Procedia Computer Science
Volume 29, 2014, Pages 844–854
ICCS 2014. 14th International Conference on Computational Science

Variable Neighborhood Search Based Set covering ILP model for the Vehicle Routing Problem with time windows

Amine Dhahri[1], Kamel Zidi[2], and Khaled Ghedira[3]

[1] National School of Computer Science (ENSI)
Tunis, 2010 Manouba, Tunisia
aminedhahri@gmail.com
[2] University of Gafsa, Faculty of Sciences of Gafsa
Zarroug, Gafsa 2112, Tunisia
kamel.zidi@fsgf.rnu.tn
[3] SOIE-Management Higher Institute
41,Cite Bouchoucha Le Bardo 2000, Tunisia
khaled.ghedira@isg.mu.tn

Abstract

In this paper we propose a hybrid metaheuristic based on General Variable Neighborhood search and Integer Linear Programming for solving the vehicle routing problem with time windows (VRPTW). The problem consists in determining the minimum cost routes for a homogeneous fleet of vehicles to meet the demand of a set of customers within a specified time windows. The proposed heuristic, called VNS-SCP is considered as a matheuristic where the hybridization of heuristic (VNS) and exact (Set Covering Problem (SCP)) method is used in this approach as an intertwined collaborative cooperation manner. In this approach an initial solution is first created using Solomon route-construction heuristic, the nearest neighbor algorithm. In the second phase the solutions are improved in terms of the total distance traveled using VNS-SCP. The algorithm is tested using Solomon benchmark. Our findings indicate that the proposed procedure outperforms other local searches and metaheuristics.

Keywords: Vehicle Routing Problem With Time Windows , variable neighborhood search , matheuristics, ILP

1 Introduction

Supply chain involves many activities like supply and production. For a long time, research on logistics has focused on optimizing these activities since this optimization has eliminated the waste of time. The transport activity is one of the most important activities in logistics. A better organization of this activity mainly vehicle routing presents an economic challenge. Because of this economic importance and priority, researchers have paid great interest to the

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2014

doi:10.1016/j.procs.2014.05.076

vehicle routing problem VRP. The Vehicle Routing Problem (VRP) is considered as an NP-hard combinatorial optimization problem, which was proposed, by Dantzig and Ramser in [8]. VRP is identified as a plan to follow for serving a number of customers by a number of vehicles, knowing that the cost of allocating vehicles to customers must be reduced. There are many variations of VRP, such as the Capacitated Vehicle Routing Problem (CVRP), Vehicle Routing Problem with Pickup and Delivery (VRPPD), Dynamic Vehicle Routing Problem (SVRP) and the Vehicle Routing Problem with Time Windows (VRPTW). Finding near-optimal solutions within a shorter computational time using approximate approaches for the VRPTW problem was the mainly objective for researchers over the years and the main objective of this paper. In [22], Rochat and Taillard used a probabilistic diversification and intensification technique to a local search algorithm for the VRPTW. A tabu search which is a local search-based metaheuristic was implemented in 1997 in [28] to tackle this problem and improve the best known solution. Since the tabu search metaheuristic has been widely used to solve the VRPTW, we can find more research work about it in [19], [6], [3], [13], [17] and [24]. A simulated annealing metaheuristic and its parallel version was successfully designed by Chiang and Russell in [4] and Debudaj-Grabysz in [9]. Results have shown that these two approaches have succeeded in solving a large-scale of VRPTW. As an evolutionary computation (EC), population-based metaheuristic algorithms are widely used to solve the VRPTW and much research effort has been carried out to develop evolutionary algorithms (EAs) to solve the VRPTW. Consider, for instance, the example of paper which is used by Genetic algorithm (GA) to solve the VRPTW, [30]. In [29], a solution is represented as an integer string, genetic operation like crossover and adaptive mutation scheme are used to improve the quality of solutions. The findings showed the effectiveness of the hybrid GA for the VRPTW. Recently, Alvarenga in [1] has hybridized the genetic algorithm with a set partition formulation and. Also Good feasible solutions produced by a General Variable Neighborhood Search in work of Dhahri et al [10].

2 Problem Definition

The VRPTW is defined as follows: given a graph (V,A) where $V = \{0, ..., n+1\}$ a customer set and $A = \{(i;j)|i,j; \in V\}$ an arc set. Node 0 is the depot, and node i is a customer location with demand q_i, service time windows (e_i, l_i) where e_i is the earliest time that service can begin and l_i is the latest time that service can begin and service time s_i. Each arc (i,j) has a non-negative distance d_{ij} . A set K of identical vehicles with capacity Q should attend n customers, represented by vertices $1, .., n$ Consider that $N = V - \{0; n+1\}$. The total demand of each route should not exceed Q.[5];

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ travelled directly from customer } c_i \text{ to } c_j \\ 0 & \text{otherwise} \end{cases}$$

$$\min \quad z = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{v} c_{ij}x_{ijk} \tag{1}$$

$$\text{s.t.} \quad \sum_{k\in K}\sum_{j\in V} x_{ijk} = 1 \quad \forall i \in N \tag{2}$$

$$\sum_{j\in V} x_{0jk} = 1 \quad \forall k \in K \tag{3}$$

$$\sum_{i\in V} x_{ijx} - \sum_{i\in V} x_{jix} = 0 \quad \forall k \in K \tag{4}$$

$$\sum_{j\in V} x_{i(n+1)k} = 1 \quad \forall k \in K \tag{5}$$

$$\sum_{i\in N} q_i \sum_{j\in V} x_{ijk} \leq Q \quad \forall k \in K \tag{6}$$

$$b_{ik} + d_i + c_{ij} - (1 - x_{ijk})M_{ij} \leq b_{jk} \quad \forall k \in K, \quad \forall (i,j) \in A \tag{7}$$

$$e_i \leq b_{ik} \leq l_i \quad \forall k \in K, \quad \forall i \in V \tag{8}$$

$$x_{ijk} \in \{0,1\} \tag{9}$$

The objective function (1) represents the total cost to be minimized. Constraints (2) assure that customer i is delivered only by one vehicle k. Constraints (3)-(5) are flow constraints of the vehicle k; that is, each vehicle leaves the depot, visits the customers, and then, returns to the depot. Constraints (6) guarantee that the vehicle capacity is not exceeded. Constraints (7) and (8) signify the time constraints; to make sure that time window is not violated, where b_{ik} represents the time from which the vehicle k begins to serve the customer i and M_{ij} are large constants. According to [5], M_{ij} can be replaced by $max\,\{e_i + d_i + c_{ij} - e_j, 0\}\,\forall i,j \in A$. Constraints (9) define the domain of the decision variables.

3 Variable Neighborhood Search for the VRPTW

Proposed by Hansen and Mladenovic in [18], variable neighborhood search VNS is a metaheuristic based on the concept of systematic change of neighborhoods during the search, knowing that there are several rules of crucial importance to governor this change. The basic method Basic VNS works in several stages described in 3. Firstly it requires the definition of different neighborhood structures, denoted $\mathcal{V}_1...\mathcal{V}_{MAX}$ generally from different sizes. A relation order between these structures is defined in terms of problem. From an initial solution x, a neighbor of x is selected randomly using the neighborhood structure $\mathcal{V}_1$. If the solution obtained after the application of a local search method (which is defined) is better than the initial solution, then VNS resumes the search from this solution. Otherwise, a solution is selected using a neighborhood of upper range, here is the neighborhood $\mathcal{V}_1$. Local search is applied to this solution, and so on. This method allows escaping from a local optimum if it is a local optimum for neighborhood for a range smaller than the max.

3.1 Initial Solution

In this work we used The Nearest Neighbor algorithm for which Solomon in [26] proposed a cost function to generate a initial solution. The nearest-neighbor heuristic start by finding the

Algorithm 1 Basic Steps of VNS

Initialization:choose a set of neighborhood structure $\mathcal{V}_k, k = 1, ..., k_{max}$, choose a distribution function of random variable, choose a local search method, an initial solution x and define the stopping criterion.
repeat
  $k \leftarrow 1$
  repeat
    Shaking: generate randomly a solution y from the k_{th} neighborhood $y \in \mathcal{V}_{k(x)}$; k_{th} neighborhood $y \in \mathcal{V}_{k(x)}$;
    Local search: apply local search methods with y initial solution to obtain a local optimum denoted y_1
    change:if the local optimum y_1 is better than x then change $(x \leftarrow y_1)$ and continue the search with $\mathcal{V}_1(K = 1)$, else $k \leftarrow k + 1$.
  until $k < k_{max}$
until the stopping criterion is checked

closest (in terms of a measure) unrouted customer to the depot. At every subsequent iteration, the heuristic searches for the customer closest to the last customer added to the route. This search is performed among all the customers. The mentioned cost function is defined as follows:

$$C_{ij} = \gamma_1 D_{ij} + \gamma_2 T_{ij} + \gamma_3 V_{ij} \tag{10}$$

$$\gamma_1 + \gamma_2 + \gamma_3 = 1, \gamma_1 \geq 0, \gamma_2 \geq 0, \gamma_3 \geq 0$$

where i is in the partial solution and j is not in the partial solution.with D_{ij} the Euclid distance from i to j. T_{ij} the Time difference between completing i and starting j and V_{ij} the Time remaining until last possible start of j following i.

3.2 Shaking

Generation of the whole neighborhood structure is the most important part of the VNS metaheuristic. In this paper five adjacent structures were designed. There is Intra-route and Inter-route. In the shaking phase customers who will be changed are selected randomly and the target route where we will insert the segment of customers is also selected randomly , thus, the algorithm stopped with the first changed solution. Table 1

1. $Two - opt$: Introduced by Croes in [7], the basic idea is replacing two arcs in a route by two others of the same route while inversing the direction of the route.

2. $Insert-Move$:The Insert-Move neighborhood consists to change the position of a segment route.

3. $Or - opt$ Introduced by Or in [20] for the traveling salesman problem . The basic idea is to relocate a set of consecutive vertices. This is achieved by the relocation of three arcs in the original route by another while maintaining the same orientation of route.

4. $Two - opt^*$: Introduced by Potvin in [21]. The basic idea of the Two-opt* is to combine two routes to insert the last customers of a given route after the first customer of another route while maintaining the orientation of the route.

5. *Exchange*: Introduced by Taillard in [21]. This operator enables the exchange of a value of γ customers between two different routes where in our case $\gamma \in [1, 5]$.

6. *Relocate*: Introduced by Lin [15]. The basic idea is to change the location of a customer in another route. In our case we exchanged the location of γ customers located consecutively in the original route with $\gamma \in [1, 5]$.

In the literature, the objective function is defined as hierarchy of objectives for the VRPTW, in which we minimize the number of vehicles at first then the total travel distance.We define a procedure to minimize the number of route based on relocate neighborhood as follows;

Algorithm 2 route minimizing procedure

```
for r1 =1...v do
  for r2 =1...v do
    if r1 ≠ r2 then
      for each customer k ∈ r1 do
        for each position in r2 do
          if the route r2 after insertion of k in position i is feasible then change the neigh-
          borhood solution s' of s
        end for
      end for
    end if
  end for
end for
```

K	Operator	Segment length
1	Insert-Move	1
2	Insert-Move	2
3	Or-Opt	1
4	Relocate	1
5	Relocate	2
6	Relocate	3
7	Exchange	1
8	Exchange	2
9	Exchange	3
10	Exchange	4
11	Exchange	5
12	Exchange	6

Table 1: set of neighborhood structure with $K_{max} = 12$

3.3 Local Search

To ameliorate the solution obtained through the shaking process, it will be subjected to a Local search LS method using five neighborhood with the best improvement strategy; $Two-opt$, *Relocate* with the relocation of one vertex, $Two-opt^*$, $Or-opt$, $swap(1)$, $swap(2)$ and $swap(3)$ Exchange with the exchange of one and two vertexes. The algorithm is described as follows;

Algorithm 3 basic steps of Local Search

Require: an initial solution x.
  $x_1 \leftarrow$ Best improvement on x using neighborhood $\mathcal{N}_1$
  $x_2 \leftarrow$ Best improvement on $x1$ using neighborhood $\mathcal{N}_2$
  $x_3 \leftarrow$ Best improvement on $x2$ using neighborhood $\mathcal{N}_3$
  $x_4 \leftarrow$ Best improvement on $x3$ using neighborhood $\mathcal{N}_4$
  $x_5 \leftarrow$ Best improvement on $x4$ using neighborhood $\mathcal{N}_5$
  if $f(x_4) < f(x)$ **then**
    $x \leftarrow x_5$
  end if

3.4 Acceptance Criterion

After shaking and local search phases have been completed, the resulting solution should be compared with the current one. Deciding whether it will be selected or not we adopted an acceptance criterion. In the basic VNS only solutions that improve the cost will be chosen, but we can easily find them in a local optimum. Thus, in most cases, therefore, it is essential to have a strategy to not accept Improvement solution under certain conditions. Instead, we put a system that is inspired by the simulated annealing (SA) [14]. More specifically, the solution obtained after the local search procedure is always accepted if the quality of the solution obtained is better and it is also accepted with a cost higher than the current solution with a probability $\exp(-(f(x') - f(x)/T))$ with $f(x)$ the cost of the solution. The temperature T decreases linearly in n/k steps during the search, n represents the total number of iterations. Thus, in each k iteration, T is reduced by an amount $T * n/k$. In the next section we will introduce our VNS-SCP approach for the VRPTW. The resulting method is called Skewed VNS which is an extension of VNS.

4 Set Covering ILP Model for the VRPTW

Let the index set of all feasible routes be $r = \{1, 2, ..., R\}$. Let c_r be the cost (e.g., length) of route r, and let $S_r \subseteq V$ denote those customers appearing in route r for all $r \in R$. Define

$$\alpha_{ir} = \begin{cases} 1 & \text{if route } r \text{ is in the optimal solution,} \\ 0 & \text{otherwise} \end{cases}$$

for each customer $i \in V$ and each route $r \in R$. Also, for every $r \in R$, let

$$y_{ir} = \begin{cases} 1 & \text{if customer } c_i \text{ is served by vehicle } r \\ 0 & \text{otherwise} \end{cases}$$

$$\min \quad z = \sum_{r \in R} c_r y_{ir} \tag{11}$$

$$\text{s.t.} \quad \sum_{r \in R} \alpha_{ir} y_{ir} \geq 1 \quad \forall i \in V \tag{12}$$

$$\sum_{r \in R} y_{ir} \leq K \tag{13}$$

$$y_{ir} \in \{0, 1\} \tag{14}$$

In the set-covering formulation of the VRPTW, the objective is to select a minimum-cost set of feasible routes such that each customer is included in some route. Constraints (12) ensure that each customer appear in at least one route, constraints (13) guarantee that the number of used routes is less or equal to K.

5 Hybridizing the VNS with the ILP Approach

The motivation for VNS-SCP is to exploit feasible routes of VNS solutions to obtain a high quality solution by applying the ILP solver on the set covering model. At a given time a set of routes is provided to the ILP model. The model will be solved by a branch-and-cut based generic ILP solver. As a conclusion, the efficacy of ILP solver depends on the quality of routes contained in the model i.e. the amount of routes, cost-effective routes and the diversity of routes. Not enough routes will prevent the ILP solver from ameliorating the cost of solutions; also, a large number of routes will prevent solving the model in short time. Consequently, selected routes to be added to the model must be carefully selected. At each iteration, if the current best solution is improved then it will be added to the pool. After each application of local search , if the solution is improved , this new solution is transferred to pool δ. Since the ILP used allows to customers to be served more than one time; redundant visits for customers are removed with the minimization of the total distance traveled. In contrast, a set partitioning model (derived by turning inequalities (12) into equalities) would yield only feasible solutions but at the same time exclude many potentially improving combinations. In case routes were altered during this transfer process, corresponding new columns are also added to the ILP model.

6 Experimental Results

The VNS algorithms have been implemented in Java, and executed on a 2.53 GHz intel Core i5 with 4GB RAM. For solving the ILP model in the hybrid VNS-SCP we apply ILOG CPLEX 12 MIP solver implemented with C++ with the same configuration machine. The VNS-SCP is tested by the classical set of 56 benchmark problems [27]. Those problems are composed of six different problem types (R1, C1, RC1, R2, C2, RC2), each type contains a number of problem and each problem contains 100 customers. The Euclidean distances between 2 customers are represented by the travel times between them. Sets C have the clustered customers whose time windows are generated based on a known solution. Sets R have the customers locations generated randomly over a square. Sets RC have a combination of randomly placed and clustered customers. Sets of type 1 have narrow time windows and a small capacity of the vehicle. Sets of type 2 have larger time windows and a larger capacity of the vehicle. Therefore, the solutions

Algorithm 4 basic steps of VNS-SCP

Initialization:choose a set of neighborhood structure $\mathcal{N}_k, k = 1, ..., k_{max}$ and $\mathcal{N}_k, k = 1, ..., k_{max}$ choose a distribution function of random variable, initial solution x and define the stopping criterion.
$\delta \leftarrow \emptyset$
repeat
  $K \leftarrow 1$
  repeat
    Shaking: generate randomly a solution y from the k_{th} neighborhood $y \in \mathcal{N}_{k(x)}$; k_{th} neighborhood $y \in \mathcal{N}_{k(x)}$
    Local search: apply Local Search procedure 3.3 with $\mathcal{N}_s$ and y initial solution to obtain a local optimum denoted y_1
    if the local optimum y_1 is better than x **then**
      (1) $(x \leftarrow y_1)$
      (2) continue the search with $\mathcal{N}_1(k = 1)$
      (3) add routes of solution x to δ // enrich the ILP model
    else
      $k \leftarrow k + 1$.
    end if
    $x \leftarrow$ apply ILP solver on δ;
  until $k < k_{max}$
until the stopping criterion is checked

to type 2 problems have very few routes and more customers per route. An initial temperature of 30 and we apply linear cooling every 100 iterations $T_0 = 10$, $(\gamma_1, \gamma_2, \gamma_3) = (0, 1, 0)$. For the VNS we set an iteration limit of either 1000, an initial temperature of 10 and a linear cooling every 100 iterations. For solving the ILP model in hybrid VNS-SCP we apply the general purpose MIP solver ILOG CPLEX 12. Each algorithm is run 20 times per instance and we report best results; e.i. best travel costs and the time in seconds were the best result is found.For the time execution of the ILP solver we can solve the model with 1500 route on a 2 seconds or less. In Table 2, the best results obtained by many authors in the literature [25]. The first column lists the authors. Columns R1, R2, C1, C2, RC1 and RC2 shows the average number of vehicles(NV), the average total distance(TD) and the average time in in minutes(Time) .It is observed that our approach, as a metaheuristic can produce good solutions even best known soltions was founded in a short time .

7 Conclusion

This paper proposed an algorithm based on the hybridization of exact an heuristic method to solve the Vehicle Routing Problem with Time Windows (VRPTW). The algorithm was presented in detail to allow a more deeply understanding of the metaheuristic operation. The use of the SCP formulation as a cooperative combination of VNS and a generic ILP solver applied to a set covering ILP formulation was proposed. The VNS provides the ILP with a set of feasible routes of the actual best solutions, and the ILP solver uses a parallel branch and cut to solve the formulation. The reached solution is added to VNS and so on, until a given time limit is reached. By applying the algorithms to 56 instances used as benchmarking in the literature, the approach was able to find a good or even the best known solutions with the

		R1	R2	C1	C2	RC1	RC2
[28]	NV	12.17	2.82	10.00	3.00	11.50	3.38
	TD	1209.35	980.27	828.38	589.86	1389.22	1117.44
	Time	229.6	337.2	-	-	187.7	193.3
[4]	NV	12.17	2.73	10.00	3.00	11.88	3.25
	TD	1209.35	980.27	828.38	589.86	1389.22	1117.44
	Time	-	-	-	-	-	-
[16]	NV	12.17	2.82	10.00	3.00	11.88	3.25
	TD	1249.55	1016.58	830.06	591.03	1412.87	1204.87
	Time	225.8	33.9	-	-	155.3	36.3
[11]	NV	12.00	2.73	10.00	3.00	11.50	3.38
	TD	1217.73	967.75	828.38	589.86	1382.42	1229.54
	Time	210.0	210.0	-	-	210.0	210.0
[12]	NV	11.92	2.73	10.00	3.00	11.63	3.25
	TD	1209.35	980.27	828.38	589.86	1389.22	1117.44
	Time	275.0	352.0	-	-	242.0	363.0
[23]	NV	12.08	3.00	10.00	3.00	11.63	3.38
	TD	1210.21	941.08	828.38	589.86	1382.78	1105.22
	Time	-	-	-	-	-	-
[6]	NV	12.08	2.73	10.00	3.00	11.50	3.25
	TD	1210.14	969.57	828.38	589.86	1389.78	1134.52
	Time	-	-	-	-	-	-
[2]	NV	12.00	2.73	10.00	3.00	11.50	3.25
	TD	1229.48	989.62	828.38	590.30	1394.26	1141.07
	Time	125.7	107.2	-	-	102.2	51.9
VNS-SCP	NV	12.17	3.00	10.00	3.00	11.75	3.25
	TD	1204.19	982.96	828.38	589.86	1366.46	1200.62
	Time	12.69	19.03	2.68	2.23	10.28	15.4579

Table 2: Comparison of solutions produced by the Proposed approach, with the results of different metaheuristics Proposed by other authors.The Time line depict the time consumption in minutes

minimum number of vehicles.

References

[1] Guilherme Bastos Alvarenga, Geraldo Robson Mateus, and G De Tomi. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6):1561–1584, 2007.

[2] Olli Bräysy. A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS Journal on Computing*, 15(4):347–368, 2003.

[3] Olli Bräysy and Michel Gendreau. Tabu search heuristics for the vehicle routing problem with time windows. *Top*, 10(2):211–237, 2002.

[4] Wen-Chyuan Chiang and Robert A Russell. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63(1):3–27, 1996.

[5] Jean-Francois Cordeau, Guy Desaulniers, Jacques Desrosiers, Marius M Solomon, and François Soumis. Vrp with time windows. *The vehicle routing problem*, 9:157–193, 2002.

[6] Jean-François Cordeau, Gilbert Laporte, and Anne Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, pages 928–936, 2001.

[7] GA Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, 1958.

[8] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[9] Agnieszka Debudaj-Grabysz and Zbigniew J Czech. Theoretical and practical issues of parallel simulated annealing. In *Parallel Processing and Applied Mathematics*, pages 189–198. Springer, 2008.

[10] Amine Dhahri, Kamel Zidi, and Khaled Ghedira. Vehicle routing problem with time windows under availability constraints. In *Advanced Logistics and Transport (ICALT), 2013 International Conference on*, pages 308–314. IEEE, 2013.

[11] Luca Maria Gambardella, Éric Taillard, and Giovanni Agazzi. Macs-vrptw: A multiple colony system for vehicle routing problems with time windows. In *New ideas in optimization*. Citeseer, 1999.

[12] Hermann Gehring and Jörg Homberger. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In *Proceedings of EUROGEN99*, volume 2, pages 57–64, 1999.

[13] Sin C Ho and Dag Haugland. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, 31(12):1947–1964, 2004.

[14] Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986, 1984.

[15] Shen Lin et al. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(10):2245–2269, 1965.

[16] Fuh-Hwa Franklin Liu and Sheng-Yuan Shen. A route-neighborhood-based metaheuristic for vehicle routing problem with time windows. *European Journal of Operational Research*, 118(3):485–504, 1999.

[17] Qi Ming-yao, Miao Li-xin, Zhang Le, and Xu Hua-yu. A new tabu search heuristic algorithm for the vehicle routing problem with time windows. In *Management Science and Engineering, 2008. ICMSE 2008. 15th Annual Conference Proceedings., International Conference on*, pages 1648–1653. IEEE, 2008.

[18] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.

[19] William P Nanry and J Wesley Barnes. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2):107–121, 2000.

[20] Ilhan Or. *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*. Xerox University Microfilms, 1976.

[21] Jean-Yves Potvin and Jean-Marc Rousseau. An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, 46(12):1433–1446, 1995.

[22] Yves Rochat and Éric D Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, 1(1):147–167, 1995.

[23] Louis-Martin Rousseau, Michel Gendreau, and Gilles Pesant. Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of heuristics*, 8(1):43–58, 2002.

[24] Paul Shaw. A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*, 1997.

[25] sintef. Benchmarks-Vehicle Routing and Travelling Salesperson Problems. Benchmark data sets for vehicle routing problems with time window. `http://www.sintef.no/Projectweb/TOP/VRPTW/`

Solomon-benchmark/100-customers/. [Online; Access in Octobre 2013].

[26] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[27] Marius M Solomon. Vrptw benchmark problems. *Excerpt from unpublished article, n. pag. http://www. cba. neu. edu/˜ msolomon/.(15 May 2000)*, 2003.

[28] Éric Taillard, Philippe Badeau, Michel Gendreau, François Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2):170–186, 1997.

[29] Kay Chen Tan, Loo Hay Lee, QL Zhu, and Ke Ou. Heuristic methods for vehicle routing problem with time windows. *Artificial intelligence in Engineering*, 15(3):281–295, 2001.

[30] Kenny Qili Zhu. A new genetic algorithm for vrptw. In *Proceedings of the International Conference on Artificial Intelligence*. Citeseer, 2000.

appendix

best known solutions:Problem RC103, Distance=1235.7256

Route 1: 1 3 7 8 9 47 5 46 6 4 2 1; 1 7 61 80 74 79 89 56 1;
Route 2: 1 97 55 43 45 44 41 37 36 38 82 1 ;
Route 3: 1 99 13 12 16 15 48 18 17 14 11 83 1;
Route 4: 1 81 95 94 72 73 39 40 42 62 71 101 69 1;
Route 5: 1 100 54 1;
Route 6: 1 57 64 77 90 19 49 26 78 1;
Route 7: 1 92 96 63 68 85 86 52 21 23 25 84 1;
Route 8: 1 91 53 60 10 88 98 76 59 1 ;
Route 9: 1 67 65 50 20 24 22 75 87 58 66 1 ;
Route 10: 1 93 51 34 33 31 29 27 28 30 32 35 1