REPORTS
IN
INFORMATICS

ISSN 0333-3590

Path relinking for the vehicle routing problem

Sin C. Ho and Michel Gendreau

REPORT NO 282 October 2004

Department of Informatics
UNIVERSITY OF BERGEN
Bergen, Norway

Path relinking for the vehicle routing problem

Sin C. Ho[*] Michel Gendreau[†]

October 18, 2004

Abstract

This paper describes a tabu search heuristic with path relinking for the vehicle routing problem. Tabu search is a local search method that explores the solution space more thoroughly than other local search based methods by overcoming local optima. Path relinking is a method to integrate intensification and diversification in the search. It explores paths that connect previously found elite solutions. Computational results show that tabu search with path relinking is superior to pure tabu search on the vehicle routing problem.

Keywords: vehicle routing, tabu search, path relinking

[*]Department of Informatics, University of Bergen, N-5020 Bergen, Norway. sin@ii.uib.no

[†]Centre de recherche sur les transports and Département d'informatique et de recherche opérationelle, Université de Montréal, C.P. 6128, succ. Centre-ville, Montréal, Canada H3C 3J7. michelg@crt.umontreal.ca

1 Introduction

The *Vehicle Routing Problem* (VRP) is defined on a complete directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, 1, \ldots, n\}$ is the set of vertices and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ is the arc set. Vertices $1, \ldots, n$ represent customers; with customer i are associated a nonnegative demand d_i and a nonnegative service duration t_i. Vertex 0 is the depot at which is based fleet of m homogenous vehicles of capacity q. To each arc (i, j) is associated a traveling cost or travel time c_{ij}; travel costs satisfy the triangle inequality. The VRP consists of designing m vehicle routes on $\mathcal{G}$ such that (i) every route begins and ends at the depot; (ii) every customer is visited exactly once; (iii) the total demand of any vehicle route does not exceed q; (iv) the total duration (i.e., the total length of the arcs traversed plus the service times of served customers) of any vehicle route does not exceed a given limit L; and (v) the total cost of all vehicle routes is minimized. A comprehensive survey of solution techniques for the VRP can be found in the book edited by Toth and Vigo (2002).

While many heuristic approaches have been used to successfully tackle the VRP during the past five decades (Alfa et al. 1991, Kawamura et al. 1998, Bullnheimer et al. 1998, Bullnheimer et al. 1999, Reimann et al. 2004, Berger and Barkaoui 2003, Prins 2004, Tarantilis 2004), Tabu Search (TS) has been the most widely applied technique to provide good approximate solutions for the problem. This method was originally proposed by Glover (1986) as a local search technique that does not terminate when it encounters local optima, but rather proceeds with moves that degrade the objective function. Cycling is prevented by the use of tabu lists in which the recent history of the search trajectory is recorded. An early TS approach for the VRP was proposed by Osman (1993) who introduced the λ-*interchange* operator, which consists in swapping subsets of at most λ customers between two routes. A rather more involved TS heuristic of the early 90's is *Taburoute* (Gendreau et al. 1994). In Taburoute, the neighborhood operator consists in removing a vertex from its current route, inserting it in a route containing one of its closest neighbors (this could be the same route), and performing a local reoptimization of this route. This method contains some innovative features like the temporary acceptance of infeasible solutions and self-adjusting penalty parameters for infeasible solutions. Taillard (1993) proposed a TS algorithm using a decomposition method that makes it possible to benefit from parallel computing. To this date, this algorithm remains one of the most effective TS heuristic for the VRP. A highly effective method for generating good and diverse solutions is the adaptive memory procedure proposed by Rochat and Taillard (1995). It keeps track of good solutions found during the search and uses them as a basis for the construction of new ones. Xu and Kelly (1996) proposed a TS heuristic based on a network flow model, and defined the neighborhoods by ejection chains. Rego and Roucairol (1996) also used ejection chain based neighborhoods; parallel computing allowed for a more extensive exploration of the search space. Cordeau et al. (2001) proposed a simple and flexible TS heuristic - *Unified Tabu Search.* Its neighborhood structure relies on the *relocate* operator that simply moves one vertex from one route to another. Like Taburoute, it allows intermediate infeasible solutions. The granularity concept proposed by Toth and Vigo (2003) is a candidate list strategy whose purpose is to limit the size of the neighborhood by permanently removing long arcs that have a small probability of belonging to an optimal solution. This method is fast and capable of generating very good solutions. For surveys on VRP metaheuristics, the reader is referred to Gendreau et al. (2002), Cordeau and Laporte (2004) and Cordeau et al. (2004).

While TS has been widely used in VRP heuristics for several years, Path Relinking is a fairly new approach introduced just a few years ago (Glover and Laguna 1993), and has not yet been applied to the VRP. However, it has been applied to other combinatorial problems with great success (Aiex et al. 2003, Aiex et al.

2004, Ghamlouche et al. 2004, Oliveira et al. 2004, Resende and Ribeiro 2003, Souza et al. 2003).

In this paper, we present a new TS heuristic for the VRP that makes use of path relinking as an intensification and diversification mechanism. The paper is organized as follows. In Section 2, we present a basic TS heuristic for the VRP. A brief review of path relinking and how it is combined with TS to solve the VRP is given in Section 3. Experimental results showing the improvements in the performance of the TS heuristic when path relinking is used are presented in Section 4. Finally, Section 5 concludes the paper.

2 The basic tabu search heuristic

Tabu search is a memory-based search strategy that allows the local search process to proceed beyond local optima. This is achieved by allowing the objective function to deterioriate when the current solution is a local optimum, and by keeping track of recent moves or solutions in a so-called tabu list. Whenever the algorithm attempts to move to a solution or to perform a move recorded in the tabu list, the move is banned. This rule prevents cycling and forces other solutions to be explored. However, this feature is not strict, as it can be overridden when some aspiration criterion is satisfied. A commonly used criterion is that the objective function value of a tentative solution be the best ever seen. If this is the case, the search may be allowed to proceed to this tentative solution, since it has obviously never been encountered before, which guarantees that cycling cannot occur.

2.1 Notation of the heuristic

As in Gendreau et al. (1994), we allow the search to be conducted in the infeasible part of the solution space. We let $\mathcal{X}$ denote the set of solutions satisfying constraints (i) and (ii). Each solution $x \in \mathcal{X}$ consists of m vehicle routes starting and ending at the depot, such that every customer is visited exactly once. This solution may violate the capacity and duration constraints.

For a solution x, let $c(x)$ denote its travel cost, and let $q(x)$ and $t(x)$ denote the total violation of the load and duration constraints, respectively. The routing cost of a vehicle k corresponds to the sum of the costs c_{ij} associated with the arcs (i, j) traversed by this vehicle. The total violation of capacity and duration constraints is computed on a route by route basis with respect to q and t. Each solution x is evaluated by a cost function $z(x) = c(x) + \alpha q(x) + \beta t(x)$, where α and β are self-adjusting positive parameters.

2.2 Initial solution

A total of I initial solutions are generated by a stochastic insertion heuristic, which works as follows. The routes are initialized by randomly selecting m seed customers. Each route only services a single customer. The remaining unrouted customers are then inserted one by one (in a random order) at the location that minimizes the cost of inserting this customer over the current set of routes. The solutions thus obtained are usually not very good ones, therefore they need to be improved before proceeding any further. To do so, a short tabu search with the relocate neighborhood operator is applied to each solution for μ iterations. The best solution among these I solutions is chosen to initiate the main search process.

2.3 Neighborhood structure

The neighborhood structure is based on the relocate operator: the neighborhood of solution x, denoted $\mathcal{N}(x)$, is made up of all solutions that can be reached from x by moving a customer i from its route k to another route l. Such a move is denoted by (k, i, l).

2.4 Recency based memory and tabu tenure

To avoid cycling whenever a move (k, i, l) is performed, any move that transfers i back into route k is declared tabu for θ iterations, where θ is a user defined parameter. A tabu move will still be chosen if it satisfies the aspiration criterion of improving the best known solution.

2.5 Frequency-based memory

To diversify the search and to induce it to explore a wider part of the solution space, frequently made moves are penalized. The frequency associated with move (k, i, l) is the number of times customer i has been moved to vehicle l. For any solution $\overline{x} \in N(x)$, whenever $z(\overline{x}) \geq z(x)$, a penalty is added to $z(\overline{x})$. Only moves that lead to non-improving solutions are penalized since it makes no sense to penalize improving moves. The penalty $\phi(\overline{x})$ is defined as $\lambda c(\overline{x})\sqrt{nm'}\vartheta_{ik}$, where ϑ_{ik} denotes the number of times customer i has been moved to vehicle k during the search so far, m' denotes the number of non-empty vehicles in solution $\overline{x}$, and λ is an user defined parameter that controls the intensity of diversification.

2.6 Search process

The tabu search heuristic starts off with the initial solution defined in section 2.2. At each iteration, the least cost non-tabu solution $\overline{x}$ is selected from $\mathcal{N}(x)$. Then parameters α and β are modified. Parameter α is adjusted as follows: if there is no violation of the capacity constraints, the value of α is divided by $1 + \delta$, otherwise it is multiplied by $1 + \delta$, where δ is a positive parameter. A similar rule applies also to β with respect to route duration constraints. This process terminates after γ iterations, and is summarized below.

Algorithm 1 Tabu search

```
If $x$ is feasible, set $x^* = x$ and $c(x^*) = c(x)$; otherwise set $c(x^*) = \infty$.
Set $\alpha = 1$ and $\beta = 1$.
for $i = 1$ to $\gamma$ do
    Select a solution $\overline{x} \in \mathcal{N}(x)$ that minimizes $z(\overline{x}) + \phi(\overline{x})$ and is non-tabu or
    satisfies the aspiration criterion.
    If $\overline{x}$ is feasible and $c(\overline{x}) < c(x^*)$, set $x^* = \overline{x}$ and $c(x^*) = c(\overline{x})$.
    Set the reverse move tabu for $\theta$ iterations.
    Compute $q(\overline{x})$ and $t(\overline{x})$, and update $\alpha$ and $\beta$.
    Set $x = \overline{x}$.
return $x^*$
```

3 Path relinking

Path relinking was first introduced by Glover and Laguna (1993) in connection with TS as a way of exploring trajectories between elite solutions. The fundamental idea behind this method is that good solutions to a problem should share some characteristics. By generating paths (i.e., sequences of intermediate solutions) between elite solutions, one could reasonably hope to find better ones. A more thorough description of Path relinking can be found in Glover (1997), Glover (1998) and Glover et al. (2000).

Path relinking can be interpreted as an evolutionary method where solutions are generated by combining elements from other solutions. Unlike other evolutionary procedures, such as genetic algorithms, where randomness is a key factor in the creation of offsprings from parent solutions, path relinking utilizes systematic, deterministic rules for combining solutions. To generate the desired paths, an *initial solution* and a *guiding solution* are chosen in a so-called *reference set* of elite solutions to represent the starting and the ending points of the path. Attributes from the guiding solution are gradually introduced into the intermediate solutions, so that these solutions contain less characteristics from the initial solution and more from the guiding solution as one moves along the path.

Any path relinking implementation revolves around the following three components that are critical in the design of the algorithm:

- Rules for building the reference set,
- Rules for choosing the initial and guiding solutions,
- A neighborhood structure for moving along paths.

3.1 Building the reference set

The quality (w.r.t. to the objective function of the problem) and the level of diversity of the solutions included in the reference set $\mathcal{R}$ have a major impact on the quality of the generated solutions. In our method, $\mathcal{R}$ is built during TS and enriched during the path relinking phase. We consider five strategies, originally proposed by Ghamlouche et al. (2004), for building $\mathcal{R}$:

Strategy S_1: $\mathcal{R}$ is built with the solutions that at some point during TS become the best overall solution. Here, the idea is to link the overall improving solutions.

Strategy S_2: $\mathcal{R}$ contains the best local minima encountered during the TS phase. This strategy is motivated by the fact that local minimum solutions should share some common characteristics with optimum solutions.

Strategy S_3: This strategy selects $\mathcal{R}$-improving local minima, i.e., local minimum solutions that have a better objective function value than those already in $\mathcal{R}$. The idea here is to introduce the time aspect into the selection process: since usually the better solutions are encountered when the search has been proceeding for some time, this strategy considers less local minima obtained at that stage and thus retains potentially good solutions found early during the search.

Strategy S_4: This strategy accounts both for the attractiveness and the diversity, or *dissimilarity*, of a potential solution when deciding whether or not it should be included in the reference set. Define D_s^b, the level of dissimilarity between solution s and the best solution b, as the number of different arcs between the two solutions:

$$D_s^b = \sum_{(i,j)\in\mathcal{A}} h_{ij},$$

where

$$h_{ij} = \begin{cases} 0, & \text{if } (i,j) \text{ is an arc of both solutions } s \text{ and } b; \\ 1, & \text{otherwise.} \end{cases}$$

We also define the median position of all solutions $x \in \mathcal{R}$ relatively to the best solution b as:

$$Median = \frac{\sum_{x \in \mathcal{R}}^{x \neq b} D_x^b}{|\mathcal{R}| - 1},$$

where $|\mathcal{R}|$ denotes the number of solutions in the reference set. A solution s is included in $\mathcal{R}$ if the solution value of s is better than the value of the best solution b, or if it is better than the solution value of the worst solution in $\mathcal{R}$ and its level of dissimilarity exceeds the median, $D_s^b > Median$. In both cases, the worst solution in $\mathcal{R}$ is replaced by s.

Strategy S_5: This strategy ensures both the quality and diversity of the solutions when building $\mathcal{R}$. Laguna and Armentano (2001) pointed out that this strategy was one of the important lessons they learned in how a reference set should be updated. Starting with a large set of good solutions $\mathcal{P}$, $\mathcal{R}$ is then partially filled with the best solutions found in $\mathcal{P}$ to ensure the quality of the solutions. $\mathcal{R}$ is then extended with solutions that differ significantly from those already in $\mathcal{R}$. The procedure to implement this strategy can be described as follows:

1. Fill $\mathcal{R}$ with $\min\{|\mathcal{R}|, \mathcal{R}_{\max}\}/2$ solutions satisfying strategy S_1, where $\mathcal{R}_{\max}$ is the maximum number of solutions in $\mathcal{R}$.
2. For each solution $s \in \{\mathcal{P} \backslash \mathcal{R}\}$, calculate the level of diversity $\Delta_s^{\mathcal{R}}$ between solution s and all solutions $x \in \mathcal{R}$ where $\Delta_s^{\mathcal{R}} = \sum_{x \in \mathcal{R}} D_x^s / |\mathcal{R}|$.
3. Extend $\mathcal{R}$ with solutions $s \in \{\mathcal{P} \setminus \mathcal{R}\}$ that maximize $\Delta_s^{\mathcal{R}}$.

3.2 Choosing the initial and guiding solutions

The choice of initial and guiding solutions is as important as the contents of $\mathcal{R}$ for the path relinking phase, since the quality of the new generated solutions, and thus the performance of the method, is highly dependent upon these solutions. We followed again the suggestions of Ghamlouche et al. (2004) and examined the impact of the following five selection criteria:

C_1: The guiding and initial solutions are defined as the *best* and *worst* solutions in $\mathcal{R}$, respectively.

C_2: The guiding solution is chosen to be the *best* solution in $\mathcal{R}$, while the initial solution is the *second best* one.

C_3: The guiding solution is chosen as the *best* solution in $\mathcal{R}$, while the initial solution is defined as the solution with *maximum Hamming distance* from the guiding solution.

C_4: The guiding and initial solutions are chosen *randomly* in $\mathcal{R}$.

C_5: The guiding and initial solutions are chosen as the *most distant* solutions in $\mathcal{R}$.

3.3 Moving along paths

The aim of the path relinking phase is to introduce progressively attributes of the guiding solution into solutions obtained by moving away from the initial solution. Identical parts of the two solutions should remain unchanged during the process. In the context of the VRP, it is not always obvious to identify identical parts of the initial and guiding solutions, because similar solutions may exhibit a different numbering of the routes. For instance, route 1 of the initial solution might correspond to route 2 of the guiding solution, with perhaps a few differences. We must therefore make sure that similarities and differences in the structure of the initial and guiding solutions can be properly identified. To do so, once the initial and guiding solutions have been selected, we perform a matching of their routes. The matching procedure amounts to solving an *Assignment Problem* on an auxiliary complete bipartite graph $\mathcal{G}' = (\mathcal{V}', \mathcal{A}')$, where $\mathcal{V}' = \mathcal{V}'_i \cup \mathcal{V}'_g$ and the vertices of $\mathcal{V}'_i$ and $\mathcal{V}'_g$ correspond respectively to the routes of the initial and of the guiding solutions. To each edge $(k,l) \in \mathcal{A}'$ is associated a weight c_{kl}, which is defined as the number of identical customers in routes k and l of the two solutions. We look for a maximum weight matching in $\mathcal{G}'$, i.e., we want to find a matching of the routes of the two solutions such that the number of identical customers in matched routes is maximized. This problem is solved using a greedy approach that can be described as follows:

```
Compute the weights $c_{kl}$ for all edges $(k,l) \in \mathcal{A}'$.
Set $\mathcal{W} = \mathcal{A}'$.
Set $\mathcal{J} = \emptyset$.
**repeat**
  Choose the pair of routes $(k,l) \in \mathcal{W}$ with largest weight $c_{kl}$.
  Set $\mathcal{J} = \mathcal{J} \cup (k,l)$.
  Delete from $\mathcal{W}$ all edges incident to vertex $k \in \mathcal{V}'_i$ and to vertex $l \in \mathcal{V}'_g$.
**until** $\mathcal{W} = \emptyset$.
```

In the path relinking phase, we must make sure that the algorithm is making progress towards the guiding solution. To do so, we use two neighborhoods $\mathcal{N}_1(x)$ and $\mathcal{N}_2(x)$. To simplify the exposition, let us assume that the routes of the guiding solution have been relabeled in accordance with the matching of the routes determined previously, i.e., if $(k,l) \in \mathcal{J}$, we now assign label k to route l of the guiding solution. The first neighborhood, $\mathcal{N}_1(x)$, is made up of all the potential solutions that can be reached from x by moving customers from their current route to another while taking into account the structure of the guiding solution. More precisely, a customer i is eligible to be moved from its current route k if it does not belong to route k in the guiding solution; it will then be relocated into the route l to which it belongs in the guiding solution. $\mathcal{N}_1(x)$ thus contains solutions that are closer to the guiding solution than the current solution x. The second neighborhood, $\mathcal{N}_2(x)$ is defined similarly as the set of all potential solutions that can be reached from x by exchanging two customers i and j between their respective routes while taking into account the structure of the guiding solution. Again, customer i is eligible to be moved from its current route k only if it belongs to a different route l in the guiding solution; it may be swapped with any customer j of route l that does not also belong to route l in the guiding solution. As the neighborhoods of solution x could be fairly restricted if they were limited only to feasible solutions, we allow tunneling through infeasible regions of the solution space. It should be noted that tabus are still imposed during path relinking and that self-adjusting penalties are also used. Algorithm 2 summarizes the path relinking procedure.

Algorithm 2 Path relinking

```
Require: $\mathcal{R}$, the reference set; $x^*$, the current best known solution
  Set $\nu = 0$, $\alpha = 1$ and $\beta = 1$.
  repeat
    Select the initial solution, $s_i$, and the guiding solution, $s_g$, according to criterion $C_*$.
    Determine the maximum weight matching of the routes of $s_i$ and $s_g$.
    Compute $D^{s_i}_{s_g}$.
    Set $x = s_i$.
    repeat
      Select a solution $\overline{x} \in \mathcal{N}_1(x) \cup \mathcal{N}_2(x)$ that minimizes $z(\overline{x})$ and is non-tabu or satisfies the aspiration criterion.
      If $\overline{x}$ is feasible and $c(\overline{x}) < c(x^*)$, set $x^* = \overline{x}$ and $c(x^*) = c(\overline{x})$.
      Set the reverse move tabu for $\theta$ iterations.
      Compute $q(\overline{x})$ and $t(\overline{x})$, and update $\alpha$ and $\beta$.
      Increment $\nu$ by 1.
      Set $x = \overline{x}$.
    until $\nu \geq D^{s_i}_{s_g}$ or $x = s_g$.
    Remove $s_i$ from $\mathcal{R}$.
  until $|\mathcal{R}| \leq 1$
  return $x^*$
```

3.4 Tabu search with path relinking

In the following, we show how tabu search and path relinking are combined in our implementation. Algorithm 3 describes the resulting procedure.

Initially, the reference set $\mathcal{R}$ is empty, and is extended with solutions according to one of the strategies of Section 3.1. Before path relinking is started, $\mathcal{R}$ is sorted and the $\mathcal{R}_{\max}$ best solutions are chosen to be retained in the set. Path relinking is performed every φ iterations of the main TS loop. One round of path relinking consists of generating several paths with different initial and guiding solutions from $\mathcal{R}$. The initial and guiding solutions are chosen according to one of the criteria listed in Section 3.2. Long paths are favored, since they will have a better chance of producing good solutions. If the output of path relinking, solution r, is better than the current best known solution, TS continues with this solution as the current solution. However, if path relinking is not able to find a better solution, a new solution based on the elite solutions found in $\mathcal{R}$ is constructed using the adaptive memory scheme proposed by Rochat and Taillard (1995). This new solution is then improved by a short tabu search before being returned to the main loop. The whole process terminates after γ iterations.

4 Computational experiments

Standard benchmark instances for the VRP were used for experimentation. These include the fourteen classical Euclidean VRP and distance-constrained VRP instances described in Christofides and Eilon (1969) and Christofides et al. (1979). The characteristics of these instances are given in Table 1. For each problem instance, the table indicates the number of customers (n), the number of vehicles (m), the vehicle capacity (q), the route maximum length (L) and the service time (t) for each customer. The table also gives for each problem instance the best known solution so far and the reference to where this solution can be found.

The results given in this paper were obtained with the following parameter

Algorithm 3 Tabu search with path relinking

```
Set $\mathcal{R} = \emptyset$.
Construct an initial solution. Let $x$ be this solution.
If $x$ is feasible, set $x^* = x$ and $c(x^*) = c(x)$; otherwise set $c(x^*) = \infty$.
Set $\alpha = 1$ and $\beta = 1$.
**for** $i = 1$ to $\gamma$ **do**
  Select a solution $\overline{x} \in \mathcal{N}(x)$ that minimizes $z(\overline{x}) + \phi(\overline{x})$ and is non-tabu or satisfies the aspiration criterion.
  If $\overline{x}$ satisfies $S_*$, set $\mathcal{R} = \mathcal{R} \cup \{\overline{x}\}$.
  If $\overline{x}$ is feasible and $c(\overline{x}) < c(x^*)$, set $x^* = \overline{x}$ and $c(x^*) = c(\overline{x})$.
  Set the reverse move tabu for $\theta$ iterations.
  Compute $q(\overline{x})$ and $t(\overline{x})$, and update $\alpha$ and $\beta$.
  Set $x = \overline{x}$.
  **if** mod $(i, \varphi) = 0$ **then**
    $r =$PathRelinking$(\mathcal{R}, x^*)$.
    **if** $c(r) < c(x^*)$ **then**
      Set $x^* = r$, $c(x^*) = c(\overline{r})$ and $x = r$.
    **else**
      $a =$AdaptiveMemory$(\mathcal{R})$ and set $x = a$.
      If $c(a) < c(x^*)$, set $x^* = a$ and $c(x^*) = c(a)$.
**return** $x^*$
```

Table 1: Characteristics of the benchmarks instances used for computational experiments

Problem	n	m	q	L	t	Best known	Ref.
E051-05e	50	5	160	∞	0	524.61	Taillard (1993)
E076-10e	75	10	140	∞	0	835.26	Taillard (1993)
E101-08e	100	8	200	∞	0	826.14	Taillard (1993)
E101-10c	100	10	200	∞	0	819.56	Taillard (1993)
E121-07c	120	7	200	∞	0	1042.11	Taillard (1993)
E151-12c	150	12	200	∞	0	1028.42	Taillard (1993)
E200-17c	199	17	200	∞	0	1291.45	Rochat and Taillard (1995)
D051-06c	50	6	160	200	10	555.43	Taillard (1993)
D076-11c	75	11	140	160	10	909.68	Taillard (1993)
D101-09c	100	9	200	230	10	865.94	Taillard (1993)
D101-11c	100	11	200	1040	90	866.37	Taillard (1993)
D121-11c	120	11	200	720	50	1541.14	Taillard (1993)
D151-14c	150	14	200	200	10	1162.55	Taillard (1993)
D200-18c	199	18	200	200	10	1395.85	Rochat and Taillard (1995)

Table 2: PR frequency performances

φ	S_2, C_1	S_3, C_2	S_4, C_3
5,000	0.95%	0.56%	1.66%
10,000	0.83%	0.54%	0.71%
20,000	1.40%	1.10%	0.88%
30,000	1.08%	0.95%	0.86%
40,000	1.08%	1.05%	1.21%

Table 3: TS and TS with PR results for benchmark instances

Problem	TS 100,000	TS 200,000	Time	TS + PR	Time	Best TS + PR
E051-05e	**524.61**	**524.61**	2.90	**524.61**	1.49	**524.61**
E076-10e	845.49	838.79	10.13	835.77	5.08	835.28
E101-08e	835.15	835.15	13.90	829.42	7.43	827.53
E101-10c	**819.56**	**819.56**	15.83	**819.56**	8.03	**819.56**
E121-07c	1081.45	1081.45	17.67	1046.38	8.96	1046.38
E151-12c	1041.00	1038.92	38.2	1035.76	19.67	1030.51
E200-17c	1324.48	1324.48	85.3	1318.65	43.88	1304.97
D051-06c	556.68	556.68	3.60	**555.43**	1.86	**555.43**
D076-11c	910.05	910.05	11.10	**909.68**	5.67	**909.68**
D101-09c	875.28	**865.94**	16.37	**865.94**	8.67	**865.94**
D101-11c	872.89	866.53	16.77	867.29	9.32	**866.37**
D121-11c	1575.65	1573.11	25.33	1568.12	13.25	1553.58
D151-14c	1187.52	1187.52	51.83	1168.71	26.97	1165.02
D200-18c	1452.05	1428.24	105.63	1415.85	54.62	1408.76
Average time and deviation from best	1.46%	1.12%	29.61	0.54%	15.35	0.27%

settings: $\mathcal{R}_{\max} = 30$, $\delta = 0.5$, $I = 5$, $\mu = 100$, $\theta = [7.5 \log_{10} n]$ (where $[y]$ is the nearest integer to y), $\lambda = 0.015$, $\varphi = 10,000$ and $\gamma = 100,000$. The values chosen for parameters δ, θ and λ were taken from Cordeau et al. (2001) for a similar tabu search heuristic. The frequency of path relinking, φ, was determined by running every benchmark instance once for each of the predetermined values of φ and for three different combinations of criteria and strategies. This calibrating process is important because if path relinking is performed too frequently, the search will tend to focus too much on a small portion of the search space. On the opposite, if it is performed very rarely, its impact will be negligible. It is important to find a balance between these two extremes. Table 2 shows the average deviations from the best known solutions for the different values of φ, and it identifies $\varphi = 10,000$ as the value yielding the best results. Thus this value was used in all the experiments reported in the remainder of this section. The other parameters were not tuned prior to the experimentation. The heuristic was coded in C++ and all experiments were performed on a Pentium 4, 2.53 GHz computer.

We want to investigate the benefits of integrating path relinking into tabu search. This is done by performing for each instance a single run of the pure tabu search heuristic, as well as of the different variants of tabu search with path relinking. The main results obtained are reported in Table 3. In this table, the first column

Table 4: Average improvement over pure tabu search (100,000 iterations)

	C_1	C_2	C_3	C_4	C_5
S_1	0.70%	0.60%	0.59%	0.67%	0.72%
S_2	0.61%	0.78%	0.69%	0.63%	0.69%
S_3	0.67%	0.92%	0.75%	0.65%	0.73%
S_4	0.68%	0.62%	0.73%	0.73%	0.73%
S_5	0.56%	0.65%	0.74%	0.60%	0.62%

Table 5: Average improvement over pure tabu search (200,000 iterations)

	C_1	C_2	C_3	C_4	C_5
S_1	0.37%	0.27%	0.26%	0.34%	0.39%
S_2	0.29%	0.45%	0.36%	0.30%	0.36%
S_3	0.34%	0.57%	0.43%	0.33%	0.40%
S_4	0.36%	0.29%	0.40%	0.40%	0.40%
S_5	0.24%	0.32%	0.42%	0.27%	0.29%

gives the name of the respective benchmark instances while the results for pure tabu search runs of 100,000 iterations and 200,000 iterations are displayed in the two next columns, and CPU times (in minutes) for 200,000 iteration TS runs in the following one. The column *TS + PR* describes the results for the best combination of strategy and criterion, S_3 and C_2, when running tabu search for 100,000 iterations. The last column displays the best solutions among all the different combinations of strategies for 100,000 iterations. The results reported in this table clearly show that running tabu search for 100,000 iterations with path relinking is much more effective than running pure tabu search for 200,000 iterations. In the case of the combination S_3 and C_2, both the average gap and the average running time were reduced by about a half.

As shown in Table 4 and Table 5, every combination of strategy and criterion is capable of improving the results of TS. The tables identify the combination S_3 and C_2 as one giving the best results. This combination considers a set of very good solutions obtained over time, and the focus is on search intensification close to the very best solutions of the reference set. It should be noted that our conclusions differ somewhat from those of Ghamlouche et al. (2004) who obtained their best results with the combination S_3, C_5.

One would expect strategy S_2, which selects local minima, to not perform too well. This is not always the case, one explanation may be that some of the solutions in $\mathcal{R}$ are not "real" minima. As we allow infeasible regions to be explored, and infeasible solutions cannot be a part of $\mathcal{R}$, the closest feasible solutions to the actual minimum solutions are chosen. The best performance of strategy S_2 is obtained jointly with criterion C_2 that builds a path from the second best to the best solution in $\mathcal{R}$. This is the second best combination, and like the first one, it focuses on search intensification (in this case, close to the local minima).

Figure 1 and Figure 2 show how the search evolves for pure TS and for TS with path relinking. The vertical axis is the average deviation of the current best solution from the best known. The figures show that TS has a steeper slope than TS with path relinking at the beginning of the search. This may be explained by the fact that different initial solutions are used to initiate the search (due to the stochastic insertion heuristic used to construct an initial solution). Path relinking is

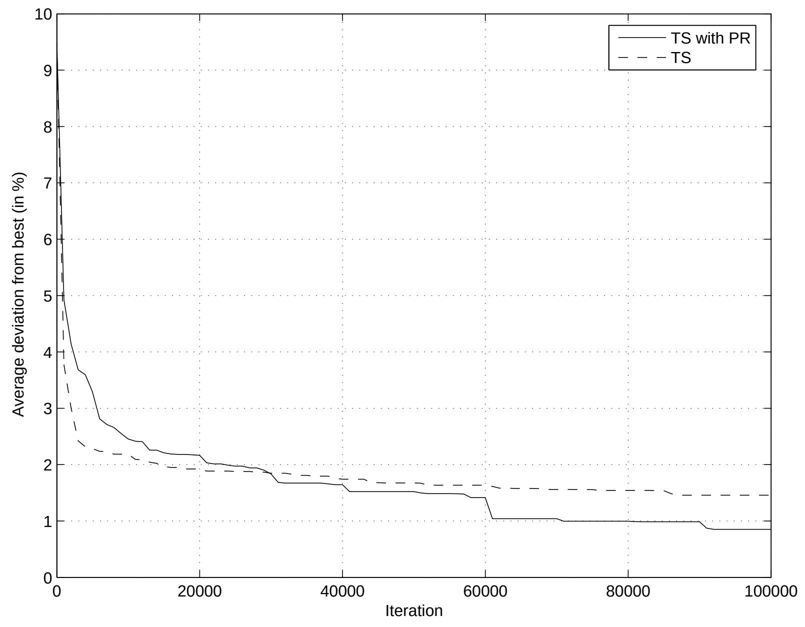

Figure 1: Strategies S_1 and C_3

performed every 10,000 iterations, and the figures show the "breaks" of the graphs around these points. As the graphs for TS show much less improvement over time, it will certainly require more iterations to obtain the same level of solution quality as with path relinking. This confirms that TS with path relinking performs much better than TS on its own.

We have shown that TS with path relinking produces better solutions than pure TS. In the following, we will further assess its performance by comparing it to other methods proposed for solving the VRP.

Table 6 provides a comparison on the results obtained on the fourteen classical instances by some recent metaheuristics for the VRP (the reader is referred to Gendreau et al. (2002), Cordeau et al. (2004) and Cordeau and Laporte (2004) for more results and comparisons). These are Cordeau et al. (2002), Toth and Vigo (2003), Berger and Barkaoui (2003), Prins (2004), Tarantilis (2004) and TS + PR (columns marked CGLPS, TV, BB, P, T and HG). The results are expressed in terms of total distance traveled. The first column refers to the various problem instances. Columns labeled "value" are the results obtained by the various methods, whereas columns labeled "time" specify the run time (in minutes). For our entry in Table 6, we report the results from combination S_3 and C_2, which is the best one.

It is not easy to compare running times for the various metaheuristics due to different computers, compilers, data structures, etc. However, we attempted to scale the computing times on various computers to equal those on a Pentium 2.53 GHz computer, using the factors determined by Dongarra (2004).

Table 6 shows that tabu search with path relinking is competitive with other VRP methods with respect to the quality of the results that it produces.

5 Conclusion

In this paper, we presented a tabu search heuristic with path relinking for the vehicle routing problem. Path relinking uses previously encountered good solutions

Table 6: Comparison of some recent metaheuristics for the VRP

	CGLPS		TV		BB	
Problem	Value	Time[1]	Value	Time[2]	Value	Time[3]
E051-05e	**524.61**	4.57	**524.61**	0.81	**524.61**	2.00
E076-10e	835.45	7.27	838.60	2.21	**835.26**	14.33
E101-08e	829.44	11.23	828.56	2.39	827.39	27.90
E101-10c	**819.56**	10.99	**819.56**	1.10	**819.56**	7.21
E121-07c	1074.13	14.15	1042.87	3.18	1043.11	22.43
E151-12c	1038.44	18.72	1033.21	4.51	1036.16	48.98
E200-17c	1305.87	28.10	1318.25	7.50	1324.06	55.41
D051-06c	**555.43**	4.61	**555.43**	0.86	**555.43**	2.33
D076-11c	**909.68**	7.55	920.72	2.75	**909.68**	10.50
D101-09c	866.38	11.17	869.48	2.90	868.32	5.05
D101-11c	866.53	10.65	**866.37**	1.41	**866.37**	4.73
D121-11c	1568.91	14.53	1545.51	9.34	1553.12	34.91
D151-14c	1171.81	19.17	1173.12	5.67	1169.15	17.88
D200-18c	1415.40	29.74	1435.74	9.11	1418.79	43.86
Average deviation from best and time	0.69%	13.75	0.64%	3.84	0.48%	21.25
Scaled time		2.63		0.12		1.34

	P		T		HG	
Problem	Value	Time[4]	Value	Time[3]	Value	Time[5]
E051-05e	**524.61**	0.01	**524.61**	0.41	**524.61**	1.49
E076-10e	**835.26**	0.77	**835.26**	4.86	835.77	5.08
E101-08e	**826.14**	0.46	**826.14**	8.57	829.42	7.43
E101-10c	**819.56**	0.05	**819.56**	0.29	**819.56**	8.03
E121-07c	**1042.11**	0.30	**1042.11**	0.32	1046.38	8.96
E151-12c	1031.63	5.50	1029.64	9.90	1035.76	19.67
E200-17c	1300.23	19.11	1311.48	13.12	1318.65	43.88
D051-06c	**555.43**	0.01	**555.43**	0.39	**555.43**	1.86
D076-11c	912.30	1.42	**909.68**	1.02	**909.68**	5.67
D101-09c	**865.94**	0.37	**865.94**	3.76	**865.94**	8.67
D101-11c	**866.37**	0.09	**866.37**	0.38	867.29	9.32
D121-11c	1542.97	10.44	1544.01	7.45	1568.12	13.25
D151-14c	1164.25	7.25	1163.19	6.74	1168.71	26.97
D200-18c	1420.20	26.83	1407.21	21.54	1415.85	54.62
Average deviation from best and time	0.23%	5.19	0.20%	5.63	0.54%	15.35
Scaled time		1.24		0.35		

[1]Sun Ultrasparc 10 440 MHz
[2]Pentium 200 MHz
[3]Pentium 400 MHz
[4]Pentium 1 GHz
[5]Pentium 2.53 GHz

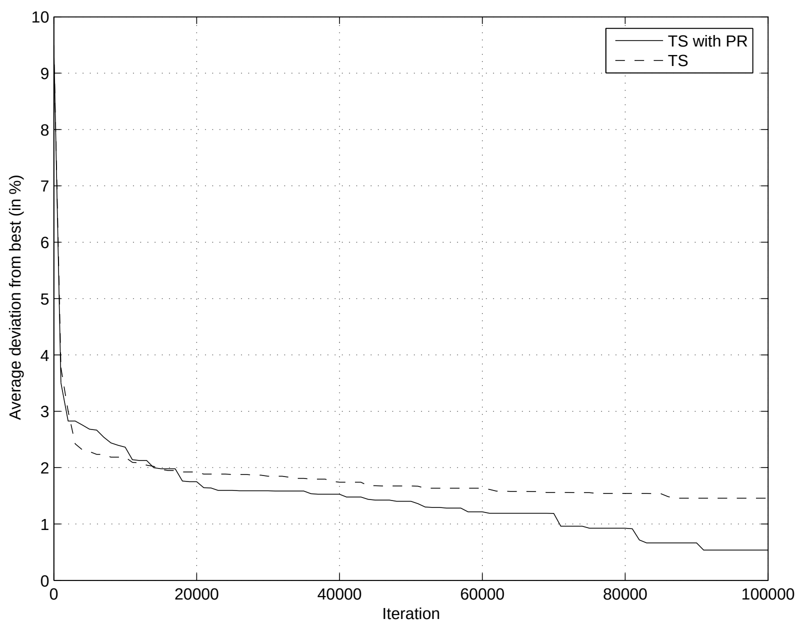

Figure 2: Strategies S_3 and C_2

to obtain diversification and intensification in the search. Using path relinking periodically in the search speeds up the identification of very good solutions. Computational results show that tabu search with path relinking is able to produce better solutions than pure tabu search using much less computing time.

Acknowledgements

This work was supported by the Research Council of Norway under grant 127533/432 and by the National Science and Engineering Research Council of Canada. This support is gratefully acknowledged. We wish to thank Lars Magnus Hvattum for spending time to make the code more efficient.

References

Aiex, R. M., Binato, S. and Resende, M. G. C. (2003), 'Parallel GRASP with path-relinking for job shop scheduling', *Parallel Computing* **29**(4), 393–430.

Aiex, R. M., Resende, M. G. C., Pardalos, P. M. and Toraldo, G. (2004), 'GRASP with path-relinking for the three-index assignment problem', *INFORMS Journal on Computing* . Forthcoming.

Alfa, A. S., Heragu, S. S. and Chen, M. (1991), 'A 3-opt based simulated annealing alogrithm for vehicle routing problems', *Computers & Industrial Engineering* **21**(1-4), 635–639.

Berger, J. and Barkaoui, M. (2003), 'A new hybrid genetic algorithm for the capacitated vehicle routing problem', *Journal of the Operational Research Society* **54**(12), 1254–1262.

Bullnheimer, B., Hartl, R. F. and Strauss, C. (1998), Applying the ant system to the vehicle routing problem, *in* S. Voss, S. Martello, I. H. Osman and C. Roucairol, eds, 'Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization', Kluwer, pp. 109–120.

Bullnheimer, B., Hartl, R. F. and Strauss, C. (1999), 'An improved ant system for the vehicle routing problem', *Annals of Operations Research* **89**(1-4), 319–328.

Christofides, N. and Eilon, S. (1969), 'An algorithm for the vehicle dispatching problem', *Operational Research Quarterly* **20**(3), 309–318.

Christofides, N., Mingozzi, A. and Toth, P. (1979), The vehicle routing problem, *in* N. Christofides, A. Mingozzi, P. Toth and C. Sandi, eds, 'Combinatorial Optimization', Wiley, Chichester, pp. 315–338.

Cordeau, J.-F., Gendreau, M., Hertz, A., Laporte, G. and Sormany, J.-S. (2004), New heuristics for the vehicle routing problem, Technical Report G-2004-33, GERAD, Université de Montréal, Canada.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y. and Semet, F. (2002), 'A guide to vehicle routing heuristics', *Journal of the Operational Research Society* **53**(5), 512–522.

Cordeau, J.-F. and Laporte, G. (2004), Tabu search heuristics for the vehicle routing problem, *in* C. Rego and B. Alidaee, eds, 'Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search', Kluwer, Boston, pp. 145–163.

Cordeau, J.-F., Laporte, G. and Mercier, A. (2001), 'A unified tabu search heuristic for vehicle routing problems with time windows', *Journal of the Operational Research Society* **52**(8), 928–936.

Dongarra, J. J. (2004), Performance of various computers using standard linear equations software, Technical Report CS-89-85, Computer Science Department, University of Tennessee, Knoxville, TN.

Gendreau, M., Hertz, A. and Laporte, G. (1994), 'A tabu search heuristic for the vehicle routing problem', *Management Science* **40**(10), 1276–1290.

Gendreau, M., Laporte, G. and Potvin, J.-Y. (2002), Metaheuristics for the capacitated VRP, *in* P. Toth and D. Vigo, eds, 'The Vehicle Routing Problem', SIAM Society for Industrial and Applied Mathematics, pp. 129–154.

Ghamlouche, I., Crainic, T. G. and Gendreau, M. (2004), 'Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design', *Annals of Operations Research* **131**(1-4), 109–133.

Glover, F. (1986), 'Future paths for integer programming and links to artificial intelligence', *Computers & Operations Research* **13**(5), 533–549.

Glover, F. (1997), Tabu search and adaptive memory programming - advances, applications and challenges, *in* R. S. Barr, R. V. Helgason and J. L. Kennington, eds, 'Interfaces in Computer Science and Operations Research', Kluwer, pp. 1–75.

Glover, F. (1998), A template for scatter search and path relinking, *in* J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer and D. Snyers, eds, 'Artificial Evolution', Vol. 1363 of *Lecture Notes in Computer Science*, Springer, pp. 13–54.

Glover, F. and Laguna, M. (1993), Tabu search, *in* C. R. Reeves, ed., 'Modern Heuristic Techniques for Combinatorial Problems', Blackwell Scientific Publishing, pp. 70–150.

Glover, F., Laguna, M. and Martí, R. (2000), 'Fundamentals of scatter search and path relinking', *Control and Cybernetics* **39**(3), 653–684.

Kawamura, H., Yamamoto, M., Mitamura, T., Suzuki, K. and Ohuchi, A. (1998), 'Cooperative search on pheromone communication for vehicle routing problems', *IEEE Transactions on Fundamentals* **E81-A**, 1089–1096.

Laguna, M. and Armentano, V. A. (2001), Lessons from applying and experimenting with scatter search, *in* C. Rego and B. Alidaee, eds, 'Adaptive Memory and Evolution: Tabu Search and Scatter Search', Kluwer. Forthcoming.

Oliveira, C. A., Pardalos, P. M. and Resende, M. G. (2004), GRASP with path-relinking for the quadratic assignment problem, *in* C. C. Ribeiro and S. L. Martins, eds, 'Efficient and Experimental Algorithms', Vol. 3059 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 356–368.

Osman, I. H. (1993), 'Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem', *Annals of Operations Research* **41**(1-4), 421–451.

Prins, C. (2004), 'A simple and effective evolutionary algorithm for the vehicle routing problem', *Computers & Operations Research* **31**(12), 1985–2002.

Rego, C. and Roucairol, C. (1996), A parallel tabu search algorithm using ejection chains for the vehicle routing problem, *in* I. H. Osman and J. P. Kelly, eds, 'Meta-Heuristics: Theory and Applications', Kluwer, Boston, MA, pp. 661–675.

Reimann, M., Doerner, K. and Hartl, R. F. (2004), 'D-Ants: Savings Based Ants divide and conquer the vehicle routing problem', *Computers & Operations Research* **31**(4), 563–591.

Resende, M. G. C. and Ribeiro, C. C. (2003), 'A GRASP with path-relinking for private virtual circuit routing', *Networks* **41**(2), 104–114.

Rochat, Y. and Taillard, É. D. (1995), 'Probabilistic diversification and intensification in local search for vehicle routing', *Journal of Heuristics* **1**(1), 147–167.

Souza, M. C., Duhamel, C. and Ribeiro, C. C. (2003), A GRASP heuristic for the capacitated minimum spanning tree problem using a memory-based local search strategy, *in* M. G. C. Resende and J. P. Sousa, eds, 'Metaheuristics: Computer Decision-Making', Kluwer Academic Publishers.

Taillard, É. D. (1993), 'Parallel iterative search methods for vehicle routing problems', *Networks* **23**(8), 661–673.

Tarantilis, C. D. (2004), 'Solving the vehicle routing problem with adaptive memory programming methodology', *Computers & Operations Research* . Forthcoming.

Toth, P. and Vigo, D. (2003), 'The granular tabu search and its application to the vehicle-routing problem', *Journal on Computing* **15**(4), 333–346.

Toth, P. and Vigo, D., eds (2002), *The Vehicle Routing Problem*, SIAM Society for Industrial and Applied Mathematics.

Xu, J. and Kelly, J. P. (1996), 'A network flow-based tabu search heuristic for the vehicle routing problem', *Transportation Science* **30**(4), 379–393.