Hong, S.-C. and Y.-B. Park (1999), “A Heuristic for Bi-objective Vehicle Routing with Time Window Constraints”, *International Journal of Production Economics* 62, 249–258.

Hooker, J. N. (1995), “Testing Heuristics: We Have It All Wrong”, *Journal of Heuristics* 1, 33–42.

Ioannou, G., M. Kritikos and G. Prastacos (2001), “A Greedy Look-Ahead Heuristic for the Vehicle Routing Problem with Time Windows”, *Journal of the Operational Research Society* 52, 523–537.

Jaffar, J. and J.-L. Lassez (1986), “Constraint Logic Programming”, Technical report 86/73, Department of Computer Science, Monash University, Australia.

Jaffar, J. and M.J. Maher (1994), “Constraint Logic Programming: A Survey”, *Journal of Logic Programming* 19/20, 503–581.

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

Knight, K. and J. Hofer (1968), “Vehicle Scheduling with Timed and Connected Calls: A Case Study”, *Operational Research Quarterly* 19, 299–310.

Kohl, N. (1995), “Exact Methods for Time Constrained Routing and Related Scheduling Problems”, Ph.D. thesis, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Kohl, N., J. Desrosiers, O.B.G. Madsen, M.M. Solomon and F. Soumis (1999), “2-path Cuts for the Vehicle Routing Problem with Time Windows”, *Transportation Science* 33, 101–116.

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.

Lin, S. (1965), "Computer Solutions of the Traveling Salesman Problem", *Bell System Technical Journal* 44, 2245–2269.

Madsen, O.B.G. (1976), "Optimal Scheduling of Trucks – a Routing Problem with Tight Due Times for Delivery", in *Optimization Applied to Transportation Systems*, H. Strobel, R. Genser and M. Etschmaier (eds), 126–136, International Institute for Applied System Analysis, Laxenburgh.

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.

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.

Prosser, P. and P. Shaw (1996), "Study of Greedy Search with Multiple Improvement Heuristics for Vehicle Routing Problems", Working Paper, University of Strathclyde, Glasgow, Scotland.

Pullen, H. and M. Webb (1967), "A Computer Application to a Transport Scheduling Problem", *Computer Journal* 10, 10–13.

Russell, R. (1977), "An Effective Heuristic for the M-tour Traveling Salesman Problem with Some Side Conditions", *Operations Research* 25, 517–524.

Russell, R.A. (1995), "Hybrid Heuristics for the Vehicle Routing Problem with Time Windows", *Transportation Science* 29, 156–166.

Savelsbergh, M.W.P. (1986), "Local Search in Routing Problems with Time Windows", *Annals of Operations Research* 4, 285–305.

Savelsbergh, M.W.P. (1990), "An Efficient Implementation of Local Search Algorithms for Constrained Routing Problems", *European Journal of Operational Research* 47, 75–85.

Savelsbergh, M.W.P. (1992), "The Vehicle Routing Problem with Time Windows: Minimizing Route Duration", *Journal on Computing* 4, 146–154.

Shaw, P. (1997), "A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems", Working Paper, Department of Computer Science, University of Strathclyde, Glasgow, Scotland.

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. (1986), “On the Worst-Case Performance of Some Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints”, *Networks* 16, 161−174.

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

Solomon, M.M., E.K. Baker and J.R. Schaffer (1988), “Vehicle Routing and Scheduling Problems with Time Window Constraints: Efficient Implementations of Solution Improvement Procedures”, in *Vehicle Routing: Methods and Studies*, B. Golden and A. Assad (eds), 85–106, Elsevier Science Publishers, Amsterdam.

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

Taillard, È. D. (2001), “Comparison of Non-Deterministic Iterative Methods”, Presented at MIC’2001, 4th Metaheuristics International Conference, July 16th-20th, Porto, Portugal.

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.

Thompson, P. M. and J. B. Orlin (1989), “Theory of Cyclic Transfers”, Working Paper, Operations Research Center, MIT, Cambridge.

Thompson, P. M. and H.N. Psaraftis (1993), “Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems”, *Operations Research* 41, 935−946.

Van Landeghem, H.R.G. (1988), “A Bi-criteria Heuristic for the Vehicle Routing Problem with Time Windows”, *European Journal of Operations Research* 36, 217−226.