The Journal of
Mathematics and Computer Science

Available online at
http://www.TJMCS.com

The Journal of Mathematics and Computer Science Vol .3 No.2 (2011) 192 - 201

# Solving Time Constrained Vehicle Routing Problem using Hybrid Genetic Algorithm

**Bhawna Minocha[1], Saswati Tripathi[2]**
[1]Amity School of Computer Sciences, Noida, India
[2]Indian Institute of Foreign Trade, Kolkata, India

Received: May 2011, Revised: June 2011
Online Publication: December 2011

## Abstract:

Vehicle Routing Problem with Time windows (VRPTW) is an example of scheduling in constrained environment. It is a well known NP hard combinatorial scheduling optimization problem in which minimum number of routes have to be determined to serve all the customers within their specified time windows. So far different analytic and heuristic approaches have been tried to solve such problems. In this paper we proposed algorithms which incorporate new local search techniques with genetic algorithm approach to solve VRPTW scheduling problems in various scenarios.

**Keywords**:
Genetic algorithm, heuristics based search techniques, vehicle routing problem with time windows, case-study

## 1. Introduction

Scheduling and routing problems have attracted considerable attention in recent years due to their wide applicability and importance in determining efficient distribution strategies to reduce operational cost in transportation logistics. VRPTW is an extension of the Vehicle Routing Problems (VRP) arising in transportation logistics that usually involve scheduling in constrained environment.

VRP may be described as follows: Given a set of vehicles with fixed and identical capacity located at common depot and a set of geographically scattered locations (cities, stores, schools, customers, warehouses etc.) which have varying demands, it is desired to determine minimum cost routes of the vehicles in such a way that each location is visited once and only once by one of the vehicles under the restriction that all routes start and ultimately end at the depot. The routes are designed in such a way that the total demands at locations enroute do not exceed the carrying capacity of the vehicle.

VRPTW is an extension of the VRP with the additional restriction that at each location there is a time window. The vehicle can visit the location in this specified time window only. The windows are of two types: Soft time windows and hard time windows. A soft time window can be violated at some cost whereas a hard time window has to be strictly adhered to. It does not accept a vehicle after the latest time specified in the time window. However, if a vehicle arrives at such a destination prior to earliest specified time of the window, it will have to wait till start of the time window. However this waiting penalizes the management either in the direct waiting cost or the increased number of vehicles to be used as because of such a waiting it will be able to service fewer destinations later even when capacity is available in the vehicle. Figure 1 shows a graphical model of VRPTW and its solution.

The objective of VRPTW is to service all the customers as per their requirement while minimizing the number of vehicles required as well as the total travel distance of all the vehicles used without violating capacity constraints of the vehicles and the customer's time window requirement such that each customer is visited in any way once and only once by one of the vehicles. All the routes are to start and ultimately end at the depot.

Routing and scheduling problems arise in a wide range of practical decision making situations. VRPTW arises in retail distribution, school bus/taxi scheduling, waste collection, courier/mail delivery/pickup, and airline/railway fleet routing etc. VRPTW is NP-hard. It has been extensively investigated in recent years using analytic optimization techniques, heuristics and meta-heuristics approaches.

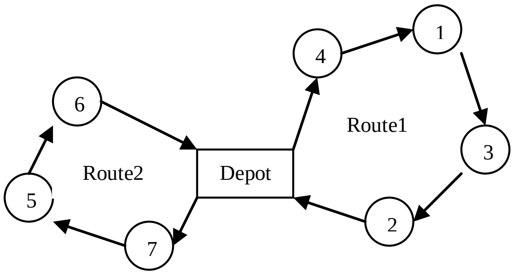

**Figure 1: Typical output for VRPTW**

The early work on VRPTW can be broadly divided into two categories: exact optimization and heuristic algorithms. Using exact optimization techniques, Kohl et al. [11], Larsen [12] and Chabrier [4] obtained significant improvements in Solomon's benchmark problem instances. Survey of the VRPTW literature by heuristics and meta-heuristics approaches has been given by Bräysy et al. [3] and Minocha et al. [14]. Cordeau et al. [5] and Rochat et al. [16] tried to solve these problems using tabu search whereas Gambardella et al. [7] considered ant colony optimization approach. Shaw [18] applied large neighborhood search (LNS). This was extended by Ropke et al. [17] as Adaptive-LNS approach to solve VRPTW problems. Homberger et al. [9] proposed parallelization of a two-phase metaheuristic technique for solving VRPTW. A complete survey of the VRPTW literature has been given by Cordeau et al. [6] which includes both the categories.

The Genetic Algorithm (GA) approach was proposed by Holland [8] in 1975. It is an adaptive heuristic search method that mimics evolution through natural selection. It works by combining selection, crossover and mutation operations of genes. The selection procedure drives the population toward a better solution while crossover uses genes of selected parents to produce new offsprings that form the next generation. Mutation is used to escape from local minima. The genetic algorithm approach has now become popular as it helps in finding reasonably good solutions for complex mathematical problems, NP-hard problems like VRP.

Blanton et al. [2] were the first to use GA approach to solve VRPTW. They hybridized GA approach with a greedy heuristic. A cluster-first, route-second method using genetic and local search optimization was used by Thangiah [22] and GENEROUS by Potvin et al. [15]. A multi-objective representation of VRPTW using pareto-ranking was used by Ombuki et al [13]. Others such as Berger et al. [1], Tan et al. [21] and many more also used GA for solving VRPTW problems.

In this study we consider an alternative approach based on the merger of genetic algorithm approach with new local search heuristics for solving VRPTW problems. In section 2 we formally describe the VRPTW problem. In section 3 we present our proposed hybrid of GA with heuristics based local search. Section 4 shows the results we obtained in solving some of the Solomon's benchmark problem by using proposed algorithm followed by conclusion in section 5.

## 2. Vehicle Routing Problem with Time Windows (VRPTW)

The VRPTW is represented by a set of uniform vehicles denoted by $V$ and directed graph $G$ such that $G = (N, A)$ is a connected directed graph with node set $N = C \cup \{0, n+1\}$ and arc set $A$, where $C$ stands for customers, each of which can be serviced only within a specified time interval. The set $A$ of arcs represents all possible connections between the nodes (including node representing depot). Nodes $0$ and $n+1$ represent the central depot, i.e. exiting and returning depot respectively. We associate a cost $c_{ij}$ and time $t_{ij}$ with each arc $(i, j)\ \varepsilon\ A$ of the routing network. Each vehicle has a capacity limit $q$, and each customer $i, i\ \varepsilon\ C$ has a varying demand $d_i$. Each customer, $i \quad \varepsilon\ C$ must be serviced within a pre-defined time window $[a_i, b_i]$. Vehicles arriving later than the latest arrival time i.e. after $b_i$ are penalized while those which reach the customer $v_i$ earlier than earliest arrival time i.e. before the $a_i$, incur waiting time $w_i$, until service is possible. Vehicles must also leave the depot within the depot time window $[a_0, b_0]$ and must return before or at time $b_{n+1}$.
The aim of the VRPTW is to service all the $C$ customers using the $V$ vehicles such that the following objectives are met and the following constraints are satisfied.

**Objectives:**

- Minimize the total number of vehicles used to service the customers.
- Minimize the distance traveled by the vehicles.

**Constraints:**

- Each customer is serviced only once.
- Time window constraint should be observed.
- Vehicle capacity constraint is observed.
- Each vehicle starts from node $0$ and ends at node $n+1$.

## 3. HYBRID GENETIC ALGORITHM

Although GA perform well in global search, but they usually take long time to converge to the global optimal solution. On contrary local searches (valid in a small region of search space) are quick in finding an optimal solution. Thus to improve the efficiency of GA we try to incorporate local searches with them. We call these as hybridized GA.

After building the initial population, all individuals are evaluated according to the fitness criteria. The evolution continues with tournament selection, where good individuals are selected for reproduction. Two best individuals are kept for next generation without going through genetic operations. Crossover and mutation are then applied to modify the selected individuals to form a new feasible generation. To further improve the individuals, local search heuristics are applied. We generate a random number *r* between 0 and 1, if *r* is less than 0.8; one of the local search algorithms is then executed, otherwise no local search algorithm executed. The Process is continued iteratively till the best solution does not change for a specified number of generations or till an overall specified number of generations have been performed. The best solution is taken as the desired optimal solution. The working of the proposed algorithm is summarized in the flowchart is shown in Figure 2.

## 3.1 Chromosome & Individual Representation

The representation of the GA chromosome in the present work is very simple. Each customer has an unique integer identifier *i*, where *i ε N*. An individual, which is a collection of chromosomes, represents a complete routing solution. Each chromosome represents a route, which is variable in length, contains a sequence of customers in the order in which they are visited by the vehicle. A different vehicle is needed to serve every chromosome of the individual. Every individual and every route must be feasible, in terms of capacity and time window constraints. The central depot is not considered in this representation, because all routes necessarily start and end in it. Figure 3 represents a complete routing solution for a problem instance with 25 customers, consisting of 3 routes which are served by 3 vehicles; genetically we call it as 3 chromosomes and complete solution as individual.

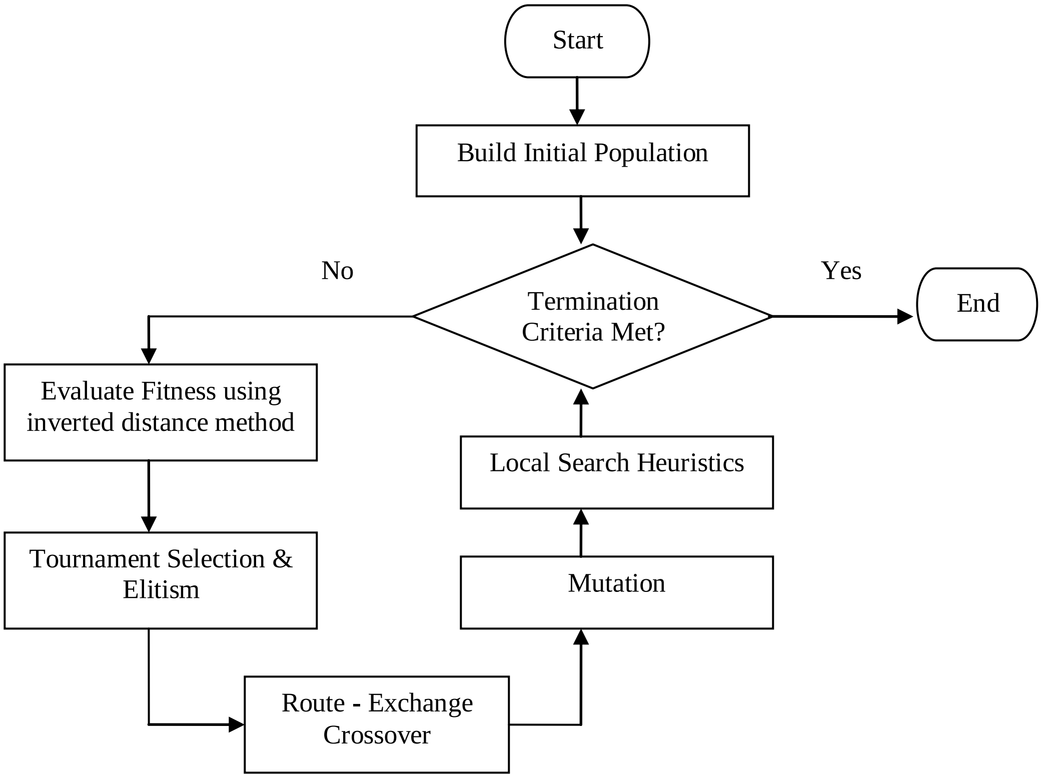
Start
Build Initial Population
No
Termination Criteria Met?
Yes
End
Evaluate Fitness using inverted distance method
Local Search Heuristics
Tournament Selection & Elitism
Mutation
Route - Exchange Crossover

**Figure 2: Flowchart of proposed Algorithm**

## 3.2 Initial Population

An initial population is built such that each solution is a feasible candidate solution i.e. every individual and every chromosome/route in the population satisfies time window and capacity constraints. We first generate a feasible solution using Push Forward Insertion Heuristics (PFIH) first introduced by Solomon [19]. This method has been frequently used in literature. Details of this method are available in Thangiah [22]. Rest of the solutions of initial population are generated by selecting the customers in a random manner and inserting them in an existing route, if one exists, otherwise a new route is created. Any customer that violates any constraint is deleted and a new route is added to serve the customer. This process is repeated until all the customers get served and a feasible initial population has been generated.

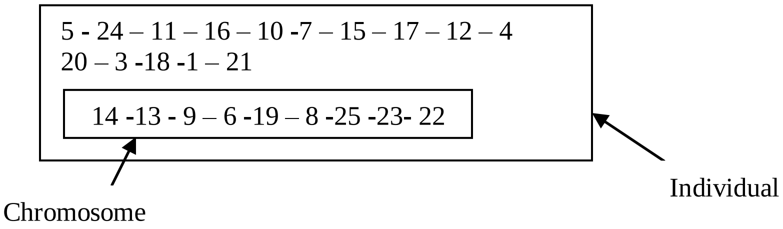

**Figure 3: An individual with 3 chromosomes / vehicles / routes. It represents a solution for a problem instance with 25 customers.**

## 3.3 Fitness

The fitness function measures the quality of the represented solutions. As soon as all the individuals have been created, they are ranked as per their fitness. It is commonly obtained from the objective function of the optimization problem but not necessarily coincide with it. In the study inverse of the traveled distance was used to calculate the fitness of the individuals.

## 3.4 Selection and Elitism

In selection, parents are selected for crossover. There are many methods proposed in the literature for this. In this study, an n-way tournament selection procedure has been used. Here n individuals are randomly selected and then the individual with highest fitness is declared as the winner. This process is repeated until the number of selected individuals equals to the number necessary for crossover. In this study, tournament size, i.e. n has been taken to be 3.

In the elitism process the good individuals are retained for reproduction. This ensures that the best solution obtained from the present population is copied unaltered in the next population. We replace the 4% worst individuals in the new population with the best 4% individuals of the parent population.

## 3.5 Reproduction

The classical single/double point crossover is not appropriate for scheduling problems like TSP or VRP because duplication and omission of vertices produce infeasible sequences in the offspring. Therefore in this study we have used route-exchange crossover. Once a pair of individuals is selected for crossover, efforts are made to exchange a route that has minimum number of nodes in each of the two individuals. To ensure that all individuals are feasible routing solutions after crossover, any duplication is deleted.

Mutation is necessary for inserting new characteristics that are not present in the current individuals. Without mutation the search gets limited to a very small area in the feasible region. In this study effort is made to transfer customers from route that has minimum number of nodes to other routes if possible to decrease the number of routes.

## 3.6 Local Search Heuristics

In this study we incorporate two new local search heuristics to search for better routing solutions of VRPTW. These searches are 'Replacing Next Neighbour' and 'Reinserting Random Customer'.

**i)** ***Replacing Next Neighbor (RNN):*** In this local search heuristics after having selected an individual randomly, a node say $C_j$ is randomly chosen on one of its routes and an effort is made to replace its next neighbor $C_{j+1}$ by some alternative acceptable node, say $C_k$ and then reinsert all the nodes from $C_{j+1}$ onwards of this route in other routes. Necessary modifications are carried to generate a new feasible solution.

**ii)** ***Reinserting Random Customer (RRC):*** This heuristic chooses a random customer from a randomly selected individual and tries to reinsert it in some other route. If possible we insert it and modify the existing solution else return the same individual.

## 4. Experimental Results & Comparison

In this section we summarize the results of the computational experiments performed by us using the proposed algorithms on a set of benchmark test problems selected from Solomon's set of problems.

Benchmark Problems: Solomon [19] generated a set of 56 problems which have been frequently used in literature to assess and compare the performance of algorithms developed for solving VRPTW problems. The problems vary in available fleet size, vehicle capacity, traveling time of vehicles and spatial and temporal distribution of customers to be served. Problems have been grouped in three categories namely R, C and RC. In problems of R category, the customers are uniformly distributed. Customers in C category problems are clustered either geographically or according to time windows. Category RC is a hybrid of problems of category R and C having an inter mix of features of problems of these two categories.

Problems in categories R, C and RC are further sub classified as R1, C1, RC1 and R2, C2, RC2. Problem sets R1, C1 and RC1 have a short scheduling horizon and allow only a few customers per route. In contrast, problems in sets R2, C2 and RC2 have a long scheduling horizon permitting many customers to be serviced by the same vehicle. Problems differ with respect to the width of the time windows sizes. Some have very tight time windows, while others have time windows which are hardly constraining. Larger problems have one hundred customers to be served. Smaller problems have been created out of these by considering only the first 25 or first 50 customers.

Each problem has been analyzed ten times using inverted distance as fitness criteria and two local search algorithms as discussed in previous section for generating new solutions from the existing ones. The two approaches which have been used to generate new solution from one of the existing ones are:

**Algo I: -** In this we randomly choose either RNN or RRC to generate a new feasible solution. For this we randomly generate a number *r*, between 0 and 1.

**Algo II: -** In this we try both RNN and RRC approaches. We accept the result of the approach which succeeds in generating a new feasible solution. If both succeed we accept the one which gives lesser objective value.

In our present study we have chosen from these sets: 15 problems of 25 customers; 10 problems of 50 customers and 5 problems of 100 customers. The best and the worst results (fitness criteria wise) are listed in Table 1 for 25 node problems. Table 2 shows results for 50 node problems and Table 3 for 100 node problems. Following parameters have been used: Population Size = 100; Generation size = 1000; Crossover rate = 0.80; Mutation rate = 0.20.

Table 1, 2 and 3 present a summary of our results and compare them with the best-known routing solutions available in literature. Bold numbers in Table 1, 2 and 3 indicate that the obtained solutions are the same as the best-known or there is an improvement on the currently best known solution. The proposed algorithm has been coded in C++ language and implemented on Intel(R) Core(TM) 2 Duo 2.0 GHz processor. Regarding the complexity of the algorithms it is of order $O(MN^2)$. Here $M$ is the number of objectives and $N$ is the total feasible solutions. This is due the fact that for each objective each feasible solution has to be evaluated and for ranking each of the $N$ solutions are compared with each other.

The proposed algorithms have produced new improved results in RC203-RC208 for 25 nodes with smaller number of vehicles but with slightly higher routing cost, as reported in Table 1. For RC201 and RC205, we obtained best-known result using Algo I. Similarly for 50 node problems, with Algo I and Algo II, C2 and R2 show some good solutions which need lesser vehicles. In 100 node problems, for R101, we obtained the best-known solution. In RC101, there are two reported solutions in the Table 3 that have better distance scores than the best known solution. Similarly for R105 in Table 3, we obtained one solution which is better than best-known. However, they need more vehicles than the best-known solutions. For rest of the cases, almost all the results are within (10%) of the best-known results. In some cases like R201, R202 and R203 there are two reported solutions in the Table 2 which have lesser number of vehicles than the best-known with higher routing cost.

| **Problem** | **Algo I** | | **Algo II** | | **Best Known [Ref.]** |
|---|---|---|---|---|---|
| | **Best** | **Worst** | **Best** | **Worst** | |
| C201 | 2/ 215.54 | 2/ 237.15 | 2/ 215.54 | 2/ 237.15 | **2/ 214.7 [11]** |
| R101 | 8/ 618.33 | 9/ 657.90 | 8/ 618.33 | 8/ 651.89 | **8/ 617.1 [11]** |
| R102 | 7/ 581.45 | 7/ 671.91 | 7/ 576.79 | 8/ 637.84 | **7/ 547.1 [11]** |
| R105 | 6/ 539.62 | 6/ 593.11 | 6/ 531.80 | 6/ 610.14 | **6/ 530.5 [11]** |
| R109 | 5/ 493.46 | 6/ 570.96 | 5/ 472.26 | 6/ 583.03 | **5/ 441.3 [11]** |
| RC105 | 4/ 477.04 | 5/ 552.47 | 4/ 458.28 | 5/ 503.63 | **4/ 411.3 [11]** |
| RC106 | 3/ 364.53 | 4/ 450.64 | 3/ 362.39 | 4/ 427.03 | **3/ 345.5 [11]** |
| RC201 | **3/ 360.2** | 3/ 486.80 | 3/ 411.53 | **2/ 596. 62** | **3/ 360.2 [12]** |
| RC202 | 3/ 373.50 | **2/ 508.93** | 3/ 378.63 | 3/ 469.80 | **3/ 338.0 [10]** |
| RC203 | **2/ 438.51** | 3/ 457.28 | **2/ 466.83** | 3/ 490.02 | **3/ 326.9 [4]** |
| RC204 | **2/ 421.26** | 2/ 491.01 | **2/ 446.57** | 2/ 490.50 | **3/ 299.7 [4]** |
| RC205 | **3/ 338.0** | 3/ 495.35 | **2/ 595.07** | 2/ 681.88 | **3/ 338.0 [12]** |
| RC206 | **2/ 501.01** | 2/ 549.85 | **2/ 499.83** | 2/ 538.61 | **3/ 324.0 [10]** |
| RC207 | **2/ 465.17** | 2/ 535.70 | **2/ 448.60** | 2/ 578.75 | **3/ 298.3 [10]** |
| RC208 | **1/ 429.53** | 1/ 461.87 | **1/ 382.56** | 1/ 429.66 | **2/ 269.1 [4]** |

**Table 1: Comparison of Best Known results with the results generated by Proposed Algorithms for Solomon's 25 customers set Problem**

| **Problem** | **Algo I** | | **Algo II** | | **Best Known [Ref.]** |
|---|---|---|---|---|---|
| | **Best** | **Worst** | **Best** | **Worst** | |
| C101 | 5/ 363.25 | 6/ 476.30 | 5/ 363.25 | 6/ 410.89 | **5/ 362.5 [11]** |
| C201 | **2/ 444. 96** | 3/ 489.65 | 3/ 408.73 | 3/ 478.16 | **3/ 360.2 [12]** |
| C205 | **2/ 721.91** | 2/ 835.18 | **2/ 818.15** | 2/ 887.38 | **3/ 360.2 [12]** |
| R101 | 12/ 1081.13 | 13/ 1134.52 | 12/ 1086.00 | 13/ 1138.14 | **12/ 1044 [11]** |
| R201 | **4/ 1185.35**<br>**5/ 1107.42** | 5/ 1185.06 | **4/1093.36** | 5/ 1138.92 | **6/ 791.9 [10]** |
| R202 | **3/ 1218.46**<br>**4/ 1107.71** | 4/ 1230.82 | **4/ 1050.81** | 4/ 1242.94 | **5/ 698.5 [10]** |
| R203 | **3/ 1172.54**<br>**4/ 1067.59** | 4/ 1162.75 | **4/ 1069.20** | 4/ 1161.61 | **5/ 605.3 [4]** |
| R206 | **3/ 969.97** | 3/ 1160.83 | **3/1065.67** | 3/ 1161.31 | **4/ 632.4 [4]** |
| R209 | **3/ 1018.91** | 3/1188.28 | **3/ 1125.86** | 4/ 1068.51 | **4/ 600.6 [10]** |
| RC101 | 9/ 1033.05 | 10/ 1105.99 | 9/ 1015.56 | 10/ 1101.67 | **8/ 944 [11]** |

**Table 2: Comparison of Best Known results with the results generated by Proposed Algorithms for Solomon's 50 customers set Problem**

| **Problem** | **Algo I** | | **Algo II** | | **Best Known [Ref.]** |
|---|---|---|---|---|---|
| | **Best** | **Worst** | **Best** | **Worst** | |
| C101 | **10/ 828.94** | 13/ 1175.81 | **10/ 828.94** | 13/ 1008.71 | **10/ 828.94 [16]** |
| R101 | 20/ 1751.74 | 21/ 1833.90 | 20/ 1755.39 | 21/ 1829.45 | **19/ 1650.8 [16]** |
| R102 | 18/ 1516.67 | 20/ 1792.63 | 18/ 1519.92 | 20/ 1747.09 | **17/ 1486.12 [16]** |
| R105 | 15/**1375.55** | 18/ 1813.68 | 15/1385.63 | 17/ 1629.78 | **14/ 1377.11 [18]** |
| RC101 | 15/ **1628.08** | 18/ 2008.90 | 15/ **1640.98** | 18/ 1882.90 | **14/ 1696.94 [20]** |

**Table 3: Comparison of Best Known results with the results generated by Proposed Algorithms for Solomon's 100 customers set Problem**

## 5. Concluding Remarks

Vehicle routing problem with time windows involves the optimization of routes for multiple vehicles so as to meet all constraints and to minimize the number of vehicles needed and total distance traveled. The proposed hybrid algorithm incorporates GA approach with new heuristics in local search. Performance of proposed algorithms is comparable to those available in literature and in some cases even better in terms of number of vehicles which means less fuel, manpower and vehicle maintenance cost with more distance to travel. As for future work, it may be interesting to test proposed algorithm on some application of VRPTW.

## References

[1] Berger J. and Barkaoui, M., "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows" Computer Operation Research, 2004, 31: 2037-2053.

[2] Blanton J.L. and Wainwright R.L., "Multiple vehicle routing with time and capacity constraints using genetic algorithms". Proceedings of the 5th International Conference on Genetic Algorithms, June 1993, USA, 452-459.

[3] Bräysy O. and Gendreau M., "Vehicle routing problem with time windows, Part II: Metaheuristics ", Transportation Science, 39:119–139, 2005

[4] Chabrier A., "Vehicle Routing Problem with Elementary Shortest Path based Column Generation." Computers and Operations Research, 33(10), 2972-2990, 2006.

[5] Cordeau, J.F., Laporte, G., Mercier, A., "A Unified Tabu Search for Vehicle Routing Problem with Time Windows", Journal of the Operational Research Society 52: 928-936, 2001

[6] Cordeau, J.F., G. Desaulniers, J. Desrosiers, M.M. Solomon and F. Soumis, "The VRP with Time Windows" , In: The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, Toth, P. and D. Vigo (Eds.), Philadelphia, USA, pp: 157-193.

[7] Gambardella, L. M., Taillard, E., Agazzi, G.," MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows", New Ideas in Optimization, McGraw-Hill, 1999

[8] Holland, J.H., "Adaptation in Natural and Artificial System". Ann Arbor, Michigan: The University of Michigan Press, 1975.

[9] Hömberger, J., Gehring, H., "A Two-Phase Hybrid Metaheuristic for the Vehicle Routing Problem with Time Windows", European Journal of Operations Research, 162(1): 220-238, 2005.

[10] Kallehauge B., Larsen J., and Madsen O.B.G. "Lagrangean duality and non-differentiable optimization applied on routing with time windows ." Computers and Operations Research, 33(5): 1464-1487,2006.

[11] Kohl N., Desrosiers J., Madsen O. B. G., Solomon M.M. and Soumis F., "2-Path Cuts for the Vehicle Routing Problem with Time Windows," Transportation Science, Vol. 33 (1) :101-116 ,1999.

[12] Larsen J. "Parallelization of the vehicle routing problem with time windows." Ph.D. Thesis IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark,Lyngby, Denmark, 1999.

[13] Ombuki, B., Ross B.J., and Hansher F., "Multi-objective Genetic Algorithm for Vehicle Routing Problem with Time Windows", Applied Intelligence, 24(1):17-30, 2006.

[14] Minocha B. and Tripathi S., "Vehicle Routing Problem with Time Windows: An Evolutionary Algorithmic Approach", Algorithmic Operations Research, 1(2): 1-15, 2006

[15] Potvin, J., Bengio. S. "The Vehicle Routing Problem with Time Windows part II: Genetic Search. INFORMS Journal on Computing, 8(2):165-172, 1996.

[16] Rochat, Y., Taillard, E., "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing." Journal of Heuristics 1: 147–167, 1995.

[17] Ropke, S., Pisinger, D., "A General Heuristics for Vehicle Routing Problems", Computers & Operations Research, 34(8): 2403-2435, 2007

[18] Shaw, P., "Using Constraint Programming and Local Search Methods to solve Vehicle Routing Problems", Principles and Practice of Constraint Programming, Springer-Verlag , 417-431, 1998

[19] Solomon, M.M.," Algorithms for Vehicle Routing and Scheduling Problems with Time Window Constraints", Operations Research 35(2):254-265, 1987

[20] Taillard, E.D., Badeau, P., Gendreau, M., Guertin, F., Potvin, J., "A Tabu Search Heuristics for the Vehicle Routing Problem with Soft Time Windows". Transportation Science 31:170-186, 1997.

[21] Tan, K. C., Lee, L. H., Ou, K., "Hybrid Genetic Algorithms in Solving Vehicle Routing Problems with Time Window Constraint", Asia-Pacific Journal of Operation Research 18:121-130, 2001

[22] Thangiah, S., "Vehicle Routing with Time Windows using Genetic Algorithms", In Application Handbook of Genetic Algorithms: New Frontiers, 2: 253-277, CRC Press, Boca Raton, 1995