Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem

Alex Van Breedam
University of Antwerp
KPMG Orinoco
E-mail: Alex.VanBreedam@kpmg.be

Scope and Purpose. Tabu Search (TS) and Simulated Annealing (SA) have demonstrated to be appropriate metaheuristics for solving NP-hard combinatorial optimization problems, such as the Vehicle Routing Problem with side-constraints. In order to compare the performances of both metaheuristics with each other and with a traditional descent implementation, a comparison of the best solution independent of computing times is fundamentally wrong because metaheuristics have no unambiguous stopping criteria, as opposed to traditional descent implementations.

Abstract. Three improvement heuristics for the vehicle routing problem are considered: a descent heuristic and two metaheuristics Simulated Annealing and Tabu Search. In order to make an in-depth comparison of the performance of these improvement heuristics, their behavior is analysed on a heuristic, time-sensitive level as well as on a parametric level. The design and the results of the experiments are outlined. The external validity of the conclusions is discussed.

1 Introduction

The Vehicle Routing Problem (VRP) can be defined as the problem of finding a set of routes for a fleet of vehicles which have to service a number of stops $|N|$. The vehicles depart and arrive at a single depot. The fleet of vehicles is assumed to be homogeneous with capacity Q. The demand quantity of each stop, q_i, is deterministic. No single demand exceeds the vehicle capacity. This corresponds to the description of the standard VRP. However, the VRP can be extended with various side-constraints, as there are mixed pick-ups (backhaules) and deliveries (linehaules) and time-windows. Lenstra and Rinnooy Kan (1981) showed that the VRP with side-constraints is a NP-hard combinatorial problem. Hence, exact algorithms are only useful for tiny problems. For problems of more realistic size, heuristics are more appropriate.

An initial solution to the VRP can be obtained with a wide variety of heuristics. Principally, three main categories of heuristics can be distinguished: route-construction heuristics, two-phase heuristics and heuristics based on exact algorithms.

An initial feasible solution to the VRP can be enhanced through the application of an improvement heuristic. This type of heuristic is aimed at improving a feasible solution by moving (relocating and/or exchanging) stops within or between routes.

Within routes improvement heuristics are typically used for the Travelling Salesman Problem (TSP) and are based on the *k*-opt and *Or*-opt principles (Croes (1958), Lin (1965), Lin and Kernighan (1973), Or (1976)).

Between routes improvement methods have been conceived for the VRP. They improve a feasible solution by moving stops between different routes.

Besides the type of improvement heuristic, a second classification criterion is the search strategy. The search strategy is a procedure which indicates the order in which new solutions are searched. Among the VRP improvement heuristics, two groups can be distinguished based on the search strategy: local and global optimisation methods.

The local optimisation heuristic, the traditional "descent" method, finds a local minimum only by performing moves of stops which result in the improvement of the objective function value.

The search strategy of these local optimisation heuristics is blind. This means that the order by which new solutions are generated is only dependent on the information gathered during the execution of the heuristic (Osman (1991)).

These heuristics are halted if no further improvement of the objective function value is possible. As a matter of fact, local optimisation heuristics are trapped in the local optimum in which they descend.

Global optimisation heuristics, on the contrary, succeed in leaving the local optimum by temporarily accepting moves which cause a worsening of the objective function value. These heuristics are often called "metaheuristics" because the procedure used to generate a new solution out of the current one, is embedded in a heuristic which determines the search strategy. The search strategy for the global optimisation heuristics is a directed search. This implies that information of the problem domain and the nature of the objective is used to direct the search procedure towards promising regions (Osman (1991)).

The main drawback of metaheuristics is that they have no stopping criterion defined. The longer the computing time, the higher the probability of finding the global minimum.

Simulated Annealing (SA) and Tabu Search (TS) are probably the most popular metaheuristics. Other metaheuristics are Genetic Algorithms, Neural Networks, Ant colonies, Great Deluge, Noise method, ... Most of these metaheuristics are based on principles of physical or biological processes. For references on metaheuristics the reader is referred to Osman and Laporte (1996) and to Laporte and Osman (1995), and Gendreau et al. (1998) for applications on routing problems.

The main contribution of this article is two-fold: a parametric and a heuristic analysis with three improvement heuristics: a descent, a SA and a TS heuristic. In order to preserve the comparability of the heuristics, the three heuristics use the same move types. Section 2 gives a detailed description of the move types together with the three heuristics and their parameters.

All experiments are performed on a test set of fifteen problem instances with vehicle-related, customer-related and time-related side constraints. This set is presented in section 3.

The parametric analysis is a statistical analysis to investigate the significance of the effect of each parameter of a heuristic. Therefor the Automatic Interaction Detection (AID) technique is used. The parametric analysis is reported in section 4.

Section 5 describes the results of the heuristic analysis. This analysis is a dynamic, time-sensitive analysis where the solution values of the three heuristics are compared at various points in time.

The rationale behind the research is quite simple and inspired by the practitioner's interest. Basically, he is not concerned with heuristics which perform just a few kilometers better on some classical benchmark problems for two reasons. First, because these problems are not necessarily comparable to the type of problems he is confronted with. Second, because these best solutions have been obtained by unique combinations of parameters, obtained after a whole lot of trials. The practitioner really wants to know why one heuristic performs better relatively, as compared to other heuristics, and absolutely, as compared to other combinations of parameters of that heuristic.

2 Description of the heuristics

2.1 Types of move

The three improvement heuristics all use a same type of move. A move can be defined as the mutation of stops between routes in order to obtain a neighborhood solution out of an existing solution.

It is assumed that only feasible moves with respect to the side-constraints are performed.

Four types of moves are considered. They are called the String Cross, the String Exchange, the String Relocate and the String Mix by convention.

String Cross. The String Cross (SC) is a move in which two strings of stops are exchanged by crossing two arcs of two different routes. An example of a SC move is given by figure 1. The concept of this type of move has also been presented by Savelsbergh (1988) and Potvin et al. (1992).

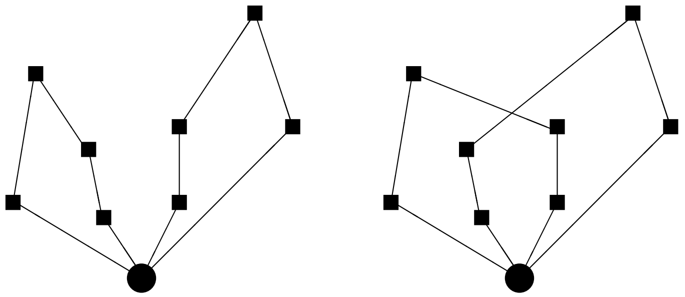

Figure 1: Example of a String Cross.

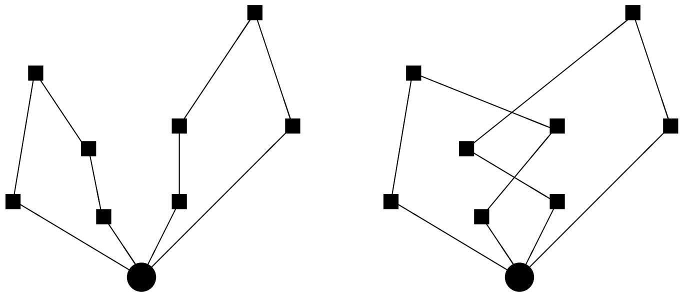

Figure 2: Example of a String Exchange.

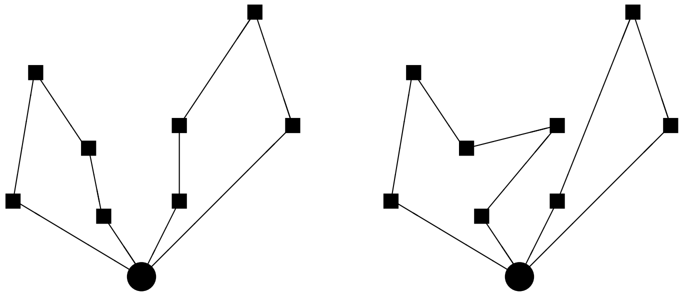

Figure 3: Example of a String Relocation.

String Exchange. The move of type String Exchange (SE) generates a neighborhood solution by exchanging two strings of stops between two routes. Symbolically, this can be represented by $(x1, x2)$, where $x1$ and $x2$ are integers representing the length of the strings to be exchanged in both routes. If K denotes the maximum length of a string of stops, then $1 \leq x1 \leq K$ and $1 \leq x2 \leq K$ must be satisfied. The length of the strings $x1$ and $x2$ is not necessarily equal. Figure 2 contains an example of an SE move.
A similar type of move has been defined by Dror and Levy (1986) and Savelsbergh (1988). The two most common values for the maximum string length are $K = 1$ and $K = 2$.

String Relocation. The String Relocation (SR) can be described as the move of a string of stops from one route to another. Symbolically, this can be represented by $(x, 0)$ or $(0, x)$. The maximum number of stops to be relocated is bounded by a parameter called the maximal string length K, i.e $1 \leq x \leq K$. This type of move is able to reduce the number of routes. See figure 3 for an example of a SR.
Dror and Levy (1986) and Savelsbergh (1988) proposed a similar type of move. The values $K = 1$ and $K = 2$ are commonly used.

String Mix. The String Mix (SM) is a mixture of the String Exchange and the String Relocate. When implemented, this move type selects the best between the String Exchange and the String Relocation. The number of routes can be reduced through the moves of the type String Relocation.

Another variant of a move generation mechanism is the λ-interchange mechanism described in Osman (1993). This mechanism considers a relocation as a special case of an exchange.

2.2 Descent heuristic

The descent heuristic (LI) is very popular due to its simplicity and its relatively short computing time. Most published implementations of this heuristic for the VRP are similar. This implementation is aimed at evaluating the effect of a number of parameters on the final solution.

Procedure

Step 1: The initial solution is the current solution.

Step 2: Select the first move by selecting the first pair of routes of the current solution, based on their order number. Select the stops required for the type of move in both routes based on the route sequence.

Step 3: If the move is unfeasible or does not improve the objective function value, then go to step 5.

Step 4: If the selection strategy (cfr. infra) is first improvement, then perform the move. The new solution becomes the current solution. Go to step 2.
If the selection strategy is the best improvement, and the neighborhood solution is better than all previous solutions, then save the neighborhood solution and go to step 6.

Step 5: If an entire evaluation cycle has been completed, and the selection strategy is direct improvement, then the heuristic is stopped.
If an entire evaluation cycle has been completed, and the strategy is the best improvement, and the best neighborhood solution has been saved, then this neighborhood solution becomes the current solution. Go to step 2. If no improved neighborhood solution has been saved during a complete evaluation cycle for the best improvement strategy, then the heuristic is stopped.

Step 6: Select the next move by selecting the next stops in the route or, if needed, the next pair of routes to be evaluated. If no move can be selected go to step 5, else go to step 3.

Parameters

Initial solution (S). This parameter is provided especially for evaluating the effect of the initial solution on the final solution. The initial solution is generated with an initial heuristic.
Values assigned to this parameter are:
1. A bad initial solution.
2. A good initial solution.

String length (K). This parameter determines the maximum length of a string of stops to be moved with move types SR, SR or SM.
Values assigned to this parameter are:
1. String length of 1 stop.
2. String length 2 stops.

Selection strategy (P). This parameter guides the choice of the next move to be performed.
Values assigned to this parameter are:
1. First improvement: the first improving move is performed.
2. Best improvement: the best improving move after completion of an evaluation cycle is performed.

Evaluation procedure for string length K > 1 (F). The order in which moves of string of two stops are evaluated. This parameter is irrelevant if the move type is SC.
Values assigned to this parameter are:
1. Evaluation of all possible string lengths between a pair of routes before selecting the next pair.
2. The string length is increased by one after having completed an evaluation cycle without improvement.

The initial solution S and the string length K can be considered as problem-specific parameters. The other three parameters are of a more generic nature.

2.3 Simulated Annealing metaheuristic

The principle of the SA metaheuristic is deduced from the physical annealing process of solids. Kirckpatrick et al. (1983) and Cerny (1985) proposed the use of SA for combinatorial problems. Their work is based on the research of Metropolis et al. (1953) in the field of Statistical Mechanics. For an overview of the research and applications of SA, the reader is referred to Van Laarhoven and Aarts (1987), Aarts and Korst (1989), Collins et al. (1988) and Eglese (1990).

As far as our implementation is concerned, the following choices have been made. In order to determine the value of the *initial temperature*, we use the approach proposed by Johnson et al. (1989). The initial temperature T_{begin} is computed by solving the expression

$$P_a = e^{-\Delta C / T_{begin}}$$

and hence

$$T_{begin} = \frac{-\Delta C}{ln P_a} \tag{1}$$

Here ΔC represents the average deterioration value, which is computed as the cumulated value of the values of all worsening moves possible from the initial solution, divided by the number of moves which cause a deterioration of the objective function value. Parameter P_a represents the acceptance fraction, i.e. the ratio of the accepted to the total number of generated moves.

The *cooling function* we use for the reduction of the temperature is the simple geometric function. The temperature at iteration t, T_t, is obtained from the temperature of the previous iteration as follows:

$$T_t = R \cdot T_{t-1} \tag{2}$$

Here, R represents the cooling rate.

The principle of an epoch is used to determine the *thermal equilibrium* at each temperature (see Golden and Skiscim (1986) and Teodorovic and Pavkovic (1992)). Therefore, the value of the current solution is stored every s moves. If the difference between the value of the last epoch and all previously saved epochs is less than t%, then thermal equilibrium is reached and the temperature can be decreased according to the cooling function 2. Here, the values of s and t are both set to 10.

The *stopping criterion* is satisfied if the percentage of accepted moves is inferior to a critical acceptance ratio for a predefined number of five consecutive temperature values. However, each time a new best solution is obtained, the counter is reset to zero. This stopping criterion has been proposed by Johnson et al. (1989).

In contradiction to the complete evaluation of moves performed in the implementations for the LI and TS heuristics proposed, the SA heuristic generates a neighborhood solution on a stochastic base. This implies that the routes and the stops required to perform a move are selected at random. In order to narrow the search space, we provide the option of a range delimiter to prevent the selection of too many bad moves with respect to the objective function value.

The range delimiter is a travel time restriction between the stops of two different routes selected at random for the move. This travel time restriction is adapted to the problem considered. It is computed for each solution with K routes $R_1, ..., R_K$, first by determining for each stop $i \in N \setminus \{0\}$, belonging to route R_k, the travel time to its nearest neighbor stop belonging to another route R_l.

$$NN_{i \in R_k} = \min_{j \in R_l, k \neq l} d_{ij} \tag{3}$$

Subsequently, the range delimiter D is set equal to the largest travel time of the set of travel times previously defined.

$$D = \max_{i \in N \setminus \{0\}} NN_i \tag{4}$$

This procedure assures the potential participation of each stop to a move.

Other implementations of SA for the VRP are proposed in the literature. Osman (1993) has proposed a hybrid SA/TS metaheuristic. This hybrid metaheuristic was adapted by Thangiah et al. (1994) to solve VRPs with time-windows and combined pick-ups and deliveries. In Teodorovic and Pavkovic (1992) SA is used for generating an initial solution as well as for improving it in the case of the VRP with stochastic demand. Robusté et al. (1990) and Alfa et al. (1991) integrated SA in an initial heuristic for the VRP.

The SA implementation for the VRP proposed by Van Breedam (1995) and Janssens and Van Breedam (1995) is for the greater part comparable to the one presented here. The good solutions obtained by the last authors for some classical benchmark problems confirm the quality of the SA heuristic proposed here.

Procedure

Step 1: The initial solution is the current solution and is saved.

Step 2: Determine the initial temperature T_{begin} by means of expression 1, taking account of the value for the acceptance ratio P_a. The current temperature T is set to the initial temperature T_{begin}.

Step 3: Generate a neighborhood solution by randomly selecting a move from the current solution.

Step 4: Compute δ as the difference in objective function value (total travel time) between the neighborhood solution and the current solution.
If $\delta > 0$ and $r \geq e^{-\delta/T}$, with r a pseudo-random number in [0,1], then go to step 3.

Step 5: The neighborhood solution becomes the current solution. The neighborhood solution is saved if it is the best solution so far.
Check for thermal equilibrium, if required. Every 10 moves the value of the current solution, an epoch, is compared to all previously saved epoch values. If the deviation is superior to 10%, then no equilibrium is reached and go to step 3.

Step 6: Decrease the current temperature T by means of the cooling function 2. If the percentage of accepted moves is inferior to the critical acceptance ratio for more than 5 consecutive temperature reductions without obtaining a new best solution, then go to step 7 else go to step 3.

Step 7: The solution stored is the final best solution.

Parameters

Initial solution (S). This parameter is provided especially for evaluating the effect of the initial solution on the final solution. The initial solution is generated with one of the initial heuristics.
Values assigned to this parameter are:
1. A bad initial solution.
2. A good initial solution.

String Length (K). This parameter determines the maximal length of a string to be moved if the move type is String Exchange, String Relocation or String Mix.
Values assigned to this parameter are:
1. String length of 1 stop.
2. String length of 2 stops.

Range delimiter (D). The range delimiter is aimed at narrowing the search space for neighborhood solutions and is computed as the largest travel time between two nearest neighbors of two different routes of the current solution. When implemented, the range delimiter assures the potential participation of each stop to a move.
Values assigned to this parameter are:
1. No range delimiter.
2. Range delimiter.

Acceptance fraction (A). This is the percentage of accepted moves obtained when performing a full move cycle on the initial solution. This parameter is used to fix the initial temperature.
Values assigned to this parameter are:
1. 0.30
2. 0.50

Cooling rate (R). This is the fraction by which the temperature is reduced in the geometric temperature function (2).
Values assigned to this parameter are:
1. 0.70 (fast cooling).
2. 0.90 (slow cooling).

Critical acceptance ratio (L). Critical percentage of accepted moves beneath which the percentage of accepted moves has to drop for more than 5 temperature reductions without an improvement of the best solution, before the SA heuristic is stopped.
Values assigned to this parameter are:
1. 0.01
2. 0.05

The initial solution, the string length and the range delimiter D can be considered as problem-specific parameters. The other three parameters are of a more generic nature. Their values are chosen based on extreme but acceptable values published in literature.

2.4 Tabu Search metaheuristic

Tabu Search has been conceived by Glover (1986). Similar ideas were developed by Hansen (1986) who has proposed a steepest ascent/mildest descent heuristic.
TS is based on the principles of intelligent problem solving. A fundamental element is the use of a flexible and dynamic memory structure i.e. the tabu list.
The procedure of TS is simple. At each iteration, the best move is selected. If this move deteriorates the objective function value, it is only performed if the inverse move does not have the tabu status, i.e. if it is not in the tabu list. If it is in the list, then the next best move not in the tabu list is selected and performed. This process is repeated until a stopping criterion is reached.

A good overview of TS and its applications is provided by Glover (1989), Glover (1990) and Glover and Laguna (1993).

With respect to the application of TS to the VRP, some publications are to be noticed. Pureza and França (1991), Osman (1991) and Osman (1993) use TS as an improvement heuristic with a type of move in which a relocation is considered as a special case of an exchange. Thangiah et al. (1994) implements a hybrid TS/SA metaheuristic for VRPs with time-windows using the λ-interchange mechanism described in Osman (1991). Taillard (1993) adds a diversification strategy to his implementation of a TS improvement heuristic for the VRP. In Semet and Taillard (1993) the use of a TS improvement heuristic is demonstrated for a practical case.
The implementation of Gendreau et al. (1994) is exceptional in so far that unfeasible solutions with respect to capacity and route-length constraints are temporarily allowed through the use of penalties. They use a move comparable to the SR type which has been embedded in a specific insertion heuristic.
Stewart et al. (1992) use a TS strategy for finding an initial solution through a repeated application of a generalized assignment heuristic.

The TS implementation proposed provides the option to use a static as well as a dynamic tabu list length. The tabu list contains records of three elements: the list position, the origin route and the string of stops moved. The list is implemented as a queue. At each iteration, the last move performed is added to the end of the list. Subsequently, the list is rearranged by removing the move at position 1 in the list and by pulling all subsequent moves one position forward in the list.
The main advantage of this type of list is its ease of adaptation to static and dynamic list lengths. Moreover, all possible lengths for the string of stops to be moved are allowed. Most existing TS implementations for the VRP (cfr. supra) of which we are aware cannot relocate or exchange more than one stop between routes. A drawback of this structure occurs when a move involves the exchange of two strings of stops, as it is the case with the SC and SE move type. Two separate list positions are required to administrate the move.

The TS implementation comes with a long-term memory to allow diversification. To prevent stops from being too frequently selected for moves, a penalty function value is computed for each move. This amount is added to the objective function value of the neighborhood solution that would be obtained if the move was performed. The penalty function value for a move is determined first by computing the penalty value of each stop involved in the move. The penalty associated with a stop is computed with the method proposed by Taillard

(1993). The penalty value for stop i is given by $W \cdot f_i$, in which f_i stands for the frequency of occurrence of a stop i in a move. A value for parameter W is obtained by selecting a value at random from the interval

$$(0.1 \cdot \Delta_t^{max}\sqrt{N \cdot k}\ ;\ 0.5 \cdot \Delta_t^{max}\sqrt{N \cdot k}) \tag{5}$$

Here, N and k represent the number of stops and the number or routes of the current solution, respectively. The factor $\sqrt{N \cdot k}$ can be considered as a normalising factor because the frequency of occurrence of a stop in a move decreases with the size of the problem. The term Δ_t^{max} represents the largest improvement up to iteration t.
The use of a penalty function related to a stop is not appropriate for our implementation because the moves (2,0) or (2,2) would be a priori disfavored with respect to the moves (1,0) or (1,1). Consequently, we decided to associate a penalty function value with a move rather than with a stop. The penalty of a move is computed as the average of the penalties of all stops involved in a move.

The stopping criterion of our implementation is arbitrarily set to 500 iterations. This number is of secondary importance because the objective function value as a function of the computing time will be used as a basis for the heuristic analysis.

Procedure

Step 1: The initial solution is the current solution.

Step 2: Perform a complete evaluation cycle of moves of the current solution. Select the routes according to the route numbers and the stops according to the route sequence.

Step 3: If the best move found gives an improvement of the objective function value, then go to step 5.

Step 4: If a long-term memory is used, an additional penalty associated with the move must be taken into account when selecting the least deteriorating move. If the best move is in the tabu list then select the move which causes the least deterioration of the objective function value and which is not in the tabu list.

Step 5: Perform the move. The new solution becomes the current solution. If the new solution is better than the best solution so far, then save the new solution. Add the move to the tabu list if there is still some place left. Otherwise, discard the move.

Step 6: Rearrange the tabu list by removing the first move from the list and pushing the successive moves up forward by one position. Adapt the length of the tabu list, if required.

Step 7: Go to step 2 if the number of iterations is less than 500.

Step 8: The solution saved is the final best solution.

Parameters

Initial solution (S). This parameter is provided especially for evaluating the effect of the initial solution on the final solution. The initial solution is generated with one of the initial heuristics.
Values assigned to this parameter are:
1. A bad initial solution.
2. A good initial solution.

String Length (K). This parameter determines the maximal length of a string to be moved if the move type is String Exchange, String Relocation or String Mix.
Values assigned to this parameter are:
1. String length of 1 stop.
2. String length of 2 stops.

Length of tabu list (L). In the case of a static tabu list length, only the lower bound of the list length interval is considered. For a dynamic list the length of the tabu list is randomly selected in the interval.
Values assigned to this parameter are:
1. 10-20
2. 20-30
3. 30-40

Iterations for adapting tabu list length (I). This parameter sets the number of iterations after which the length of the tabu list is changed.
Values assigned to this parameter are:
1. $+\infty$ (static tabu list length)
2. 5
3. 25
4. 50

Long-term memory (G). This parameter determines whether or not a long term memory is used.
Values assigned to this parameter are:
1. Implementation without long-term memory.
2. Implementation with long-term memory.

Two parameters, the initial solution S and the string length K are problem-specific parameters. The other parameters are more generic of nature. As for the SA implementation, the values chosen for the generic parameters are determined based on publications containing acceptable values.

3 Description of the test set of problems

All experiments are performed on a test set containing fifteen test problems. The test set has not been taken from literature. We are aware that a classical set of benchmark problems exist, but we intended to investigate systematic behavior of various aspects as there are geographical variation and vehicle- related, customer-related and time-related constraints. As the primary aim of this research is the evaluation of the behavior of improvement heuristics, our intention is by no means creating a new, additional test set of VRP instances but rather to ascertain ourselves that we analyse what we want to analyse.

In order to cope with systematic geographical variation a basic set of 60 test problems has been constructed. The set is constructed on the basis of three criteria: the location of the depot, the grouping of customers and the spreading of customers. Three different depot locations are considered: central, inside and outside. For the grouping of customers, five patterns are distinguished: singleton, clusters, 50% clusters, cones and 50% cones. Four patterns are chosen for the spreading of customers: uniform, 50% central, concentric and compressed. The combination of these patterns gives rise to a basic set of 60 geographic problems.
Three test sets are constructed by adding vehicle-related constraints to the VRPs. The vehicle fleet is homogeneous, i.e. all vehicles have equal capacity, but this capacity is different in the test sets. Two more test sets are constructed by adding customer-related constraints to an unconstrained VRP. One set includes the mixed pick-up and delivery service. Half of the stops require pick-up, the other half are delivery stops. The other set includes heterogeneous demand. Instead of using a fixed demand of 10 units per stops, demand varies within this set between 4 and 16 units per stop. Test sets six and seven are constructed by adding time-related constraints to an unconstrained VRP. The time constraint has to be interpreted as a time window, i.e. a limit in time during which the stops can be visited. Both sets are different in the number of time windows. Set six contains a homogeneous, continuous time window for all stops, while set seven implements two time windows separated by a closing time. The time windows are equal for all stops.

The combination of 60 geographic problems with seven types of constraints leads to a set of 420 problems. For the investigation of the behavior of metaheuristics and the influence of the local improvement operator a reduced set has been produced. By means of the 'Partitioning around medoids clustering method' (Kaufman and Rousseeuw (1989)), a successful clustering of problems of the seven test sets around medoids has been done. A medoid can be considered as the most representative problem of its cluster. Satisfactory results for the cluster analysis were obtained by reducing the complete problem set to 15 problems. Each set is represented by two medoids except the test set with the heterogeneous demand, which is represented by three medoids.

All VRP problems of the reduced set have an equal number of 100 stops, a single depot and an unlimited homogeneous fleet of vehicles. Problems 1 to 6 are constrained only by vehicle capacity. The demand of each stop amounts 10 units. The vehicle capacity equals 100 units for problems 1 and 2, 50 units for problems 3 and 4 and 200 units for problems 5 and 6. Half of the stops of problems 7 and 8 are pick-up points while the other half are delivery points with a demand of 10 units and a homogeneous vehicle fleet of 100 units. Specific to problems 9, 10 and 11 is the heterogeneous demand at the stops. The remaining four problems, 11 to 15, have time-window constraints. The stops of problems 12 and 13 have one homogeneous time-window associated, while for problems 14 and 15 two separated time-windows are assigned to each stop. All problem instances are available on the website `http://www.ruca.ua.ac.be/TEW/PHP/alexvb.htm`.

The 'good' and 'bad' initial solutions for problems 1 to 7 contain the least possible number of routes, i.e. the vehicles are filled to full capacity. For these problems, the move types String Relocation and String Mix cannot be applied and only String Cross and String Exchange can be used. For the remaining problems, the four move types are applicable.

The reader interested in the performance of the heuristics presented in this research on some classical benchmark problems, is referred to Van Breedam (1995) and Janssens and Van Breedam (1995).

4 Parametric analysis

In order to determine the significant effect of the parameters of both metaheuristics, an AID parametric analysis is performed.

The AID technique is aimed at discovering the structure of the relation between variables. The dependent variable, the total travel time, is continuous of nature, while the independent variables, the parameters of the heuristic are nominally scaled.

An AID solution can be represented by a tree structure containing all binary splits. At each phase, an analysis of variance is performed on each two groups of values of the same variable. The split with the highest significant F-value among the analyses of variance is selected. The splitting process is halted if no more binary splits with a significant F-value can be found.

The interested reader is referred to Morgan and Sonquist (1963) and Sonquist et al. (1971) for additional information on the AID-technique.

Some remarks have to be made for the parametric analysis of improvement heuristics. Improvement heuristics require an initial solution to start from. Consequently, the behavior of improvement heuristics depends at least as much on the structure of the neighborhood of the current solution than on the specific problem characteristics. The problem characteristics only have an indirect effect because they affect the neighborhood structure.

The generic parameters of the improvement heuristics also have an influence on the neighborhood structure. Nevertheless, the relation between problem characteristics and the values of the generic parameters is hard to analyse. Generic parameters affect the choice of moves required to go from initial to final solution. The sequence of solutions between initial and final solution is called the trajectory (Pirlot (1992)). The sequential nature of the trajectory makes it difficult to trace the effect of the generic parameters on the final solution.

4.1 Descent heuristic

If the type of move is SE, SR or SM, 16 solutions are obtained by combining all four parameter values of the LI heuristic. Only 4 replications are available if the move is of type SC. The

Heuristics	Parameters	SC	SE	SR	SM	Aggregation
LI	S	+	+	++	++	+
	P	--	--	--	--	--
	F		--	--	--	--
	K		+	-+	-+	
SA	S	+	+	+	++	+
	A	--	--	--	--	--
	L	--	--	--	--	--
	R	-+	-+	-+	-+	-+
	D	+	+	-+	-+	-+
	K		+	-+	-+	
TS	S	++	++	++	++	++
	L	--	--	--	-	-
	I	--	--	--	--	--
	G	++	++	++	++	++
	K		+	-+	+	

Table 1: Summary of the parametric analysis of the improvement heuristics for the four types of move and the aggregated solutions. Symbols: ”--”: significant effect for a very limited number of problems; ”-”: significant effect for a number of problems without common characteristics; ”-+”: significant effect for a number of problems with common characteristics; ”+”: significant effect for most problems with common characteristics; ”++”: consistent significant effect for all problems.

lower number of observations in the case of SC is caused by the fact that parameters K and F are of no relevance for this type of move. The AID-results for the SC move type are less reliable due to the insufficient number of replications.

Initial solution (S). The results of the AID-analyses reveal that the good initial solution gives the best final solutions for almost all problems. For most of these problems, the final solution obtained with the good initial solution is significantly better than the one obtained with the bad initial solution.

As far as the move of type SR and SM is concerned, a significantly better final solution is obtained with a bad initial solution only for problem 14. A possible explanation is that the minimum of the bad initial solution in which the heuristic descends is accidentally deeper than that of the good initial solution.

In general, the results indicate a high degree of dependency of the LI heuristic upon a good initial solution.

String Length (K). For the type SE, a string of length K=2 gives the best solutions for all fifteen problems. For ten out of fifteen problems, K=2 is even significantly better. For the remaining problems 3, 6, 7, 8 and 15 some reasons can be mentioned for the fact that $K=2$ is better but not significantly better than $K=1$. The route sequence imposed by mixed pick-ups and deliveries (problems 7 and 8) or the time windows (problem 15) can hinder the exchange of strings containing more than 1 stop. For problems 3 and 6 where only exchanges of equal string lengths are allowed, exchanges with K=2 are, in essence, equivalent to two exchanges with K=1.

Concerning the move of type SR, the substitutability of a relocation of K=2 stops with two relocations of K=1 stops is the primary reason for the insignificance of the effect of K on the final solution for all problems.

The results for the move of type SM, are a mixture of those of SE and SR.

Selection strategy (P). The difference between the first and the best improvement hardly gives significantly different final solutions.
No dominant strategy is observed even if only the value of P in the best solution is considered.

The observations made by Osman (1993) are comparable.

Evaluation procedure for string length $K > 1$ (F). No significant difference can be observed between the two procedures for the order of evaluation.

Move type. If the results of the four types of moves are compared, it is observed that for problems 8 to 15 the solutions of the SM type are always present in the significantly better move types. Moreover, for seven of these problems the best solution is obtained with the SM type. The SC and SE are mostly significantly worse for these problems.

For the problems 1 to 7, no significant differences are observed between the solutions of SC and SE. If only the move type in the best solution is considered, a dominance of the SE is noticed for five out of seven problems.

The results of the AID analyses for the LI heuristic indicates a clear tendency. The problem-specific parameters, the initial solution and the string length have an explainable behavior.
Conversely, the generic parameters have no significant effect on the final solution. No further conclusions can be drawn for this latter group of parameters.

4.2 Simulated Annealing metaheuristic

The AID-analyses for the SA heuristic are performed on 32 solutions per problem for move type SC and 64 solutions for move types SE, SR and SM. These solutions were obtained by combining the six parameters of the SA implementation. The different number of replications is explained by the fact the string length K is not relevant for move type SC.

Initial solution (S). The quality of the initial solution seems to be somewhat less important for the SA heuristic than for the LI heuristic. The effect of the initial solution on the final solution is for a larger part of the problems significant. However, good initial solutions do not always produce the significantly better final solutions. For about a half of the problems, the significantly better final solutions are obtained with a good initial solution.

This demonstrates the greater independence of the SA heuristic of the quality of the initial solution. A reason for this phenomenon can be the great variability of the SA heuristic and its ability to escape from a local minimum fast.

String Length (K). The results for the string length K have to be moderated somewhat in the case of the SA metaheuristic. Due to the random selection of a move, there is no certainty on the proportion of moves with K=1 and with K=2. In an extreme case, all accepted moves could have the same string length.

The results reveal that for about half of the problems the string length K=2 gives significantly better solutions. Although the dominance of K=2 in the best solution is observed for most problems, any inferences are pure conjecture due to the above-mentioned uncertainty.

Range delimiter (D). A number of tendencies can be deduced from the AID-analyses of the 15 problems for the travel time restriction as range delimiter. A travel time restriction gives significantly better solutions in the cases where the routes of the initial solution are well-separated and/or the only side-constraints are capacity constraints. In these cases, the travel time restriction somewhat prevents the selection of moves which cause a substantial deterioration of the objective function value.

An implementation without travel time restriction gives significantly better solutions for problems with a constraining geographic structure, which mostly contains clustered stops.

For most problems with time-windows, no travel time restriction is required to generate better and often even significantly better solutions. The time-windows play the role of range delimiter for these problems, because they hinder the selection of moves which considerably worsen the objective function value.

Acceptance fraction (A). The acceptance fraction has a significant effect on the final solution for only a very small part of the problems. No dominant value of A is observed, even if only the value of A in the best final solution is considered. The results are approximately comparable for the four types of move.

These results imply that the differences in the initial temperature caused by the different acceptance fractions, 0.30 en 0.50, only give significantly different solutions for a very limited number of problems.

Temperature reduction fraction (R). The significant effect of the cooling rate R occurs only for a small number of problems. Nevertheless, by considering only the value of R in the best final solution, a dominance of R=0.90 is clearly observed. This indicates that a slow cooling mostly yields better solutions.

Our observations confirm the findings of Johnson et al. (1989).

Obviously, the processing time required for a slow cooling (R=0.90) is considerably higher than that for a fast cooling (R=0.70).

Critical acceptance ratio (L). There is no single problem of the four move types for which the critical acceptance ratio gives significantly different solutions. Moreover, for a large number of problems, both values of L give the same best solutions. This means that for these problems, the percentage of accepted moves during the last five temperature reductions of the process was already inferior to the lower of both values of L, i.e. 0.01.

As far as the type of move is concerned, it can be observed that the types SR and SM are significantly better for problems 8 to 15. This can be explained by the ability of both move types to reduce the number of routes. The advantage of the SM over the SR occurs when the number of routes is reduced to the minimal number where no further relocations can be performed. The SM can continue with the exchange of stops in such situations.

For problems 1 to 7, significantly better solutions are obtained with SE for problems 1 to 4 and with SC for problems 5 to 7. The significantly better solutions of the SC for problems 5 to 7 can possibly be related to the long routes of these problems. The exchange of entire route segments as is done with the SC can possibly increase the probability of obtaining a greater improvement in the objective function value.

As a conclusion, we can state that the problem-specific parameter, the initial solution, the string length and the range delimiter, usually have a significant effect on the final solution. Moreover, their effects can more or less be related to the problem characteristics.

As far as the generic parameters are concerned, no consistent significant effect is observed. Only for the parameter representing the cooling rate, it can be observed that a slow cooling is most often better. No clear dominant values can be observed either for the parameters representing the acceptance ratio for determining the initial temperature, or for the critical acceptance ratio.

4.3 Tabu Search metaheuristic

Combining the values of the five parameters of the TS heuristic gives rise to 96 solutions per problem if the move type is SE, SR or SM. For the move type SC, 48 solutions are available for the AID-analyses because the parameter string length, K, is not relevant.

Initial solution (S). For almost all problems of the four move types, the good initial solution gives significantly better final solutions. This high degree of dependency of the TS heuristic on the good quality of the initial solution can be explained by the traditional path followed by the objective function of the TS heuristic. First, the objective function value descends as far as possible. Once it arrived at the point where no further improvements can be performed, the TS heuristic tends to stagnate and starts to oscillate. Consequently, the quality of the initial solution is determining for the quality of the final solution, just like for the LI heuristic.

String Length (K). The analyses reveal that a string length of K=2 stops is significantly better than K=1 stop if a move with K=2 can, in essence, not be substituted by two moves with K=1. This means that K=2 gives significantly better solutions for the SE, particularly for problems where exchanges of unequal string lengths are allowed (problems 8 to 15).

For the SR move type, the string length has a significant effect on half of the problems. For most of these problems K=1 is significantly better. This is explained by the above-mentioned principle of substitution.

The string length K=2 is significantly better for five out of eight problems in the case of move type SM.

On average, the processing time for K=2 is twice as much as the one for K=1.

Length of tabu list (L). The effect of parameter L is significant only for a minority of problems. Even if only the value of L in the best solution is considered, no meaningful deduction can be made with respect to the length of the tabu list.

Iterations for adapting tabu list length (I). The problems for which the different values of parameter I give significantly different solutions are not numerous. Even by considering only the value of I in the best solution, no inferences can be made. For the move of type SC, a light preference in favor of the use of a dynamic tabu list is observed.

The limited effect of the tabu list length and its adaptation confirms the findings of Pureza and França (1991), Semet and Taillard (1993), Osman (1991) and Osman (1993).

Long-term memory (G). For almost all problems, it can be observed that an implementation with a long-term memory gives significantly worse solutions than the implementation without it. This can be explained by the fact that the use of a long-term memory hinders the selection of more favorable stops because they carry a high associated penalty. Consequently, the least favorable stops are involved in the move, which mostly cause a considerable worsening of the objective function value. This diversification strategy induces a greater variety in the path of the objective function value.

Gendreau et al. (1994) suggest that bad results obtained with a long-term function can possibly be caused by the inappropriateness of the constants 0.1 and 0.5 in formula 5.

The computing time for the implementation with a long-term memory is somewhat more time-consuming than the one without because of the calculation of the penalty function for every stop involved in a move.

With respect to the type of move, it can be observed that the SM is mostly included in the significantly best move types for the problems 8 to 15. For the problems 1 to 7, a light dominance in favor of the SE is observed.

The conclusion for the TS is much like these of the two previous heuristics. The significant effect of the problem-specific parameters can mostly be related to the problem characteristics. The most important deduction for this group of parameters is the dependency of the quality of the final solution upon that of the initial solution.

Only for one generic parameter reliable conclusions can be drawn. The use of the long-term memory gives significantly worse solutions for the majority of problems.

5 Heuristic analysis

Due to the fact that the stopping criterion of the SA and TS heuristics is not defined as with the descent heuristic, a simple comparison of only the final solution values of the three heuristics would not be appropriate.

Besides, the computing time of both heuristics highly depends on the value assigned to the parameters. Nevertheless, it remains difficult to estimate the processing time of the SA and TS heuristics. Moreover, the probability of finding a better final solution increases with the run time. A simple comparison of the final solution of the three improvement heuristics without taking into account the run time is not appropriate.

An alternative for comparing the improvement heuristics dynamically is required. The specific feature of the dynamic analysis is that not only the final solutions of the three improvement heuristics are compared, but also their intermediary solutions at various time points. Three time points are considered, corresponding to the time at which the best final solution of each individual heuristic is obtained.

The dynamic heuristic analysis is performed with the LI, SA and TS improvement heuristic for each of the four types of move and for each problem separately.

A statistical analysis of the different solutions is not meaningful, because it is not clear at

Figure 4: Example of the path of the objective function value for the LI, TS and SA heuristics. The dashed vertical lines represent the minima attained by one of the heuristics.

which time point to perform the analysis without (dis)favoring one of the three heuristics.

An important analysis tool for the dynamic heuristic analysis is the graphical representation of the path of the objective function value of each heuristic versus computing time. An example is given in figure 4.

The path of the objective function of the LI heuristic is represented by a straight line connecting the initial with the final solution value.

With respect to the SA, the objective function value of the best solution and its run time are registered at each temperature reduction. Consequently, the SA is represented by a monotonic descending curve. The recording of the best solution found during a temperature reduction instead of the current solution at the time of the temperature reduction can be justified as follows. If the current solution at the time of the temperature reduction were saved, it would not possible to know the best solution if it was reached before the moment of the temperature reduction. The drawback of this approach is that deteriorations of the objective function value are not registered. The only alternative to this approach would have been to record every move performed. However, this would make the graph less clear. Moreover, we are primarily interested in the improvement instead of the worsening of the objective function value of the SA heuristic.

The objective function value of the TS heuristic is recorded at each iteration. In the graph, the path of the objective function value can be identified by its oscillating nature.

The three dashed vertical lines represent the three time points corresponding to the minima obtained by the individual heuristics.

Tables 2 to 5 illustrate the dynamic comparison of the objective function values of the LI, SA and TS heuristics with each move type at the three time points corresponding with the minimum of each heuristic. The symbol '*' indicates which heuristic attains its minimal value after the given run time. The best solution of the three heuristics at each time point is printed in bold face. The column at the right of each cell contains the relative difference with respect to the best solution at that time point. The same computer has been used for all experiments.

The main findings of this analysis can be summarized as follows.

Problem		Initial	Minimum 1		Minimum 2		Minimum 3	
1	*Time*		*5*		*240*		*1006*	
	LI	1245	**1233**(*)		1233	0.02	1233	0.03
	TS	1245	1245	0.01	**1213**		**1192**(*)	
	SA	1245	1245	0.01	**1213**(*)		1213	0.02
2	*Time*		*30*		*142*		*384*	
	LI	1703	**1610**(*)		1610	0.01	1610	0.01
	TS	1703	1637	0.02	**1580**(*)		**1580**	
	SA	2352	2220	0.25	1882	0.20	1613(*)	0.01
3	*Time*		*2*		*113*		*480*	
	LI	1781	**1774**(*)		1774	0.01	1774	0.01
	TS	1781	1781	0.01	**1768**		**1763**(*)	
	SA	1781	1781	0.01	1770(*)	0.01	1770	0.01
4	*Time*		*7*		*17*		*35*	
	LI	1671	**1590**(*)		1590	0.01	1590	0.01
	TS	1671	1602	0.01	**1577**(*)		**1577**	
	SA	3094	3094	0.98	2365	0.50	1639(*)	0.04
5	*Time*		*59*		*846*		*1413*	
	LI	1072	**1019**(*)		1019	0.03	1019	0.05
	TS	1072	1022	0.01	**988**(*)		988	0.01
	SA	1266	1266	0.24	1192	0.21	**975**(*)	
6	*Time*		*83*		*3076*		*4286*	
	LI	1105	**1051**(*)		1051	0.04	1051	0.07
	TS	1105	1083	0.03	**1015**(*)		1015	0.04
	SA	1105	1105	0.05	1023	0.01	**979**(*)	
7	*Time*		*67*		*502*		*2766*	
	LI	1097	**1070**(*)		1070	0.03	1070	0.04
	TS	1097	1097	0.03	**1040**		**1032**(*)	
	SA	1332	1332	0.24	1053(*)	0.01	1053	0.02
8	*Time*		*50*		*390*		*2184*	
	LI	1251	**1198**(*)		1198	0.03	1198	0.04
	TS	1251	1247	0.04	**1154**(*)		1154	
	SA	1251	1251	0.04	1251	0.01	**1106**(*)	0.02
9	*Time*		*5*		*219*		*303*	
	LI	2089	**2009**(*)		2009	0.11	2009	0.11
	TS	2089	2089	0.04	1940	0.07	1841(*)	0.01
	SA	2089	2089	0.04	**1818**(*)		**1818**	
10	*Time*		*7*		*223*		*340*	
	LI	1202	**1141**(*)		1141	0.03	1141	0.04
	TS	1202	1182	0.04	1106	0.01	**1097**(*)	
	SA	1588	1588	0.39	**1102**(*)		1102	0.01
11	*Time*		*37*		*148*		*680*	
	LI	1386	**1303**(*)		1303	0.07	1303	0.07
	TS	1386	1313	0.01	1298	0.07	1261(*)	0.04
	SA	1386	1386	0.06	**1215**(*)		**1215**	
12	*Time*		*42*		*151*		*157*	
	LI	2212	**1830**(*)		1830	0.04	1830	0.04
	TS	2212	1876	0.03	**1754**(*)		**1754**	
	SA	2212	1904	0.04	1789	0.02	1784(*)	0.02
13	*Time*		*6*		*64*		*395*	
	LI	1235	**1127**(*)		1127	0.06	1127	0.06
	TS	1235	1198	0.03	1102	0.04	1064(*)	0.01
	SA	1235	1188	0.05	**1062**(*)		**1062**	
14	*Time*		*11*		*83*		*380*	
	LI	3011	**2857**(*)		2857	0.07	2857	0.08
	TS	3692	3692	0.30	2915	0.09	**2640**(*)	
	SA	3692	3692	0.30	**2663**(*)		2663	0.01
15	*Time*		*35*		*108*		*226*	
	LI	1398	**1236**(*)		1236	0.06	1236	0.06
	TS	1398	1270	0.03	1179	0.01	1169(*)	0.01
	SA	1398	1331	0.07	**1165**(*)		**1165**	

Table 2: Heuristic analysis of LI, TS and SA with move type SC.

Problem		Initial	Minimum 1		Minimum 2		Minimum 3	
1	*Time*		*56*		*306*		*4120*	
	LI	1245	**1158**(*)		1158	0.01	1158	0.01
	TS	1245	1199	0.03	1155	0.01	**1143**(*)	
	SA	1245	1245	0.07	**1144**(*)		1144	0.01
2	*Time*		*68*		*421*		*3503*	
	LI	1703	**1609**(*)		1609	0.01	1609	0.01
	TS	2352	2193	0.36	1903	0.20	1591(*)	0.01
	SA	1703	1703	0.06	**1585**(*)		**1585**	
3	*Time*		*138*		*177*		*638*	
	LI	2726	1756	0.01	1756(*)	0.01	1756	0.01
	TS	1781	1762	0.01	1762	0.01	**1751**(*)	
	SA	1781	**1752**(*)		**1752**		1752	0.01
4	*Time*		*169*		*193*		*265*	
	LI	3049	1487	0.01	1487(*)	0.01	1487	0.01
	TS	1671	1495	0.01	1495	0.01	**1476**(*)	0.01
	SA	1671	**1486**(*)		**1486**		1486	
5	*Time*		*249*		*4417*		*5346*	
	LI	1072	**1030**(*)		1030	0.03	1030	0.03
	TS	1072	1037	0.01	**998**(*)		**998**	
	SA	1072	1072	0.04	1072	0.07	1014(*)	0.02
6	*Time*		*172*		*2501*		*4595*	
	LI	1105	**1037**(*)		1037	0.05	1037	0.05
	TS	1105	1067	0.03	**984**(*)		**984**	
	SA	1105	1105	0.06	1105	0.11	1037(*)	0.05
7	*Time*		*69*		*888*		*1164*	
	LI	1097	**1096**(*)		1096	0.04	1096	0.06
	TS	1097	1097	0.01	1067	0.01	**1038**(*)	
	SA	1097	1097	0.01	**1055**(*)		1055	0.02
8	*Time*		*680*		*2081*		*2667*	
	LI	1251	**1200**(*)		1200	0.05	1200	0.08
	TS	1251	1209	0.01	1157	0.02	**1114**(*)	
	SA	1251	1251	0.04	**1137**(*)		1137	0.02
9	*Time*		*72*		*207*		*387*	
	LI	2089	**1852**(*)		1852	0.03	1852	0.03
	TS	2543	1907	0.03	**1800**(*)		1800	0.01
	SA	2089	2544	0.37	2207	0.11	**1799**(*)	
10	*Time*		*33*		*323*		*2229*	
	LI	1202	**1098**(*)		1098	0.01	1098	0.04
	TS	1202	1141	0.04	1114	0.02	**1052**(*)	
	SA	1588	1564	0.42	**1094**(*)		1094	0.04
11	*Time*		*131*		*359*		*372*	
	LI	1386	**1250**(*)		1250	0.03	1250	0.06
	TS	1386	1269	0.01	1238	0.02	**1174**(*)	
	SA	1948	1948	0.56	**1217**(*)		1217	0.04
12	*Time*		*105*		*259*		*5021*	
	LI	2212	1955(*)	0.04	1955	0.05	1955	0.05
	TS	2212	2035	0.06	1988	0.06	1877(*)	0.01
	SA	2212	**1926**		**1866**(*)		**1866**	
13	*Time*		*48*		*215*		*2765*	
	LI	1235	**1108**(*)		1108	0.05	1108	0.09
	TS	1235	1168	0.05	1116	0.07	**1019**(*)	
	SA	1235	1149	0.04	**1046**(*)		1046	0.03
14	*Time*		*36*		*280*		*3022*	
	LI	3011	2868(*)	0.03	2868	0.12	2868	0.17
	TS	3011	2890	0.04	2769	0.08	**2443**(*)	
	SA	3011	**2781**		**2558**(*)		2558	0.04
15	*Time*		*116*		*2078*		*1843*	
	LI	1398	1219(*)	0.05	1219	0.07	1219	0.07
	TS	1398	1259	0.08	1215	0.07	1149(*)	0.01
	SA	1398	**1161**		**1135**(*)		**1135**	

Table 3: Heuristic analysis of LI, TS and SA with move type SE.

Problem		Initial	Minimum 1		Minimum 2		Minimum 3	
8	*Time*		*99*		*823*		*2279*	
	LI	1251	**1128**(*)		1128	0.06	1128	0.06
	TS	1251	1132	0.01	**1067**(*)		**1067**	
	SA	1251	1251	0.11	1251	0.17	1098(*)	0.03
9	*Time*		*9*		*10*		*142*	
	LI	2089	**1969**(*)		**1969**		1969	0.07
	TS	2089	2017	0.02	2017(*)	0.02	2017	0.10
	SA	2089	2089	0.06	2089	0.06	**1833**(*)	
10	*Time*		*50*		*261*		*268*	
	LI	1202	**1108**(*)		1108	0.01	1108	0.03
	TS	1202	1120	0.02	**1097**		**1075**(*)	
	SA	1202	1202	0.08	1108(*)	0.01	1108	0.03
11	*Time*		*67*		*167*		*372*	
	LI	1386	**1252**(*)		1252	0.03	1252	0.03
	TS	1386	1267	0.01	1263	0.04	1247(*)	0.02
	SA	1386	1343	0.07	**1220**(*)		**1220**	
12	*Time*		*47*		*234*		*381*	
	LI	2212	**1844**(*)		1844	0.14	1844	0.14
	TS	2212	1968	0.07	1699	0.05	1678(*)	0.04
	SA	2212	2078	0.013	**1616**(*)		**1616**	
13	*Time*		*23*		*141*		*349*	
	LI	1235	1134(*)	0.00	1134	0.15	1134	0.15
	TS	1235	1156	0.02	1045	0.06	990(*)	0.01
	SA	1235	**1133**		**983**(*)		**983**	
14	*Time*		*244*		*804*		*1312*	
	LI	3692	2681(*)	0.11	2681	0.14	2681	0.14
	TS	3011	2691	0.12	2437	0.04	**2345**(*)	
	SA	3692	**2382**		**2350**(*)		2350	0.01
15	*Time*		*49*		*131*		*712*	
	LI	1398	**1248**(*)		1248	0.11	1248	0.13
	TS	1398	1271	0.02	1170	0.04	**1106**(*)	
	SA	2162	1301	0.03	**1120**(*)		1120	0.01

Table 4: Heuristic analysis of LI, TS and SA with move type SR.

Problem		Initial	Minimum 1		Minimum 2		Minimum 3	
8	*Time*		*150*		*735*		*2624*	
	LI	1251	**1125**(*)		1125	0.05	1125	0.05
	TS	1251	1137	0.01	**1069**(*)		1069	0.01
	SA	1251	1251	0.11	1251	0.17	**1068**(*)	
9	*Time*		*25*		*119*		*1181*	
	LI	2089	**1876**(*)		1876	0.03	1876	0.05
	TS	2089	1966	0.05	1856	0.02	**1790**(*)	
	SA	2089	2086	0.11	**1818**(*)		1818	0.01
10	*Time*		*85*		*364*		*669*	
	LI	1202	**1064**(*)		1064	0.02	1064	0.02
	TS	1202	1106	0.04	**1049**		**1045**(*)	
	SA	1588	1588	0.49	1063(*)	0.02	1063	0.02
11	*Time*		*119*		*349*		*2261*	
	LI	1386	**1205**(*)		1205	0.01	1205	0.04
	TS	1386	1261	0.05	1225	0.02	**1160**(*)	
	SA	1386	1386	0.15	**1187**(*)		1187	0.02
12	*Time*		*251*		*350*		*2654*	
	LI	2212	1770(*)	0.04	1770	0.04	1770	0.04
	TS	2212	1809	0.06	1770	0.04	1713(*)	0.01
	SA	2212	1700		**1696**(*)		**1696**	
13	*Time*		*82*		*165*		*349*	
	LI	1235	1058(*)	0.01	1058	0.06	1058	0.07
	TS	1235	1088	0.04	1050	0.06	**989**(*)	
	SA	1235	**1044**		**998**(*)		998	0.01
14	*Time*		*171*		*183*		*1915*	
	LI	3692	2645	0.09	2645(*)	0.09	2645	0.13
	TS	3011	2784	0.14	2725	0.12	**2337**(*)	
	SA	3011	**2435**(*)		**2435**		2435	0.04
15	*Time*		*119*		*131*		*1903*	
	LI	1398	1182	0.06	1182(*)	0.06	1182	0.09
	TS	1398	1242	0.11	1202	0.08	**1083**(*)	
	SA	1398	**1118**(*)		**1118**		1118	0.03

Table 5: Heuristic analysis of LI, TS and SA with move type SM.

The final solution of the LI heuristic is usually the first minimum encountered. This LI solution is mostly better than the intermediary SA and TS solution at that time point. Only for some problems with time-windows, the solution of the SA is already better at the first minimum.

For the time points corresponding to minima 2 and 3 the best solutions are obtained with the TS and SA heuristics. The results of the analysis do not show a preference of one over the other.

The SA heuristic tends to produce a final solution in less run time than the TS heuristic. The specific path of the objective function value of the TS value indicates that the heuristic can be speeded up by stopping after a fixed number of iterations without improvement and by using a range delimiter for restricting the number of moves to be evaluated at each iteration. Moreover, the difference of the final solutions between the SA and TS rarely exceeds 4%.

A final remark is the comparison of the final results of the initial with those of the improvement heuristics for the fifteen problems of the reduced test set. A remarkable observation is that the best final solution obtained with one of the three improvement heuristics is better than the best solution obtained with one of the initial heuristics only for eight out of the fifteen problems. This indicates that the quality of the initial heuristics remains very important. Moreover, the initial heuristic generates an initial solution in a very short time as opposed to that of the improvement heuristic. In addition, the TS and LI heuristics are highly dependent on the quality of an initial solution. All these arguments allow us to conclude that it is worth spending much effort on conceiving a good initial heuristic, while subsequently this good initial solution can further be improved by an improvement heuristic if sufficient run time and computer resources are affordable.

6 Conclusions

The comparison of descent heuristics and metaheuristics for the vehicle routing problem revealed some valuable results. The comparison has been conducted at a parametric as well as on a heuristic level on a test set of fifteen problem instances.

The *parametric analysis* resulted in a classification of the parameters into two main groups: the problem-specific and the generic parameters. The significant effect and the significantly better values of the problem-specific parameters could more or less be related to the problem characteristics. Therefore, the transferability of the findings for these parameters to other problems with comparable characteristics cannot be excluded.

For a large number of problems, most of the generic parameters have no significant effect on the final objective function value. The portability of these findings to other problems is speculative, due to the fact that the values for the generic parameters can hardly be related to the problem characteristics.

Useful recommendations resulting from the analysis of the generic parameters are the slow cooling in the case of SA and the bad solutions obtained with the long-term memory in the case of the TS heuristic.

The *heuristic analysis* of the improvement heuristics is a dynamic analysis. The main conclusion of the dynamic analysis is that the available run time determines the choice of the

best improvement heuristic.
The heuristic analysis did not permit to indicate the better among both metaheuristics, SA or TS. The difference between their final solution never exceeded 4%. The concept of the dynamic analysis proposed is transferable to other problems. The results obtained are only transferable to problems with almost identical characteristics.

The dynamic heuristic analysis can be a useful tool for improving the implementation of the heuristics by analysing the path of the objective function value produced by the heuristics. This has has been illustrated by the recommendations we proposed for speeding up the TS heuristic. Moreover, the opportunities for building hybrid metaheuristics, combining the best features of both metaheuristics can be evaluated. The TS provides the certainty of finding at least a local minimum, while the SA permits a fast shift of the search process towards other minima.

References

Aarts, E. and J. Korst (1989). *Simulated annealing and Boltzmann machines.* Wiley, Chichester.

Alfa, A., S. Heragu, and M. Chen (1991). A 3-opt based simulated annealing algorithm for vehicle routing problems. *Computers and Industrial Engineering 21*, 635–639.

Cerny, V. (1985). A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory Appl. 45*, 41–51.

Collins, N., R. Eglese, and B. Golden (1988). Simulated annealing - an annotated bibliography. *American Journal of Mathematical and Management Sciences 8*, 209–307.

Croes, A. (1958). A method for solving traveling salesman problems. *Operations Research 5*, 791–812.

Dror, M. and L. Levy (1986). A vehicle routing improvement algorithm comparison of a greedy and a matching implementation for inventory routing. *Computers and Operations Research 13*(1), 33–45.

Eglese, R. (1990). Simulated annealing: a tool for operational research. *European Journal of Operational Research 46*, 271–281.

Gendreau, M., A. Hertz, and G. Laporte (1994). A tabu search heuristic for the vehicle routing problem. *Management Science 40*, 1276–1290.

Gendreau, M., G. Laporte, and J.-Y. Potvin (1998). Metaheuristics for the vehicle routing problem. Technical Report G-98-52, Les Cahiers du GERAD, Montréal, Quebec, Canada.

Glover, F. (1986). The general employee scheduling problem: An integration of management science and artificial intelligence. *Computers and Operations Research 15*, 563–593.

Glover, F. (1989). Tabu search: Part i. *ORSA Journal on Computing 1*, 190–206.

Glover, F. (1990). Tabu search: Part ii. *ORSA Journal on Computing 2*, 4–32.

Glover, F. and M. Laguna (1993). Tabu search. In C. Reeves (Ed.), *Modern Heuristic Techniques for Combinatorial Problems.* Blackwell Scientific Publications, Oxford.

Golden, B. and C. Skiscim (1986). Using simulated annealing to solve routing and location problems. *Naval Research Logistics Quarterly 33*, 261–279.

Hansen, P. (1986). The steepest ascent mildest descent heuristic for combinatorial programming. In *Presented at the Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy.* presented at the Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy.

Janssens, G. and A. Van Breedam (1995). A simulated annealing post-processor for the vehicle routing problem. In V. Rayward-Smith (Ed.), *Applications of Modern Heuristics*, Chapter Case studies, pp. 175–191. Alfred Waller Ltd.

Johnson, D., C. Aragon, L. McGeoch, and C. Schevon (1989). Optimization by simulated annealing: an experimental evaluation; part i: Graph partitioning. *Operations Research 37*, 865–892.

Kaufman, L. and P. Rousseeuw (1989). *Finding groups in data. An introduction to Cluster Analysis.* Wiley, New York.

Kirckpatrick, S., C. Gelatt Jr., and M. Vecchi (1983). Optimization by simulated annealing. *Science 220*, 671–680.

Laporte, G. and I. Osman (1995). Routing problems: A bibliography. *Annals of Operations Research 61*, 227–262.

Lenstra, J. and A. Rinnooy Kan (1981). Complexity of vehicle routing and scheduling problems. *Networks 11*, 221–227.

Lin, S. (1965). Computer solutions to the traveling salesman problem. *Bell System Technical Journal 44*, 2245–2269.

Lin, S. and B. Kernighan (1973). An effective heuristic algorithm for the traveling salesman problem. *Operations Research 21*, 498–516.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics 21*, 1087–1092.

Morgan, J. and J. Sonquist (1963). Problems in the analysis of survey data and a proposal. *Journal of the American Statistical Association 58*, 415–434.

Or, I. (1976). *Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking.* Ph. D. thesis, Northwestern University, Evanston, IL.

Osman, I. (1991). *Metastrategy simulated annealing and tabu search algorithms for combinatorial optimization problems.* Ph. D. thesis, The Management School, University of London.

Osman, I. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research 41*, 421–451.

Osman, I. and G. Laporte (1996). Metaheuristics: A bibliography. *Annals of Operations Research 63*, 513–623.

Pirlot, M. (1992). General local search heuristics in combinatorial optimization: a tutorial. *Jorbel 32*(2), 7–67.

Potvin, J.-Y., T. Kervahut, B. Garcia, and J.-M. Rousseau (1992). A tabu search heuristic for the vehicle routing problem with time windows. Technical Report CRT-855, Centre de Recherche sur les Transports, Université de Montréal, Montréal, Canada.

Pureza, V. and P. França (1991). Vehicle routing problems via tabu search metaheuristic. Technical Report CRT-747, Centre de Recherche sur les Transports, Université de Montréal, Montréal, Canada.

Robusté, F., C. Daganzo, and R. Souleyrette (1990). Implementing vehicle routing models. *Transportation Research B 24*, 263–286.

Savelsbergh, M. (1988). *Computer Aided Routing*. Ph. D. thesis, Centrum voor Wiskunde en Informatica, Amsterdam.

Semet, F. and E. Taillard (1993). Solving real-life vehicle routing problems efficiently using taboo search. *Annals of Operations Research 41*, 469–488.

Sonquist, J., E. Baker, and J. Morgan (1971). *Searching for structure*. Survey Research Center, University of Michigan, Ann Arbor.

Stewart, W., J. Kelly, and M. Laguna (1992). Solving vehicle routing problems using generalized assignments and tabu search. Technical report, College of Business Administration, University of Colorado, Boulder, CO.

Taillard, E. (1993). Parallel iterative search methods for vehicle routing problems. *Networks 23*, 661–673.

Teodorovic, D. and G. Pavkovic (1992). A simulated annealing technique approach to the vehicle routing problem in the case of stochastic demand. *Transportation Planning and Technology 16*, 261–273.

Thangiah, S., I. Osman, and T. Sun (1994). Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problem with time windows. Technical Report SRU-CpSc-TR-94-27, Computer Science Department, Slippery Rock University, Slippery Rock, PA.

Thangiah, S., I. Osman, R. Vinayagamoorthy, and T. Sun (1994). Algorithms for vehicle routing problems with time deadlines. *American Journal of Mathematical and Management Sciences 13*(3-4), 325–355.

Van Breedam, A. (1995). Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research 86*(3), 480–490.

Van Laarhoven, P. and E. Aarts (1987). *Simulated Annealing: Theory and Practice*. Kluwer Academic Publishers, Dordrecht, The Netherlands.