CRT–95–13, ORWP 95/03, October 1995. To appear in Journal of heuristics 1, 1995, pp. 147 – 167

PROBABILISTIC DIVERSIFICATION AND INTENSIFICATION IN LOCAL SEARCH FOR VEHICLE ROUTING

Yves Rochat

Département de mathématiques, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Suisse
e-mail : rochat@dma.epfl.ch WWW : http://dmawww.epfl.ch/rose.mosaic/rochat.html

Éric D. Taillard

Centre de recherche sur les transports, Université de Montréal, C. P. 6128, succursale Centre-Ville, Montréal, Canada H3C 3J7
e-mail : taillard@crt.umontreal.ca WWW : http://www.iro.umontreal.ca/people/taillard

Abstract :

This paper presents a probabilistic technique to diversify, intensify and parallelize a local search adapted for solving vehicle routing problems. This technique may be applied to a very wide variety of vehicle routing problems and local searches. It is shown that efficient first level taboo searches for vehicle routing problems may be significantly improved with this technique. Moreover, the solutions produced by this technique may often be improved by a post-optimization technique presented in this paper too. The solutions of nearly 40 problem instances of the literature have been improved.

Key words : Vehicle routing, local searches, parallel algorithms.

1. INTRODUCTION

More and more, local search methods are used to find good solutions to combinatorial optimization problems. Throughout the paper, we use the term local search as a synonym of neighbourhood search. Local searches are sometimes restricted to steepest descent algorithms but we also include taboo search and simulated annealing in local search methods. These techniques have two main weaknesses : first they may sometimes be trapped in a very poor local optimum and this difficulty may be hard to overcome even by a fastidious tuning of the local search parameters ; second, these methods require a large computational effort. Nowadays, the most promising way of cutting down computational time is to use parallel computers. Unfortunately, local search is a process that is intrinsically sequential and therefore not always easy to parallelize.

In this paper, we describe a technique that overcomes both weaknesses. We illustrate this technique on two taboo searches we have developed for vehicle routing problems (VRPs) : first for the most elementary VRP (one depot, identical vehicles) and second for the VRP with time windows (VRPTW). This technique may also be applied to other local searches or other VRPs. In section 2, we briefly describe the problems treated and the local search used to solve them. In section 3, we present the new technique that allows us first to diversify the search by exploring solutions that are very different from each other, second to intensify the search in order to identify better local optima in a promising region of the set of feasible solutions, and third to parallelize the process. Then, we present a post-optimization technique that allows frequent improvement in the solutions produced by the diversification and intensification procedure. In section 4, we compare the new technique to the original local searches on which they are based and finally we conclude in section 5.

2. VRP AND LOCAL SEARCH

2.1. Presentation of the problems

The first problem we treat is the following : identical vehicles with a fixed carrying capacity Q must deliver order quantities q_i ($i = 1, \ldots, n$) of goods to n customers from a single depot ($i = 0$). Knowing the distance d_{ij} between customers i and j ($i, j = 0, \ldots, n$), the problem is to find tours for the vehicles in such a way that :

- The total distance travelled by the vehicles is minimized.
- Only one vehicle handles the deliveries for a given customer.
- The total quantity of goods that a single vehicle delivers cannot be larger than Q.

The second problem we treat is similar to the elementary VRP, but distances are interpreted as durations and each customer i requires a service time s_i and must be served during a time window $[b_i, e_i]$. When a vehicle arrives at customer i before time b_i, it must wait. Solomon (1987) has proposed a set of 56 problems where the first objective is to minimize the number of vehicles needed to deliver all orders and the second objective, subject to optimizing the first, is to minimize the total distance travelled by the vehicles.

2.2. Local search for VRP

Very generally, a local search can be formulated as follows :

(a) Choose an initial solution s^0 ; set $k := 0$;

(b) While a stopping criterion is not satisfied, repeat :

(b1) Choose a solution $s^{k+1} \in N(s^k)$, the set of neighbour solutions of s^k.

(b2) Set $k := k + 1$

The choice of a policy for points (a) to (b2) leads to various iterative searches. Generally, the initial solution is chosen in such a way that its generation is fast and easy. The choice of the stopping criterion is often related to the type of iterative search chosen : for example, a descent method stops as soon as there is no better solution than s^k in $N(s^k)$; a taboo search stops when the number k of iterations is greater than a threshold K. In fact, common terminology equates local search with a descent method. We use a more general terminology based on allowing the rules for choosing among its elements to follow a broader design than customary. From the perspective of the formulation adopted here, the way of defining $N(s)$ and the way of selecting a solution in $N(s)$ are the most difficult parts of the design of a local search. For the VRP, the reader may refer to the works of Taillard (1993), Gendreau et al. (1994) or Gendreau, Laporte and Potvin (1995) for more details on these choices.

The method of Taillard (1993) is one of the most efficient for the elementary VRP. Its main feature is to partition large problems into independent subproblems and to optimize each subproblem independently. A partition generates sets (subproblems) involving four to eight tours (or 30 to 60 customers) that are near one another. Once the subproblems are optimized, all the tours of the subproblems are grouped together to construct a solution to the original problem and this solution is again partitioned, and so on. There is a random component in the partition process. Therefore, the algorithm may produce very different solutions from one run to the next. The taboo search we have used in this paper has been slightly changed from the original version of Taillard (1993) : the partition procedure has been improved and the exact procedure used for the optimization of the tours has been replaced by a heuristic approach that is more reliable. Generally, these modifications improve the behaviour of the method on non-uniform problem instances.

For uniform problems (i. e., with customers uniformly distributed in the plane around the depot and ordering quantities much smaller than the vehicle capacity), this algorithm finds good solutions in a small amount of computation time. If the problems are non-uniform, the search may be trapped in a poor solution with a high probability. Since most real-life problems are non-uniform (see e. g. the 120-city problem of Christofides et al. (1979), Taillard (1993), Semet and Taillard (1993), Fisher (1994), Rochat and Semet (1994)), it is worth designing a method that works well on this type of problems.

The taboo search we used for the VRPTW is derived from the adaptation of Rochat and Semet (1994) for a real-life problem that is more complex than the VRPTW ; for example this real-life problem incorporates an heterogeneous fleet, driver's breaks and accessibility constraints (where each customer can only be reached by a subset of the vehicles). So our taboo search for the VRPTW is more nearly a simplification of this complex taboo search than a new taboo search specially designed for the Solomon's VRPTW instances. However, the resulting method is competitive, since it succeeds in improving the quality of 16 previous best known solutions out of the 56 Solomon instances (see table 6). There is a random component in our taboo search for the VRPTW. This means that two runs of the method will generally produce two different solutions.

3. IMPROVING LOCAL SEARCHES APPLIED TO THE VRP

3.1. Probabilistic diversification and intensification technique

A fundamental principle of taboo search is to exploit the interplay between diversification and intensification where diversification drives the search to examine new regions, and intensification focuses more intently on regions previously found to be good. (Intensification typically operates by re-starting from high quality solutions, or by modifying choice rules to favour the inclusion of attributes of these solutions).

Our approach to achieving such an exploitation is based on two primary perspectives, which we describe as follows. The first of these comes from probabilistic taboo search, which is founded on the idea of translating information generated by the search history, coupled with current measures of attractiveness, into evaluations that are monotonically mapped into probabilities of selection. Operating in a neighbourhood framework, the approach then successively selects among available alternatives according to a probability assignment that is strongly biased to favour the choice of higher evaluations. (In contrast to some terminology, taboo search refers to neighbourhoods that are constructive and destructive as well as transitional, since it includes strategies not only of restarting, but also of alternating between constructive and destructive steps).

This approach is motivated by the following premise : an intelligent use of randomization, which is not blindly uniform but embedded in probabilities that account for history and measures of attractiveness, offers a useful type of diversification that can substitute for more complex uses of memory. As noted in Glover (1989), "... the use of randomization, via assigned probabilities, allows a gain in efficiency by obviating extensive record keeping and evaluation operations that a more systematic pursuit of diversity may require". We take advantage of this means of achieving diversity in a special way, by rules, we indicate subsequently.

The second main perspective that underlies our approach derives from one of the most basic (and earliest) types of intensification strategies. The heart of this approach lies in generating solutions by reference to the notions of strongly determined and consistent variables. A strongly determined variable is one whose value cannot be changed except by inducing a disruptive effect on the objective function value or on the values of other variables. The identification of strongly determined variables is by reference to best solutions from previous solution efforts, which motivates us to measure their strength by the quality of the solutions in which variables lie (at particular values). This weighting by objective function values corresponds to the practice of emphasizing relative attractiveness in probabilistic taboo search, allowing us to exploit both approaches together.

A consistent variable is one that is frequently strongly determined at a particular value (or in a narrow range). Specifically, the more often a variable receives a particular value in the best solutions (where we weight these solutions by their objective value), the more highly it qualifies as consistent.

The rationale for isolating such variables, and the strategy for exploiting them, are embodied in the following expectations. First, a variable that is highly consistent is likely to receive its preferred value (or lie in its preferred range) in optimal and near-optimal solutions. Second, once some variables are assigned specific values, other variables that previously did not seem highly consistent will now become a good deal more so. Third, imposing narrow restrictions on selected variables will yield increasingly reliable measures of the relative consistency of remaining variables, given the imposed restrictions.

The strategy to take advantage of these tendencies may then be summarized as follows (Glover, 1977).

(a) Select one or more variables with greatest relative consistencies and constrain these to their preferred values.

(b) Determine new relative consistencies for the variables on the basis of the restriction of step (a).

(c) Repeat the process until all variables have been constrained to specific values.

This process is then joined with a heuristic improvement procedure to transform its value assignments into a new solution, thus creating an iterative method for obtaining progressively more refined outcomes.

In our approach for executing the preceding steps in the vehicle routing setting we select variables in blocks, composed of specific tours to which customers have been assigned in the best previous solutions. The new relative consistency measures produced for step (b) then result by observing that once particular variables are assigned values, we cannot reassign them within the same solution — that is, we cannot select a new tour containing customers already assigned on a previous step. Our improvement heuristics for obtaining better solutions, and which are the drivers of the method, are those of Taillard (1993) and Rochat and Semet (1994).

We now sketch the details of our approach more precisely. We emphasize that our method can be applied to most forms of local search (as encompassed by our rather general definition), and to most types of VRPs. To play a diversification rôle, a local search must be able to produce solutions that are very different from each other, but not necessarily among the very best. This condition is generally satisfied by iterated local searches that either start with solutions randomly generated or that apply a random component in their subsequent decisions, thereby causing different runs generally to produce different solutions. However, by incorporating the principle of probabilistic taboo search to guide our choices, we create a more strategic type of diversification than that embodied in simple randomization. The technique we employ may be described as follows:

In a first phase (initialization), the search is diversified by generating, with the local search, I solutions that are different from one another. For our approach to the elementary VRP, this generation is done by considering various initial decompositions of the problem. For the VRPTW — as well as for elementary VRPs of small size that cannot be decomposed — the non-deterministic characteristics of our local search guarantee the diversity of the solutions.

By generating several initial solutions with the local search, one hopes that all the information necessary to create solutions of very high quality exists in these solutions, but in a non-apparent way. In the case of the VRP, this information is included in the tours. So, one hopes that the initialization creates a set of tours that includes members not very different from the tours of a good solution. It is not unrealistic to think that it is easy to create good tours. The challenge is to find a set of tours whose members are simultaneously good for all customers.

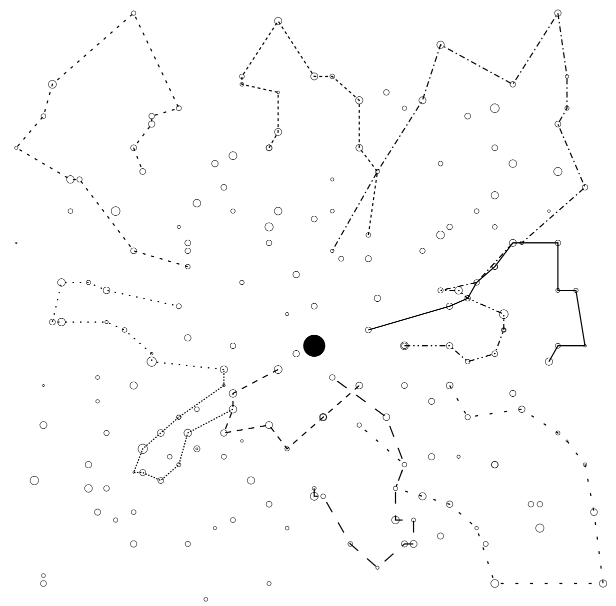

Figure 1 : Example of tours belonging to *T* after the initialization phase.

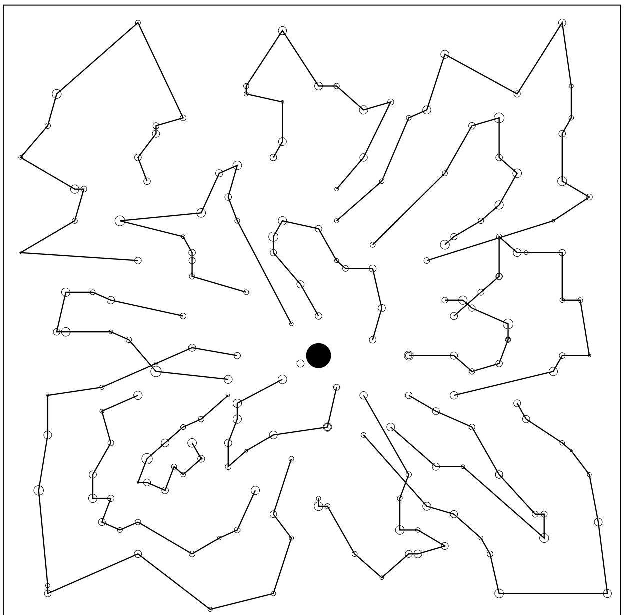

Figure 2 : Best known solution for the problem of Christofides et al. (1979) with 199 customers.

The generation of *I* initial solutions creates a set *T* of tours. In figure 1, we give a subset of tours of *T* after the initialization phase for a problem of Christofides et al. (1979) with 199 customers. This figure may be compared with figure 2 where a very good solution to the same problem is given. We see that several tours are similar, two of them being identical. However, *T* contains several other tours and, after the initialization phase, the good tours cannot be directly identified. In these figures, the area of the customers (empty circles) is proportional to the quantity ordered ; the filled circle is the depot and its area is proportional to the capacity of one vehicle ; the first and last trips of each vehicle are not drawn.

In a second phase, the goal is to extract these good tours and improve them. We start from the principle that, if a solution ***s*** includes tours that belong to a good solution, then the objective function value of ***s*** is probably better than that of a solution that does not contain such tours. The second phase thus favours the extraction of tours that belong to the best solutions generated during the initialization phase. This extraction must not totally exclude tours belonging to bad solutions.

To implement this phase, each tour is labelled with the value of the solution to which it belongs. The set *T* is sorted by increasing values of the labels and the tours with only one customer are removed from *T*, since they do not contain interesting information. Then, we choose tours of *T* probabilistically, by giving preference to tours with low labels and by ignoring tours that include customers belonging to tours already extracted. This choice is repeated until it is not possible to extract new tours from *T*.

Let *S* be the set of tours thus extracted. Since *S* may not contain all the customers of the problem, *S* is a partial solution. In order to construct a feasible solution from *S*, the set of customers not belonging to the tours of *S* may be considered as an independent VRP (of small size) that can be solved by the local search. The tours of this independent VRP are added to *S* to create a feasible solution to the initial VRP. This feasible solution is considered as the initial solution of a local search that will try to improve its quality. Another way to initialize the local search is to modify the procedure, embedded in the local search, that produces an initial solution from a set of customers. This procedure is modified in such a way that it also accepts the tours of the partial solution *S*. In our implementations, we have chosen this way to initialize the local searches.

Once a new solution is generated by the current application of local search, the tours of this solution are labelled with the value of the objective function and are included in the set *T*. It is important to note that the same tour may occur in more than one solution. In order to represent this fact appropriately (in the sense of identifying the relative consistency of consistent variables), a tour is included in *T* multiple times, one for each solution in the set *T*. The extraction of tours of *T*, followed by the optimization with the local search and the insertion of the new tours in *T* is repeated until a stopping criterion is met. More formally, this algorithm may be formulated as follows :

Diversification and intensification algorithm :

Initialization :

(a) Generate I different initial solutions with the local search.

(b) Label each tour with the value of the solution to which it belongs.

(c) Remove the tours having only one customer.

(d) Insert the remaining tours in a set T of tours.

(e) Sort T by increasing values of labels.

Diversification and intensification (to repeat until a stopping criterion is satisfied) :

(1) Set $T' := T$, $S := \varnothing$.

(2) While $T' \neq \varnothing$, repeat

- (2a) Choose $t \in T'$, probabilistically, based on its current relative evaluation.
- (2b) Set $S := S \cup \{t\}$.
- (2c) Remove from T' all the tours including one or several customers belonging to t.

(3) If some customers are not covered by the tours of S, construct a feasible solution S', including them, using the partial solution S.

(4) Improve with the local search the solution S'.

(5) Label the tours of the improved solutions, remove tours with only one customer, insert the remaining tours in T, sort T as in steps (b) to (e) of the initialization.

To favour the generation of good solutions, the tours are not uniformly chosen at step (2a), but the ith worst tour of T' has a probability of $2i/(|T'|\,|T' + 1|)$ of being chosen. For practical reasons, it is necessary to limit the size T to a value L. After each sorting of T, the last $|T| - L$ tours are removed from T if $|T| > L$.

We note this procedure also contains an implicit aspect of “combining” solutions since we construct new ones out of components of previous ones. Hence in this sense our approach embodies some of the spirit of genetic algorithms (see, e. g., Holland (1976) and Davis (1987)). This “implicit combination by intensification” (by exploiting strongly determined and consistent variables), whose origins are approximately contemporaneously with those of genetic algorithms provides a useful counterpart to combination by “genetic” operators.

The working principles of this algorithm may be explained as follows : If I and L are large enough, the initialization phase guarantees that various regions of the solution space will be explored. The creation of partial solutions at the beginning of the second phase (step (2)) allows the search to extend the diversity of the solutions visited.

As far as the process goes, T grows and the partial solutions are more and more complete. At the end, they are often feasible or even, in few cases, better than the best solution found so far. After having performed the second phase several times, the process automatically intensifies the search in promising regions, since the tours are not uniformly chosen at step (2a) and the worst solutions of T are removed. Moreover, identical tours are not removed from T and, more and more often, there exist tours that are not modified by the local search. So, the best tours of T are more and more frequently extracted during the construction of the partial solution and the search progressively changes from a diversification to an intensification process.

This process may easily be parallelized. The steps (a) and (4) (applying the local search) are those consuming the most computation time. A master-slave approach is convenient : The master process executes steps (b) to (e) of the initialization phase and steps (1), (2), (3) and (5) of the diversification and intensification phase. In addition I slave processes independently perform one local search of step (a), then transmit one initial solution to the master process and enter a loop in which they wait for a partial solution from the master process, improve this solution with the local search (step (4)) and finally transmit the improved solution to the master process.

3.2. A post-optimization technique

When examining the empirical behaviour of this procedure, it turns out that during the diversification and intensification phase, the process sometimes succeeds in improving the best solution directly from the solution constructed at step (2) before applying the local search of step (4). This means that, with the tours contained in the set T, it may be possible to build solutions better than those already produced.

Let c_j be the length of the j^{th} tour of T ($j = 1,\ldots, |T|$) and, for $i = 1, \ldots, n$ and $j = 1,\ldots, |T|$:

$$a_{ij} = \begin{cases} 1 \text{ if customer } i \in j^{\text{th}} \text{ tour of } T \\ 0 \text{ Otherwise} \end{cases}$$

Then, the best solution that can be built using tours of T may be found by solving the following set partitioning problem :

$$\begin{aligned} \min \quad & \sum_{j=1}^{|T|} c_j x_j \\ \text{s. t.} \quad & \sum_{j=1}^{|T|} a_{ij} x_j = 1 \qquad i = 1, \ldots, n \\ & x_j \in \{0, 1\} \qquad j = 1, \ldots, |T| \end{aligned}$$

The assignment $x_j = 1$ indicates that the j^{th} tour of T is chosen. By extension, the solution space may be enriched by first adding tours containing only one customer ($c_j = d_{0\ j-|T|} + d_{j-|T|\ 0}, j = |T| + 1,\ldots, |T| + n$) and second by replacing the equalities in the constraints by inequalities (allowing customers to be visited more than once when distances satisfy the triangle inequality, since then customers visited more than once can simply be removed from the inappropriate tours).

However, these extensions make the problem harder to solve and, practically, do not improve the quality of the solutions built. We propose instead to include in the model only the subset of single customer tours that have been produced by the local search. There is no merit embedding this post-optimization technique in the diversification and intensification method, since this would not enrich the set T with new tours. Thus, the appropriate application of this partitioning approach is at the conclusion of the iterated search efforts, which is why we call it a “post-optimization” technique.

4. COMPUTATIONAL RESULTS

4.1. Diversification and intensification technique

In this section, we analyse the behaviour of the diversification and intensification technique when used with our previous taboo searches (Taillard (1993) and Rochat and Semet (1994)). First, we identify the parameter values of our numerical experiments.

- The number of initial solutions, I, is set to 20.
- The number L of tours kept in the set T is set to 260.
- The number of iterations performed by the local search at steps (a) and (4) has to be fixed : for the elementary VRP, we have chosen to perform six decompositions of the problem, followed by optimizations, this means a total number of $14n$ iterations, where n is the number of customers ; for the VRPTW, we perform 2000 iterations.

These parameters were chosen in a relatively arbitrary fashion and we have not undertaken to fine tune their values ; therefore, it is likely the results that follow can be improved. All tests were performed on a Silicon Graphics Indigo (100 Mhz) and all the computation times are given in seconds.

Elementary vehicle routing problem

In table 1, we give the computation times for the elementary VRP required by the previous local search approach of Taillard (1993) and by our new method to produce solution at three different average levels of quality. Specifically, these levels are selected to be 5%, 2% and 1% above the best known solution of certain instances proposed by Christofides et al. (1979), (without tour length limit), Fisher (1994), (with 71 and 134 customers) and Taillard (1993), (with 385 customers). We have run the initial algorithm and our new method five times. A time is printed in bold characters if it is significantly (more than 1.5 times) smaller than the time required by the alternative method shown. We refer to the local search approach of Taillard (1993), as adapted to the current application, a first level local search, because we also embed it (with smaller iteration limits) in our diversification and intensification method.

Problem size	First level local search			Diversification and intensification		
	5%	2%	1%	5%	2%	1%
50	2.8	7.3	11	2.8	7.3	11
75	1.5	9.4	**27**	1.5	11	68
100	15	94	**420**	15	**57**	900
100b	**33**	**140**	**180**	51	280	350
120	>5000	—	—	**690**	**2100**	**2700**
150	30	**250**	>3900	25	670	**1800**
199	110	**1100**	>14000	89	>3700	—
71	**58**	>1400	—	190	**1080**	**1130**
134	32000	>35000	—	**520**	**2300**	**3100**
385	**370**	8800	>23000	1300	**5400**	**18000**

Table 1 : Computation times needed to find solutions that are, on average, a specified per cent above the best known value ; problems of Christofides et al. (1979), Fisher (1994) and Taillard (1993).

We see that, for most instances of Christofides et al. (1979), our new technique is generally not competitive with the first level local search (at least for the parameters we have chosen). However, the instances with 120, 71, 134 and 385 customers are generally better solved with our new technique.

Taillard (1994) has observed that real life quadratic assignment problems may sometimes be very poorly solved by good local search methods. Real-life problems are not necessarily the most difficult to solve (the local searches sometimes find the best known solution extremely rapidly, though not at each run), but the local searches may often be trapped in very poor local optima. Therefore, the average solution values they produce are not very good.

We think that similar behaviour occurs for the VRP, as suggested by the preceding results for the instances of Christofides et al. (1979) with 120 customers, and the selected instances of Fisher (1994) and Taillard (1993). We have tried to generate problems with these challenging characteristics by using a procedure similar to the one described in Taillard (1994) to generate non-uniform quadratic assignment problems : the customers are located on the plane (we have used Euclidean distances) and are spread in several clusters. The number of such clusters and their compactness can be quite variable. The quantities ordered by the customers are exponentially distributed (so, one customer may use almost the entire capacity of one vehicle). We have generated four problem of each size : 75 customers (9 or

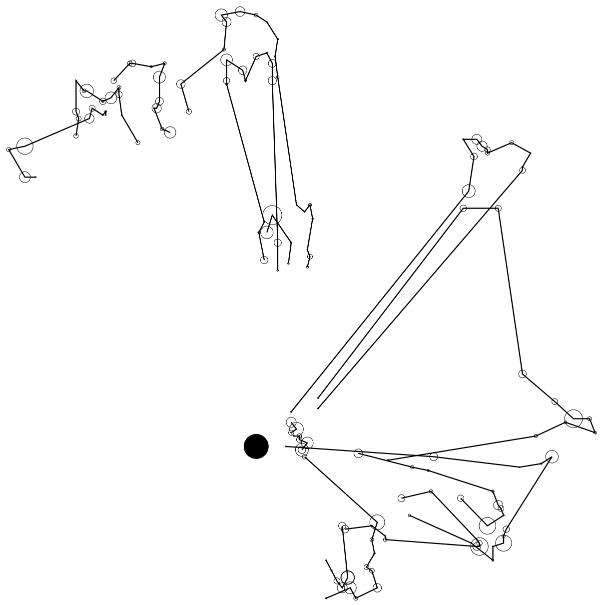

Figure 3 : Best known solution for a non-uniform problem with 150 customers.

10 vehicles), 100 customers (11 or 12 vehicles) and 150 customers (14 or 15 vehicles).

In figure 3, we give the best solution we have found for problem 150a. This figure is to be compared with figure 2 in which we give the best known solution for the problem of Christofides et al. (1979) with 199 customers. We see that the structures of these problems and their solutions are very different. The solution structure of problem 150a may also be compared to the real-life problems proposed by Fisher (1994), and we see that the structures are similar.

In table 2, we provide the same statistics for non-uniform problems as in table 1. We see that our new technique may vastly improve the performance of the local search on these problems, especially when very good solutions are required. In addition, we show in this table that the non-uniform problems we have generated are not only ill-conditioned for our first level local search (Taillard, 1993) but also for the taboo search of Gendreau et al. (1994) : we give the computation times needed by this method and the solution quality it has found (measured in per cent above the best known solutions). We see that the algorithm is very often trapped by local optima of poor quality, even for the problems with 75 customers. It is shown in Taillard (1993) that the algorithm of Gendreau et al. (1994) is superior for a few problem instances of Christofides et al. (1979).

All the computational results given here are for sequential computers ; our new technique may easily be parallelized and an implementation on distributed computers would reduce the computation times by one order of magnitude or more, depending on the number of processors.

Vehicle routing with time windows

We now present computational results obtained on Solomon's VRPTW instances. Each of these 56 instances has 100 customers and the travel time between customers are equal to the corresponding Euclidean distance. As these problems vary in fleet size, vehicle capacity, travel time of vehicles, spatial distribution of customers, time window density and width they are divided into three categories : R-type (uniformly distributed customers), C-type (clustered customers) and RC-type (a mix of R and C types). Two sets of problems are proposed for each of these three categories. Problem sets R1, C1 and RC1 have narrow scheduling horizon while problem sets R2, C2 and RC2 have large scheduling horizon. Narrow scheduling horizon problems have vehicles with small capacities and short route times hence only a few customers can be served by the same vehicle. Conversely, large scheduling horizon problems

Problem size	First level local search			Diversification and intensification			Gendreau et al. (1994)	
	5%	2%	1%	5%	2%	1%	Time	%
75a	**3.3**	30	170	6.5	**18**	**48**	1900	1.2
75b	**2.8**	14	170	6.4	13	**17**	1700	0.7
75c	15	> 2100	—	14	**120**	**170**	1600	9.6
75d	**4.4**	29	52	7.5	**13**	**17**	1700	1.1
100a	13	**53**	> 2000	10	220	**720**	3600	4.6
100b	21	110	460	**15**	**38**	**150**	2900	2.3
100c	**35**	190	> 6800	55	**140**	**770**	2900	5.1
100d	25	**120**	> 2100	28	180	**680**	3300	4.7
150a	> 3700	—	—	**480**	**1600**	**2800**	7000	4.3
150b	78	> 3100	—	**25**	**390**	**1100**	8600	2.8
150c	**33**	> 2800	—	84	**420**	**820**	6400	8.1
150d	100	2000	> 3800	96	**980**	**1700**	5800	5.0

Table 2 : Computation time needed to find solutions that are, on average, a specified per cent above the best known value and performances of the algorithm of Gendreau et al. (1994) ; non-uniform problems.

Type	CPU time	Initial taboo search Minimum		Average		Maximum		Diversification and intensification	
R1	450	12.75	1204.81	13.00	1225.62	13.33	1255.71	12.83	1208.43
	1300	12.67	1206.54	12.92	1223.74	13.17	1259.10	12.58	1202.31
	2700	12.67	1204.76	12.92	1222.36	13.17	1257.72	12.58	1197.42
C1	540	10.00	831.24	10.00	838.93	10.00	863.10	10.00	832.59
	1600	10.00	831.01	10.00	838.00	10.00	861.41	10.00	829.01
	3200	10.00	830.32	10.00	836.87	10.00	858.75	10.00	828.45
RC1	430	12.63	1383.06	13.00	1418.58	13.25	1475.49	12.75	1381.33
	1300	12.25	1396.48	12.88	1417.92	13.25	1470.99	12.50	1368.03
	2600	12.25	1385.28	12.77	1418.36	13.25	1470.99	12.38	1369.48
R2	1600	3.36	991.20	3.62	996.03	3.91	1015.97	3.18	999.63
	4900	3.36	988.75	3.62	992.48	3.91	1010.86	3.09	969.29
	9800	3.36	988.43	3.62	990.09	3.91	1004.07	3.09	954.36
C2	1200	3.00	594.64	3.10	616.44	3.50	654.99	3.00	595.38
	3600	3.00	591.43	3.00	611.25	3.00	654.91	3.00	590.32
	7200	3.00	591.43	3.00	610.28	3.00	653.38	3.00	590.32
RC2	1300	4.00	1171.18	4.18	1249.80	4.38	1422.50	3.62	1207.37
	3900	4.00	1166.78	4.18	1245.06	4.38	1417.02	3.62	1155.47
	7800	4.00	1166.78	4.18	1244.77	4.38	1417.02	3.62	1139.79

Table 3 : Average solution values after given computation times for the initial taboo search and for the diversification and intensification method. Problems of Solomon (1987).

use vehicles with large capacities and long travel times so more customers can be served by the same vehicle. In table 3, we compare the solutions values produced by five runs of the initial taboo search for the VRPTW to those produced by one run of the diversification and intensification method. For each type of problem, we give : the average CPU time needed to perform 50’000, 150’000 and 300’000 iterations of taboo search (for the diversification and intensification technique, this CPU time corresponds to 25, 75 and 150 calls to a first level taboo search performing 2’000 iterations), the average solution value of the best out of five runs, the average value of all runs and the average solution value of the worst run. The first number is the average number of vehicles and the second is the average distance travelled. Again, we see that the diversification and intensification technique vastly improves the local search. The best over five of the first level taboo search runs is often worse than the average solution produced by the diversification and intensification technique, this means that the new method is more efficient and more robust, at least for the longest runs. We have also observed that, when running the method several times, the gap between the best and the worst run is much smaller for the diversification and intensification than for the first level taboo search. Therefore, the new method is more robust and it was not necessary to have average solutions on several runs in table 3.

4.2. Post-optimization procedure

The post-optimization procedure has allowed us to improve the quality of many best known solutions of problem instances of the literature. The improvements that can be obtained with the post-optimization procedure are not very large (generally less than one per cent), but these improvements may often be obtained with a modest increase in computation time. We illustrate this fact on the new elementary VRP instances we propose. We consider 5 runs of our diversification and intensification technique with I = 20 initial solutions and 50 (respectively 70) diversification and intensification steps for the problem instances with 100 (respectively 150) customers. All solutions produced by the first level local searches (i. e. 350 or 450 for each problem) have been stored. Then, for each problem instance, we consider a set T that contains 250 different tours chosen from the best solutions produced by the diversification and intensification procedure. Finally, we have solved the set partitioning problem induced by T in an exact way, using the cplexmip software (Cplex inc., 1993).

Problem size	Diversification and intensification Value	Time	Post-optimization Value	Time	Best known solution
100a	2062.27	1000	2047.90	5.5	2047.90
100b	1940.61		1940.61	3.9	1940.61
100c	1421.59		1407.44	71	1407.44
100d	1581.25		1581.25	1.2	1581.25
150a	3077.77	2100	3070.91	0.43	3055.23
150b	2735.16		2733.60	1.9	2727.99
150c	2367.65		2364.31	18	2362.79
150d	2668.34		2663.20	22	2655.67

Table 4 : Comparison of the solutions produced with the diversification and intensification technique and with the post-optimization procedure.

In table 4, we give, for the non-uniform problems : the number of customers, the value of the best solution produced by the diversification and intensification procedure over 5 runs, the average computation time of one run of the diversification and intensification procedure, the value of the solution produced by the post-optimization procedure, the computation time of the post-optimization procedure and finally we give the best known solution value. In this table, we see that the diversification and intensification procedure has found the best known solution on half of the problems instances with 100 customers. The post-optimization procedure has succeeded in improving the other solutions and in finding the best known solutions. For the problem instances with 150 customers, we see that the average gap between the best solution produced by the diversification and intensification procedure and the best known solution has been reduced by nearly 40%. The running times of the post-optimization procedure are often very low but the standard deviation, for a given problem size, is very high. So, this post-optimization procedure makes it possible to rapidly improve the solutions produced by a local search but the set partitioning problems generated varies enormously.

4.3. Best known solutions

Our diversification and intensification technique, accompanied by the post-optimization procedure, has allowed us to improve the quality of several of the best solutions reported in the literature. Table 5 gives, for elementary VRP instances : the reference where the problem is described, the number of the problem in this reference (when available), the size of the problem, the value of the best solution published (with the reference reporting the best solution published) and finally the value of the best solution obtained during the elaboration of this paper.

Table 6 gives the same information for the Solomon's VRPTW instances, but the origin of the problem (Solomon, 1987) and the size of the problem (100) are not reported and the value of the solutions is a pair (number of vehicles and distance travelled). We see in this table (bold characters) that the taboo search we

Problem source	Problem number	Size	Best solution published	New best
Fisher (1994)	12	134	1163.60[a]	1162.96
Christofides et al. (1979)	5	199	1298.79[b]	1291.45
Christofides et al. (1979)	10	199	1397.94[b]	1395.85
Taillard (1993)	—	385	24599.6[b]	24435.5

Table 5 : Best solutions published and new best solutions found for elementary VRP instances.

a. Fisher (1994)
b. Taillard (1993)

Number	Best solution published		Best solution taboo search		Best solution we found	
R101	18	1607.70[a]	19	1656.20	(19	1650.80)
R102	17	1434.00[a]	18	1477.41	(17	1486.12)
R103	13	1207[b]	14	1222.90	(14	1213.62)
R104	10	1048[b]	**10**	**1013.26**	10	982.01
R105	14	1420.94[c]	**14**	**1404.75**	14	1377.11
R106	12	1350[b]	**12**	**1293.92**	12	1252.03
R107	11	1146[b]	**11**	**1085.77**	10	1159.86
R108	10	989[b]	10	965.28	9	980.95
R109	12	1205[c]	**12**	**1186.41**	11	1235.68
R110	11	1105[b]	11	1107.90	11	1080.36
R111	10	1151[b]	11	1070.90	10	1129.88
R112	10	992[b]	**10**	**965.66**	10	953.63
C101	10	827.30[a]	**10**	**828.94**	10	828.94
C102	10	827.30[a]	**10**	**828.94**	10	828.94
C103	10	835[b]	**10**	**828.06**	10	828.06
C104	10	840[b]	10	841.59	10	824.78
C105	10	828.94[c]	**10**	**828.94**	10	828.94
C106	10	827[a]	**10**	**828.94**	10	828.94
C107	10	827.30[a]	**10**	**828.94**	10	828.94
C108	10	827[a]	**10**	**828.94**	10	828.94
C109	10	828.94[c]	**10**	**828.94**	10	828.94
RC101	14	1669[b]	15	1737.03	(15	1623.58)
RC102	13	1557[b]	**13**	**1480.66**	13	1477.54
RC103	11	1110[b]	11	1264.30	(11	1262.02)
RC104	10	1204.07[c]	**10**	**1157.23**	10	1135.83
RC105	14	1602[b]	15	1543.16	13	1733.56
RC106	12	1485.67[c]	**12**	**1415.62**	12	1384.92
RC107	11	1274.71[c]	**11**	**1262.43**	11	1230.95
RC108	10	1281[b]	11	1149.64	10	1170.70
R201	4	1354[b]	4	1485.36	4	1281.58
R202	3	1530.49[c]	**4**	**1101.49**	(4	1088.07)
R203	3	1126[b]	4	912.98	3	948.74
R204	2	914[d]	3	824.62	2	869.29
R205	3	1128[b]	3	1205.58	3	1063.24
R206	3	833[b]	3	956.05	(3	912.97)
R207	3	904[b]	**3**	**814.84**	3	814.78
R208	2	759.21[c]	3	708.78	2	738.60
R209	2	855[b]	4	901.88	(3	944.64)
R210	3	1052[b]	3	1087.32	3	967.50
R211	3	816[b]	**3**	**794.46**	2	949.49
C201	3	591.56[c]	3	591.56	3	591.56
C202	3	591.56[c]	**3**	**591.56**	3	591.56
C203	3	591.55[c]	**3**	**591.17**	3	591.17
C204	3	590.60[c]	3	597.76	3	590.60
C205	3	588.88[c]	**3**	**588.88**	3	588.88
C206	3	588.49[c]	**3**	**588.49**	3	588.49
C207	3	588.32[c]	3	588.49	3	588.29
C208	3	588.49[c]	**3**	**588.49**	3	588.32
RC201	4	1249[b]	5	1469.73	(4	1438.89)
RC202	4	1221[b]	4	1443.66	4	1165.57
RC203	3	1203[b]	4	1013.99	3	1079.57
RC204	3	897[b]	**3**	**843.12**	3	806.75
RC205	4	1389[b]	5	1286.70	4	1333.71
RC206	3	1213[b]	4	1207.76	3	1212.64
RC207	3	1181[b]	4	1079.07	3	1085.61
RC208	3	919[b]	3	919.83	3	833.97

Table 6 : Best solutions published, best solutions found with taboo search and new best solutions we found for the Solomon's problems.

a. Desrochers et al. (1992)
b. Thangiah et al. (1994)
c. Potvin & Bengio (1994)
d. Chiang & Russel (1994)

used (5 initial solution, 400’000 iterations) has improved or reached the quality of about 27 out of 56 best solutions previously published. Due to rounding and truncating, the length quoted in the publications may be inaccurate ; we report our results on real, double-precision distances. The last digit has been rounded. The last column reports the best solution we have found during our computational experiments (in most cases with the new method). In this column, when a solution is not the best known, we have put it in parentheses. We see that we did not reach or improve a best solution only 9 times.

Finally, we report the best solution values we found for the problem instances D417 and E417 (417 customers, Russel, 1995) : For D417 we have found 54 / 6264.80 and 55 / 3467.83 ; for E417, we have found 54 / 7211.83 and 55 / 3693.24. We were not able to substantially reduce the total length for the solutions with 54 vehicles, therefore we give also the solution length when using 55 vehicles. Thangiah et al. (1994) report the solution values 54 / 4866 (D417) and 55 / 4149 (E417).

5. CONCLUSIONS

We have presented a probabilistic technique that allows us to diversify, intensify and parallelize almost any local search for almost any VRP. This technique makes the local search more robust since it converges more often toward solutions whose quality is close to that of the best known solution. This technique has several advantages :

First, it is relatively easy to design a local search that locally finds good tours, but it is hard to design a search that finds good tours for all customers simultaneously ; our technique makes it possible to overcome this difficulty and to design a fairly robust method more easily.

Second, this technique may be applied to several types of VRPs, for example those including the following constraints :

- Time windows for the customer deliveries.
- Differentiated vehicles (cost of use per kilometre, volume capacity, carrying capacity).
- Constraints on the tours (maximum length, driver breaks, customers that cannot be reached by any vehicle).
- Backhauls.
- Multiple depots.

Third, this technique may easily be parallelized with an arbitrary number of processors (not depending on problem size). But we have shown that it can be interesting to use it even on a sequential computer.

Finally, the post-optimization procedure we propose allows the solutions to be improved with a modest increase in computation time.

Acknowledgements

The authors would like to thank Gilbert Laporte whose valuable suggestions have improved the presentation of this paper. This work was partly supported by an NSERC International Post-Doctoral Fellowship and by an NSERC Strategic Grant on *Parallel Software for Intelligent Vehicle-Highway System*.

REFERENCES

W.-C. Chiang and R. Russell, (1993), *Simulated Annealing Metaheuristics for the Vehicle Routing Problem with Time Windows*, working paper, Department of Quantitative Methods, University of Tulsa, Tulsa, OK 74104.

N. Christofides, A. Mingozzi and P. Toth, (1979), *The vehicle routing problem*, in : Combinatorial optimization (N. Christofides, A. Mingozzi, P. Toth and C. Sandi, eds), Wiley, Chichester.

Cplex Optimization inc. (1993), *Using the CPLEX Callable Library and CPLEX Mixed Integer Library*, Incline Village, NV.

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

M. Desrochers, J. Desrosiers and M. Solomon (1992), *A new Optimization Algorithm for the Vehicle Routing Problem with Time Windows*, Operations Research 40, 342 – 354.

M. L. Fisher (1994), *Optimal solution of vehicle routing problems using minimum K-trees*, Operations Research 42, 626 – 642.

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

M. Gendreau, G. Laporte and J.-Y.Potvin (1995), *Local search algorithms for the vehicle routing problem*, in : Local Search Algorithms, (J. K. Lenstra and E. H. L. Aarts, eds), Wiley, Chichester.

B. E. Gillett and J. Johnson, *Sweep algorithm for the multiple depot vehicle-dispatch problem*, ORSA National Meeting, San Juan, 1974.

F. Glover, (1975), *Surrogate constraint duality in mathematical programming*, Operations Research 23, 434 – 451.

F. Glover, (1977), *Heuristics for Integer Programming Using Surrogate Constraints*, Decision Sciences, Vol. 8, No. 1, 156 – 166.

F. Glover, (1990), *Tabu Search — Part II*, ORSA Journal on Computing 2, 4 – 32.

J. H. Holland, (1976), *Adaptations in Natural and Artificial Systems*, University of Michigan press, Ann Arbor.

J.-Y. Potvin, S. Bengio, (1994), *A Genetic Approach to the Vehicle Routing Problem with Time Windows*, Publication CRT-953, Centre de recherche sur les transports, Université de Montréal.

J.-Y. Potvin, T. Kervahut, B.-L. Garcia, J.-M. Rousseau, (1993), *A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows*, Publication CRT-855, Centre de recherche sur les transports, Université de Montréal.

Y. Rochat, F. Semet, (1994), *A tabu search approach for delivering pet food and flour in Switzerland*, Journal of the Operational Research Society 45, 1233 – 1246.

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

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

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

É. D. Taillard, (1993), *Parallel Iterative Search Methods for Vehicle Routing Problems*, Networks 23, 661 – 676.

É. D. Taillard, (1994), *Comparison of heuristic methods for the quadratic assignment problem*, Publication CRT-989, Centre de recherche sur les transports, Université de Montréal.

S. R. Thangiah, I. H. Osman and T. Sun, (1994), *Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows*, working paper UKC/OR94/4, Institute of Mathematics and Statistics, University of Kent, Canterbury.

P. M. Thompson and H. Psaraftis, (1989), *Cyclic Transfer Algorithms for Multi-Vehicle Routing and Scheduling Problems*, Operations Research 41, 935 – 946.

Appendix : few best known solutions.

Number of customers	Tours	Length	Total capacity
15	89–166—60—84—17–113– –86–140—38—14—43—15– –57–144–137	117.07	194
14	59–151—92—37—98–100– 193—91—85—93–104—99– –96—6	57.16	197
14	82—46–124–168—47—36– 143—49—64—11–175–107– –19–123	130.76	200
14	101–122—20–188—66—71– –65–136—35–135–164—34– –78—50	122.72	200
14	195–134–163—24—29–121– 169–129—79–185–158—3– –77–184	83.23	195
13	2–178–115–145—41—22– 133—75—74–171—73—21– 105	70.37	197
13	54–130–165—55—25–170– –67—23–186—56–197—72– –40	99.52	200
13	189—10–108—90–126—63– 181—32–131–160–128—30– –70	97.09	200
13	5–173—61—16–141–191– –44–119–192–142—42–172– –87	83.14	200
12	146—88–148–159—62–182– –48—7–194–106–153—52	73.43	200
11	26–149–179–155—4–139– 187—39–110–198–180	71.21	194
11	76–196–116—68—80–150– 177–109—12–138–154	49.69	199
11	27–167–127–190—31–162– –69–132–176–111—28	47.26	198
10	147–118—83–199–125—45– 174—8–114—18	66.13	195
10	1—51–103–161—9–120– –81—33–157–102	77.09	199
10	112–183—94—95—97–117– –13—58–152—53	41.08	199
1	156	4.47	19

Table 7 : Best known solution to the problem instance of Christofides et al. (1979) with 199 customers, tour length not limited. Total length : 1291.45.

Number of customers	Tours	Length	Total capacity
7	120–109–108–107–106–114– 115	335.34	2209
11	46–118—18–132–116–131– 117–119–130—65—19	205.33	2029
41	91—21—25—26—27—28– –92—29—93—94—45—44– –43—40—3—41—42—2– —4—5—6—7—8—9– –10—12—11—14—88—15– –13—16—90—89—87—86– –85—84—83—20—82	187.06	2145
27	72—75—1—62—53–102– 104–101—35—36—99–100– –98—37—95—39—38—96– –97–105—57—56–103—55– –54—61—60	64.89	2066
13	73—74–134—76—77—64– –63—79—67—80—33—71– –66	89.42	1864
15	47—32—34—48—49—50– –51—52—58—30—31—59– –23—24—22	55.63	2140
20	78–133—68—70—69–110– 111–125–112–126–124–123– 122–121–127–128–129–113– –81—17	225.29	2167

Table 8 : Best known solution to the problem instance of Fisher (1994) with 134 customers. Total length : 1162.96.

Number of customers	Tours	Length	Total capacity
11	52—7—11—62—88—31– –10—63—90—32—70	116.33	138
11	98—85—91—44—14—38– –86—16–100—37—97	103.56	197
11	42—43—15—41—22—56– —4—74—72—73—58	113.57	128
11	60—83—45—46—8—84– —5—17—61—93—96	112.17	121
10	94—95—92—59—99—6– –87—57—2—13	75.71	152
10	28—27—69—76—40—53– –26—68—24—80	121.70	136
9	21—75—39—23—67—55– –25—54—12	117.84	159
9	1—33—30—20—9—66– –71—35—81	132.76	141
9	47—36—64—49—19—48– –82—18—89	126.64	167
9	51—65—79—78—34—29– —3—77—50	139.58	119

Table 9 : Best known solution to the problem instance r107 of Solomon (1987). Number of vehicles : 10, total length : 1159.86.

Number of customers	Tours	Length	Total capacity
13	95—97—59—96—99—93——5—84—17—45—83—60——89	86.43	182
12	27——1—69—50—76—28——53—26—40—13—94——6	96.27	167
12	92—98—91—44—14—38——86—16—61—85–100—37	106.08	200
12	2—57—15—43—42—87——41—22—74—73—21—58	107.37	129
11	80—24—29—79—81——9——51—33——3—68—12	108.04	172
10	72—75—56—23—67—39——55——4—25—54	125.95	179
10	52——7—82——8—46—36——47—19—48—18	108.10	137
10	70—30—20—66—65—71——35—34—78—77	119.65	134
10	31—88—62—11—49—64——63—90—32—10	123.05	158

Table 10 : Best known solution to the problem instance r108 of Solomon (1987). Number of vehicles : 9, total length : 980.95

Number of customers	Tours	Length	Total capacity
11	95—92—98—42—15—57——87—97—43—13—58	116.22	138
10	59—14—44—38—86—84——93—37–100—91	123.29	172
10	12—76—33—81—78—79——3—50—68—80	94.64	163
10	27—69—31—30—51——9——66—20—70——1	106.27	145
10	2—72—73—21—40—53——26—54—55—25	115.88	118
9	52—88——7—18——8—46——17—60—89	105.52	65
9	39—67—23—75—22—41——74—56——4	120.25	159
9	83——5—61—16—85—99——94——6—96	80.08	160
8	28—29—71—65—35—34——24—77	143.44	99
7	45—82—19—47—36—49——48	124.37	147
7	62—11—64—63—90—32——10	105.71	92

Table 11 : Best known solution to the problem instance r109 of Solomon (1987). Number of vehicles : 11, total length : 1235.68

Number of customers	Tours	Length	Total capacity
51	95——2—21—72—75—23——67—39—12—76—29—79——33—81——9—65—71—51——30—90—63—64—49—36——47—48—46——8—45—84——5——6—94—96—59—97——13—58—26—74—56——4——25—55—54—24—80—68——77——1—70	476.06	800
49	28—27—69—52—82—19——11—62—31—88——7—18——83—61—16—86—38—14——44—85—98—99—92—87——42—43—15—57—41—22——73—40—53——3—78—34——35—66—20—32—10—50——37—93–100—91—17—60——89–	473.44	658

Table 12 : Best known solution to the problem instance r211 of Solomon (1987). Number of vehicles : 2, total length : 949.49