



# Journal of Applied Sciences

ISSN 1812-5654

**science**  
alert

**ANSI***net*  
an open access publisher  
<http://ansinet.com>

## Hybrid Genetic Algorithm for Vehicle Routing and Scheduling Problem

K. Ghoseiri and S.F. Ghannadpour

Department of Railway Engineering, Iran University of Science and Technology, 16846-13114, Tehran, Iran

**Abstract:** This study aims to solve Vehicle Routing Problem with Time Windows (VRPTW), which has received considerable attention in recent years, using hybrid genetic algorithm. Vehicle Routing Problem with Time Windows is an extension of the well-known Vehicle Routing Problem (VRP) and involves a fleet of vehicles set-off from a depot to serve a number of customers at different geographic locations with various demands within specific time windows before returning to the depot eventually. To solve this problem, this study suggests a hybrid genetic algorithm combined with Push Forward Insertion Heuristic (PFIH) to make an initial solution and  $\lambda$ -interchange mechanism to neighborhood search and improving method. The proposed genetic algorithm uses an integer representation in which a string of customer identifiers represents the sequence of deliveries covered by each of the vehicles. Part of initial population is initialized using Push Forward Insertion Heuristic (PFIH) and part is initialized randomly. A  $\lambda$ -interchange mechanism interchanges the customers between routes and generates neighborhood solution. At the end, in order to prove the validity of the suggested model, fourteen instances of Solomon's 56 benchmark problems-selected randomly- are solved and compared with the other meta-heuristic methods. The results indicate the good quality of the method.

**Key words:** Vehicle routing problem with time windows, genetic algorithm, push forward insertion heuristic,  $\lambda$ -interchange mechanism

### INTRODUCTION

Vehicle Routing Problem with Time Windows (VRPTW) is a kind of important variant of Vehicle Routing Problem (VRP) with adding time windows constraints to the model. VRP is one of the most attractive topics in operation research and deals with the determination of the least cost routes from a central depot to a set of geographically dispersed customers. Vehicle Routing Problems (VRPs) are well known combinatorial optimization problems arising in transportation logistic that usually involve scheduling in constrained environments. In transportation management, there is requirement to provide goods and/or service from a supply point to various geographically dispersed points with significant economic implications. Since VRP has many various applications, many researchers have attempted to develop solution approaches and techniques to solve this problem (Pisinger and Ropke, 2005; Laporte and Semet, 2001; Bräysy and Gendreau, 2005).

In VRPTW, shown in Fig. 1, a set of vehicles with limited capacity is to be routed from a central depot to a set of geographically dispersed customers with known demands and predefined time windows to minimize total traveling distance and capacity and time windows constraints are not violated.

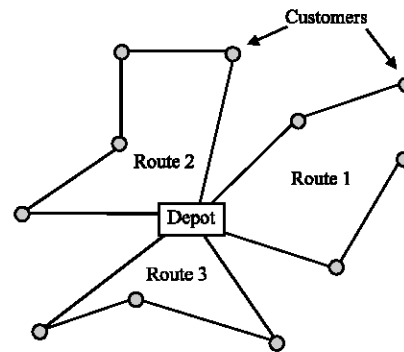


Fig. 1: Typical output for VRPTW

Due to its inherent complexities and usefulness in real life, the VRPTW continues to draw researchers' attention and has become a well-known problem in network optimization; so many researchers have developed different solution approaches based on the following methods:

- Exact algorithms
- Classical heuristics developed mostly between 1960 and 1990
- Metahuristics whose growth has occurred nearly in 20 years ago

In terms of exact algorithms, Achutan *et al.* (2003) presented modern branch and cut techniques for routing problems. There are also other exact approaches proposed in the literature for the VRPTW, however no algorithm has been developed to date that can solve the optimality of all VRPTW with 100 customers or more. It should be noted that exact methods are more efficient in situations where the solution space is restricted by narrow time windows; since there are fewer combinations of customers to define feasible routes (Gambardella *et al.*, 1999). So many researchers have investigated the VRPTW using Classical heuristics and Meta-heuristics approaches. Classical heuristics perform a relatively limited exploration of the search space and typically produce good quality solutions within modest computing times. These methods can be broadly classified into three categories: (1) Constructive heuristic, (2) Two-phases heuristics and (3) Improvement methods. Constructive heuristics gradually build a feasible solution while keeping an eye on solution cost, but do not contain an improvement phase (Petch and Salhi, 2003). In two-phase heuristics the problem is decomposed into its two natural components: clustering of vertices into feasible routes and actual route construction, with possible feedback loops between the two stages. Two-phase heuristics will be divided in two classes: cluster-first, route-second methods and route-first, cluster-second methods. In the first case, vertices are first organized into feasible clusters and a vehicle route is constructed for each of them. In the second case, a tour is first built on all vertices and is then segmented into feasible vehicle routes (Wren, 1971). Finally, improvement methods attempt to upgrade any feasible solution by performing a sequence of edge or vertex exchanges within or between vehicle routes. In metaheuristics, the emphasis is on performing a deep exploration of the most promising regions of the solution space. The quality of solution produced by these methods is much higher than that obtained by classical heuristics, but the price to pay is increased computing time. In a major departure from classical approaches, metaheuristics allow deteriorating and even infeasible intermediary solutions in the course of the search process. According to the Fig. 2, designing the methods based on heuristics and metaheuristics causes the new generation of methods which can produce results that are effective among the solution quality and the computation time.

Various heuristic methods are found in literature for VRPTW (Tan *et al.*, 2001; Thangiah, 1999). In this area, Czech and Czarnas (2002) solved VRPTW with simulated annealing, Rochat and Taillard (1995) and Bräysy and Gendreau (2002) solved VRPTW with tabu search and Tan *et al.* (2007) applied multiple ant colony

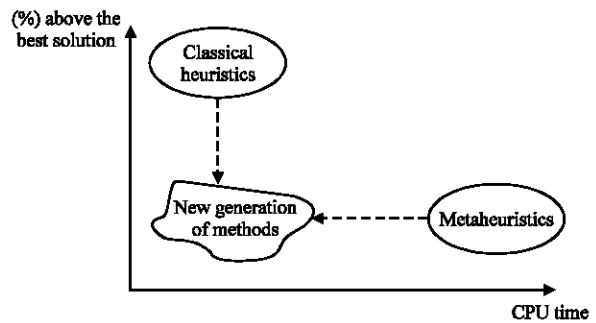


Fig. 2: The difference between classical heuristics and metaheuristics

system for VRPTW. There are some other efficient studies that use genetic algorithm for VRPTW (Ombuki *et al.*, 2006; Berger and Barkaoui, 2003; Tan *et al.*, 2006; Alvarenga *et al.*, 2007). In applying genetic algorithm, Blanton and Wainwright (1993) presented two new crossover operators, Merge Cross#1 and Merge Cross#2 and showed that the new operators are superior to traditional crossover operators. A cluster-first, route-second method using genetic algorithm and local search optimization process was done by Thangiah (1999). Comparative studies of the performance of genetic algorithm, tabu search and simulated annealing for the VRPTW are given (Tan *et al.*, 2001; Thangiah, 1999). Other heuristics that have been applied to the VRPTW include constraint programming and local search (De Backer *et al.*, 2000; Shaw, 1998). Other very good techniques and applications of VRPTW can be found in (Li *et al.*, 2005; Kim *et al.*, 2006; Tan *et al.*, 2007; Cerda and Dondo, 2007; Irnich *et al.*, 2006; Crevier *et al.*, 2007).

So, this study tries to design efficient methods based on heuristics and metaheuristics for solving the VRPTW. To solve this problem, this study suggests a hybrid genetic algorithm (with special operators) that is combined with Push Forward Insertion Heuristic (PFIH) to make an initial solution and  $\lambda$ -interchange mechanism to neighborhood search and improving method. The proposed genetic algorithm uses an integer representation in which a string of customer identifiers represents the sequence of deliveries covered by each of the vehicles. Part of initial population (50%) is initialized using Push Forward Insertion Heuristic (PFIH) and part is initialized randomly. A  $\lambda$ -interchange mechanism interchanges customers between routes and generates neighborhood solution. Finally, the suggestive algorithm is applied to solve some of benchmark Solomon's (1987) 56 VRPTW 100-customer instances that are selected randomly and the results are compared with the other meta-heuristic methods produced by Tan *et al.* (2001) and Thangiah (1999).

**MODEL DESCRIPTION**

The Vehicle Routing Problem with Time Windows (VRPTW) is given by a special node called the depot, a set of customer  $C$  to be visited and a directed network connecting the depot and the customers. Also homogeneous fleet of vehicles is available. They are located at the depot, so they must leave from and return to the central depot. It is assumed that there is no limitation on the number of vehicles that can be used, but in order to facilitate the model formulation it is denoted by  $K$  the maximum possible size of the fleet. The actual number of vehicles will be found after solving the model whose number is equal to the number of routes in the traffic network. Let us assume there are  $N+1$  customers,  $C = \{0, 1, 2, \dots, N\}$  and for simplicity, the depot is assumed as customer 0. Each arc in the network corresponds to a connection between two arcs. A route is defined as starting from depot, going through a number of customers and ending at the depot. A distance  $d_{ij}$  and travel time  $t_{ij}$  are associated with each arc of the network. The travel time  $t_{ij}$  may include service time at customer  $i$ . Every customer in the network must be visited only once by one of the vehicles. Since each vehicle has a limited capacity  $q_k (k = \{1, \dots, k\})$  and each customer has a varying demand  $m_i$ ,  $q_k$  must be greater than or equal to the summation of all demands on the route traveled by that vehicle  $k$ . On the other hand, any customer  $i$  must be serviced within a pre-defined time interval  $[e_i, l_i]$ , limited by an earliest ( $e_i$ ) and latest arrival time ( $l_i$ ). Vehicle arriving later than the latest arrival time are penalized while those arriving earlier than the earliest arrival time incur waiting. Figure 3 shows the time window of customer  $i$ . Assuming that waiting time is permitted at no cost and  $e_0 = l_0 = 0$  that is, all routes start at time 0. Vehicles are also supposed to complete their individual routes within the total route time which is essentially the time window of the depot.

**GENETIC ALGORITHM FOR VRPTW**

This study suggests an efficient method of solution for VRPTW such that the objective is met and the constraints are satisfied. The algorithm that is adapted to use in this study is Genetic Algorithm (GA), a class of adaptive heuristics based on the drawing concept of evaluation-survival of the fitness, developed by Holland (1992) at the University of Michigan. A GA starts with a set of chromosomes referred to as initial population. Each chromosome represents a solution to the problem and the initial population is either randomly generated (in which case it would take longer for the algorithm to converge to the solution) or generated using some from heuristic (in which case the population is already closer to the solution and would hence take less time to converge).

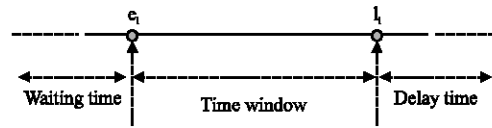


Fig. 3: Time window of customer (i)

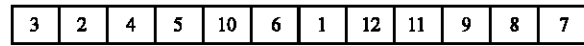


Fig. 4: Chromosome representation

This study uses some heuristics to generate initial population that will be explained later. A selection mechanism will then be used to select the prospective parents based on their fitness computed by evaluation function. The selected parent chromosomes will then be recombined via the crossover operator to create potential new population. The next step will be to mutate a small number of newly obtained chromosomes, in order to introduce a level of randomness that will preserve the GA from converging to a local optimum. The GA will then reiterate through this process until a pre-defined number of generations have been produced, or until there was no improvement in the population, which means that the GA has been found an optimal solution, or until a pre-defined level of fitness has been reached.

**Chromosome representation:** A solution to the problem is represented by an integer string of length  $N$ , where,  $N$  is the number of customers which need to be served. All routes are encoded together, with no special route termination characters in between and chromosomes are decoded back into routes based on capacity of each vehicle and maximum allowable operating time. For instance.

Figure 4 shows the chromosome representation for following routes with three vehicles and 12 customers:

- Route No. 1: 0-3-2-4-5-0
- Route No. 2: 0-10-6-1-12-11-0
- Route No. 3: 0-9-8-7-0

**Initial population:** An initial population is built such that each individual must at least be a feasible candidate solution, i.e., every route in the initial population must be feasible. In this study part of population is initialized using heuristics (50%) and part is initialized randomly. A fast and simple heuristic procedure to distribute all customers in the vehicles, if used to obtain the part of first individual generation, can reduce significantly the GA time necessary to reach the reasonable local minima. Because of this, the heuristic method proposed by Solomon (1987), called Push Forward Insertion Heuristic

(PFIH), has been frequently used by many researchers with this purpose. Detailed description of the PFIH method described by Thangiah (1999). In PFIH method, the relation (1) defines the first customer in each new route. Once the first customer is selected for the current route, the heuristic selected from the set of unrouted customers the customer which minimizes the total insertion cost between every edge in the current route without violating the time and capacity constraints.

$$c_i = -\alpha d_{0i} + \beta b_i + \gamma((p/360)d_{0i}) \quad (1)$$

In Eq. 1,  $\alpha$  is the 0.7 (empirically calculated by Solomon (1987)),  $\beta$  the 0.1 (empirically calculated by Solomon (1987)),  $\gamma$  the 0.2 (empirically calculated by Solomon (1987)),  $d_{0i}$  the distance from customer  $i$  to the central depot;  $b_i$  the upper time and  $P_i$  is the polar coordinate angle of the customer  $i$ . After the initial feasible solution ( $S_0$ ) is formed using PFIH, by letting it and its feasible random neighbors  $\forall S \in N_\lambda(S_0)$  using  $\lambda$ -interchange that is described later, a portion of starting population is completed. The rest of the population is generated totally on random basis and starts by inserting customers one by one into an empty route in a random order. Any customer that violates any constraints is deleted from current route. The route is then accepted as part of the solution. A new empty route is added to serve the deleted and other remaining customers. This process continues until all customers are routed and a feasible initial population is built. The reason for having this mixed population is that a population of members entirely from the same neighborhood cannot go too far from there and hence give up the opportunity to explore other regions.

This study uses a  $\lambda$ -interchange mechanism that moves customers between routes to generate neighborhood solution for the VRPTW. Given a feasible solution for the VRPTW represented by:

$$S = \{R_1, \dots, R_p, \dots, R_q, \dots, R_k\}$$

where,  $R_p$  is a set of customer serviced by a vehicle route  $p$ . A  $\lambda$ -interchange between a pair of routes  $R_p$  and  $R_q$  is a replacement of subset  $S_1 \subseteq R_p$  of size  $|S_1| \leq \lambda$  by another subset  $S_2 \subseteq R_q$  of size  $|S_2| \leq \lambda$ , to get the new route sets  $R'_p, R'_q$  and a new neighboring solution:

$$S' = \{R_1, \dots, R'_p, \dots, R'_q, \dots, R_k\}$$

Where:

$$R'_p = (R_p - S_1) \cup S_2, R'_q = (R_q - S_2) \cup S_1 \quad (2)$$

Figure 5 shows the instance of operator Eq. (2, 1) on two routes. The neighboring  $N_\lambda(S)$  of a given solution

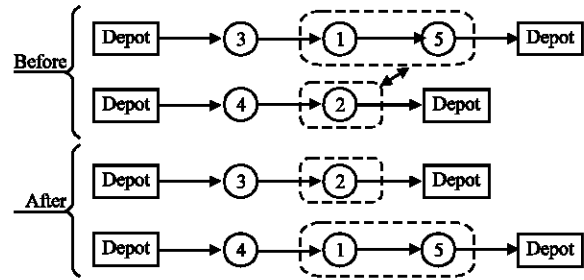


Fig. 5: Operator (2, 1) in 2-interchange mechanism

$S$  is the set of all neighbors  $\{S'\}$  generated by the  $\lambda$ -interchange method for a given  $\lambda$ . In one version of the algorithm called GB (Global Best), the whole neighborhood is explored and the best move is selected. In the other version, FB (first best), the first admissible improving move is selected if one exists; otherwise the best admissible move is implemented.

**Selection:** This study uses a ranking-based selection method applied by Correa *et al.* (2001), given by the following formula:

$$\text{Select}(R) = \left\{ r_j \in R / j = P - \left\lceil \frac{-1 + \sqrt{1 + 4 \cdot \text{rnd}(P^2 + P)}}{2} \right\rceil \right\} \quad (3)$$

where,  $R$  is a list  $R = \{r_1, r_2, \dots, r_P\}$ , with  $P$  individuals sorted in increasing order by fitness value,  $\text{rnd}$  is a uniformly-distributed random number between (0, 1) and the symbol  $\lceil b \rceil$  denotes the greatest integer smaller than or equal to  $b$ . Formula (3) returns the position in the list  $R$  of the individual to be selected. The formula is biased to favor the selection of individuals in early positions of the list-i.e., the best (smallest fitness) individuals.

**Crossover:** The classical crossover (one-point crossover,  $n$ -point crossover ...) is not appropriate for sequencing problem, like the TSP or the VRP. Use of them may cause the offspring do not have a valid sequence, due to duplication and omission of vertices. So, this study uses the Heuristic and Merge crossover for recombination phase that are applied earlier by Tan *et al.* (2001) and Thangiah (1999).

**Heuristic crossover:** This operator deals with distances between nodes: for example, according to Fig. 6, a random cut was made on two chromosomes and we will compare the genes immediately after the cut.

Let us say that  $B$  is to be the first gene in the child. Gene  $G$  has to be swapped with  $B$  in parent 2 to avoid subsequent repetition. Now we compare the distance

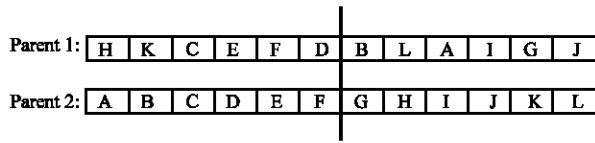


Fig. 6: Heuristic crossover

between B and the first two subsequent genes, L and H and choose the one which is geographically closer to B. Once again, in parent 2, the gene which was chosen with the remaining one is swapped to avoid duplication. This process is continued until a new chromosome of length N is formed. It must be mentioned that, just one child is produced from two parents by heuristic crossover. So for producing another child from them, the merge crossover operator is used and operated on two parents.

**Merge crossover:** This operator operates on the basis of time precedence, defined by the time window corresponding to each node. Similarly, the first gene is chosen randomly and the following genes will be the one whose time window comes earlier.

The probability that a pair of selected parents will mate is called the probability of crossover. When a couple of parent chromosomes is determined as non mating, they will be copied verbatim into the next generation. In this study, the probability of crossover is set to 80%.

**Sequenced Based Mutation (SBM):** In this operator, first, a link is randomly selected and removed from children solution that are produced after crossover phase from parents. Then, the customers that are serviced before the break point on the route of child-solution1 are linked to the customers that are serviced after the break point on the route of child-solution 2 (c.f. the black nodes in Fig. 7). Finally, the new route replaces the old one in child-solution 1. A second new chromosome can be created by inverting the role of the children. In a feasible solution, customers with early time windows are typically scheduled at the beginning of a route. Conversely, customers with late time windows are typically scheduled at the end of the route. Hence, by linking the first customers on a route of child-solution 1 to the last customers on a route of child-solution 2, the time window constraints are likely to be satisfied. Figure 7 shows the applying of SBM operator on the child-solution 1.

Unfortunately, the new solution is rarely valid, because some customers are duplicated or unrouted in the process. For example, in Fig. 7, two customers are now located on two different routes and two other customers are unrouted. Accordingly, a repair operator is applied to the new chromosome to generate a new feasible solution.

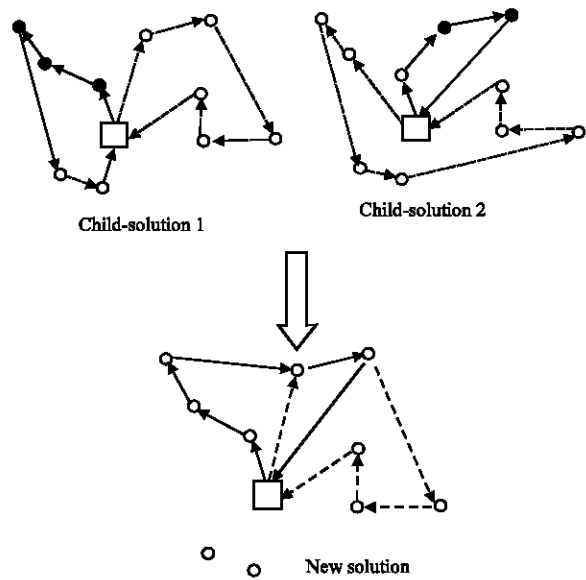


Fig. 7: Applying SBM operator on child-solution 1

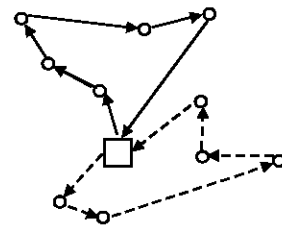


Fig. 8: Applying repair operator on new solution at Fig. 7

This operator deals with the infeasible solution in the following way:

- If a customer appears twice in the new route, one of the two copies is removed from the route. If a customer appears once in the new route and once in an old route, the customer is removed from the old route
- If a customer is unrouted, then this customer is inserted at the feasible insertion place that minimizes the additional cost and satisfies capacity and time window constraints. Obviously, there is no guarantee that there is a feasible insertion place for each one of them. If this situation occurs, the new solution is discarded and an old child-solution will be restored

Figure 8 shows the applying repair operator on new solution at Fig. 7.

**Acceptance:** In this study the offspring that produced by crossover and mutation is inserted into the population

only if they have a better (smaller) fitness than the worst individual of the current population and don't cause repetition in population.

**Improvement phase:** Improvement phase is a scheme for randomly selecting a portion (10%) of the population, decoding the chromosomes into solutions and then improving those solutions by a few iterations of removal and reinsertion. In this phase 1-interchange (FB) and 2-interchange (GB) are used to improve the selecting chromosomes.

**EXPERIMENTAL RESULTS AND COMPARISON**

This study describes computational experiments carried out to investigate the performance of the proposed GA. The algorithm was coded in MATLAB 7 and run on a PC with 1.6 GHz CPU and 512 MB memory. Here, fourteen instances of Solomon's 56 benchmark problems (Solomon, 1987)-that are selected randomly- have been solved and are compared with the other meta-heuristic methods that are reported by Tan *et al.* (2001) and Thangiah (1999).

**Solomon's 56 benchmark problems:** The Solomon's problems consist of 56 data sets, which have been extensively used for benchmarking different heuristics in literature over the years. The problems vary in fleet size, vehicle capacity, traveling time of vehicles, spatial and temporal distribution of customers. In addition, the time windows allocated for every customer and the percentage of customers with tight time-windows constraint also vary for different test cases. The customers' details are given in the sequence of customer index, location in x and y coordinates, the demand for load, the ready time, due date and the service time required. All the test problems consist of 100 customers, which are generally adopted as

the problem size for performance comparisons in VRPTW. The traveling time between customers is equal to the corresponding Euclidean distance. Solomon's data is clustered into six classes; C1, C2, R1, R2, RC1 and RC2. Problems in the C category mean the problem is clustered; that is, customers are clustered either geographically or according to time windows. Problems in category R mean that the customer locations are uniformly distributed whereas those in category RC imply hybrid problems with mixed characteristics from both C and R. Furthermore, for C1, R1 and RC1 problem sets, the time window is narrow for the depot, hence only a few customers can be served by one vehicle. Conversely, the remaining problem sets have wider time windows hence many customers can be served by main vehicles.

**Experimental results:** Here, fourteen instances of Solomon's 56 benchmark problems (Solomon, 1987)-selected randomly-have been solved and compared with the other meta-heuristic methods that are reported by Tan *et al.* (2001) and Thangiah (1999). Table 1 presents a summary of presented results and compares them with the best known solutions that are reported in the literature (Ombuki *et al.*, 2006) and with the results that are produced by other popular methods reported by Tan *et al.* (2001) and Thangiah (1999). Distance costs are measured by average Euclidian distance. The column labeled Best Known gives the best known published solutions; column present approach gives the best solution produced in 5 runs and column other methods gives the results reported by Tan *et al.* (2001) and Thangiah (1999) by other heuristics. Bolded numbers in Table 1 indicate that the obtained solutions are the same as the best known or indicate an improvement on the best currently known results in the literature.

Table 1: Summary of results and comparisons with the best solutions

Problem No.	Best known (Ombuki <i>et al.</i> , 2006)		Other methods (Tan <i>et al.</i> , 2001; Thangiah, 1999)			Present approach		
	Distance cost	Vehicle No.	2-Int	Simulated annealing (SA)	Tabu search (TS)	Genetic algorithm (GA)	Distance cost	Vehicle No.
C101	828.94	10	828.94	828.94	828.940	828.94	828.94	10
C102	828.94	10	923.37	923.37	901.520	868.80	828.94	10
C106	828.94	10	1052.07	1052.07	941.154	828.93	828.94	10
C201	591.56	3	591.55	591.55	591.550	591.55	591.56	3
C203	588.49	3	1125.10	1208.94	727.221	747.93	591.17	3
R104	1007.31	9	1184.38	1184.38	1057.020	1128.29	1017.50	10
R108	963.99	9	1256.61	1186.34	1039.340	1091.69	971.91	10
R205	994.42	3	1370.68	1344.20	1150.340	1131.18	1087.80	4
R211	910.90	2	1232.48	1140.65	946.354	932.49	1101.50	4
RC101	1696.94	14	1948.94	1940.57	1734.170	1728.30	1690.60	15
RC102	1554.75	12	1803.95	1777.92	1562.620	1603.53	1470.26	13
RC105	1633.72	13	1920.19	1809.78	1597.670	1688.77	1611.10	15
RC203	1060.45	3	1627.74	1522.68	1179.670	1189.06	1064.80	5
RC207	1062.05	3	1539.85	1497.54	1295.900	1304.48	1040.67	4

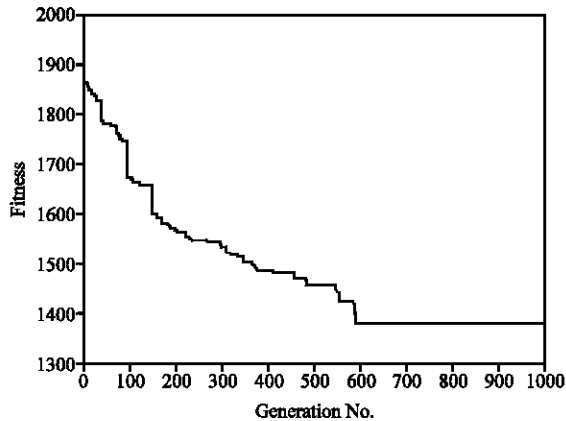


Fig. 9: Convergence diagram of RC102 problem

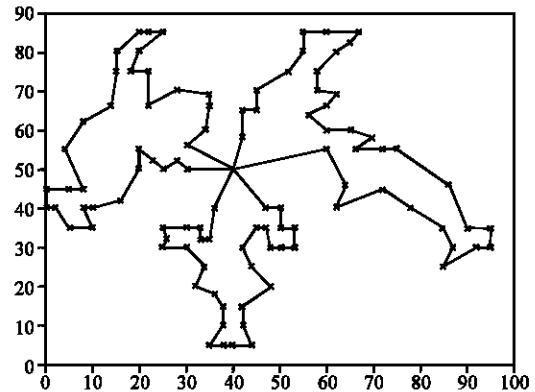


Fig. 11: Best solution of C201 problem

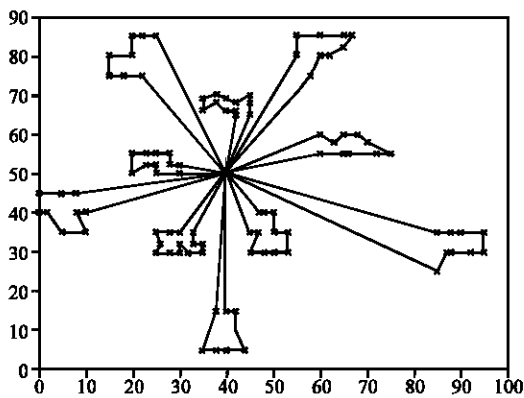


Fig. 10: Best solution of C106 problem

According to Table 1, the results obtained by the model are superior to other methods namely 2-INT, SA, GA and TS. Further, they are better than those results in all selected instances. It must be mentioned that the results presented in Table 1 are based on the following parameters:

- Population size = 100
- Generation No. = 1000
- Crossover rate = 0.80
- Mutation rate = 0.20
- No. of chromosomes that are selected for undergoing improvement phase = 10

Figure 9 shows the convergence behavior of the fitness function of the RC102 problem in 1000 generations.

Also, Fig. 10 and 11, for instance show a typical output for problems C106 and C201. Finally, according to produced results by present model, suggestive method in general are quite good and effective as compared to the

best published results and the results that are produced by other metaheuristics and the average GA performance is good.

### CONCLUSION

This study aimed to solve Vehicle Routing Problem with Time Windows (VRPTW), which has received considerable attention in recent years, using hybrid genetic algorithm. Vehicle Routing Problem with Time Windows is an extension of the well-known Vehicle Routing Problem (VRP) and involves a fleet of vehicles set-off from a depot to serve a number of customers, at different geographic locations, which various demands and within specific time windows before returning to the depot eventually. According to solve this problem, this study suggested a hybrid genetic algorithm (with special operators) that is combined with Push Forward Insertion Heuristic (PFIH) to make an initial solution and  $\lambda$ -interchange mechanism to neighborhood search and improving method. The proposed genetic algorithm used an integer representation in which a string of customer identifiers represents the sequence of deliveries covered by each of the vehicles. Part of initial population (50%) was initialized using Push Forward Insertion Heuristic (PFIH) and part is initialized randomly. A  $\lambda$ -interchange mechanism interchanged customers between routes and generates neighborhood solution. Also  $\lambda$ -interchange mechanism is used in two strategies (FB) and (GB) for more improvements on portion of solutions at the end of each generations. At the end, in order to prove the validity of the suggestive model, fourteen instances of Solomon's 56 benchmark problems-that were selected randomly- were solved and compared with the other



meta-heuristic methods namely and 2-INT, SA, GA and TS reported by Tan *et al.* (2001) and Thangiah (1999). Also the obtained results were compared with the best known solutions reported in the literature. The results show very good quality and time saving of the method. Finally, according to results produced by this model, suggestive methods in general were quite good and effective as compared to the best published results and the results that are produced by other metaheuristics and the average GA performance is good.

## REFERENCES

- Achutan, N., L. Caccettal and S. Hill, 2003. An improved branch-and-cut algorithm for the capacitated vehicle routing problem. *Transport. Sci.*, 37: 153-169.
- Alvarenga G.B., G.R. Mateus and G. de Tomi, 2007. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Comput. Operat. Res.*, 34: 1561-1584.
- Berger, J. and M. Barkaoui, 2003. A hybrid genetic algorithm for the capacitated vehicle routing problem. Genetic and Evolutionary Computation Conference (GECCO03), LNCS 2723, January 01, Springer-Verlag, Chicago, pp: 646-656.
- Blanton, J.L., Jr. and R.L. Wainwright, 1993. Multiple vehicles routing with time and capacity constraints using genetic algorithms. Proceedings of the 5th International Conference on Genetic Algorithms, June 1993, USA., pp: 452-459.
- Bräysy, O. and M. Gendreau, 2002. Tabu search heuristics for the vehicle routing problem with time windows. *TOP.*, 10: 211-237.
- Bräysy, O. and M. Gendreau, 2005. Vehicle routing problem with time windows, Part II: Metaheuristics. *Transport. Sci.*, 39: 119-139.
- Cerda, J. and R. Dondo, 2007. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *Eur. J. Operat. Res.*, 176: 1478-1507.
- Correa, E.S., M.T.A. Steiner, A.A. Freitas and C. Carnieri, 2001. A genetic algorithm for the P-median problem. Genetic and Evolutionary Computation Conference (GECCO2001), July, San Francisco, California, USA., pp: 1268-1275.
- Crevier, B., J.F. Cordeau and G. Laporte, 2007. The multi-depot vehicle routing problem with inter-depot routes. *Eur. J. Operat. Res.*, 176: 756-773.
- Czech, Z.J. and P. Czarnas, 2002. Parallel simulated annealing for the vehicle routing problem with time windows. 10th Euromicro Workshop on Parallel, Distributed and Network-Based Processing, January 9-11, Canary Islands, Spain, pp: 376-383.
- De Backer, B., V. Fumon, P. Shaw, P. Kilby and P. Prosser, 2000. Solving vehicle routing problems using constraint programming and metaheuristics. *J. Heuristics*, 6: 501-523.
- Gambardella, L.M., E. Taillard and G. Agazzi, 1999. MACSVRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In: *New Ideas in Optimization*, David, C., M. Dorigo and F. Glover (Eds.). McGraw-Hill, London, ISBN: 0-07-709506-5, pp: 63-76.
- Holland, J.H., 1992. *Adaptation in Natural and Artificial System*. 2nd Edn, University of Michigan Press, Ann Arbor, Michigan, ISBN: 0-262-58111-6.
- Irnich, S., B. Funke and T. Grunert, 2006. Sequential search and its application to vehicle routing problems. *Comput. Operat. Res.*, 33: 2405-2429.
- Kim, B.I., S. Kim and S. Sahoo, 2006. Waste collection vehicle routing problem with time windows. *Comput. Operat. Res.*, 33: 3624-3642.
- Laporte, G. and F. Semet, 2001. *Classical Heuristics for Capacitated VRP*. 1st Edn., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA., ISBN: 0-89871-498-2, pp: 109-128.
- Li, F., B. Golden and E. Wasil, 2005. Very large scale vehicle routing: new test problems algorithms and results. *Comput. Operat. Res.*, 32: 1165-1179.
- Ombuki, B., B. Ross and F. Hanshar, 2006. Multi-objective genetic algorithm for vehicle routing problem with time windows. *Applied Intell.*, 24: 17-30.
- Petch, R.J. and S. Salhi, 2003. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Math.*, 133: 69-92.
- Pisinger, D. and S. Ropke, 2005. A general heuristic for vehicle routing problems. *Comput. Operat. Res.*, 34: 2403-2435.
- Rochat, Y. and E.D. Taillard, 1995. Probabilistic diversification and intensification in local search for vehicle routing. *J. Heuristics*, 1: 147-167.
- Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming, January 01, Springer-Verlag, pp: 417-431.
- Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operat. Res.*, 35: 254-265.

- Tan, K.C., L.H. Lee, K.Q. Zhu and K. Qu, 2001. Heuristic methods for vehicle routing problem with time windows. *Artificial Intell. Eng.*, 15: 281-295.
- Tan, K.C., Y.H. Chew and L.H. Lee, 2006. A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Comput. Optim. Appl.*, 34: 115-151.
- Tan, K.C., C.Y. Cheong and C.K. Goh, 2007. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *Eur. J. Operat. Res.*, 177: 813-839.
- Thangiah, S.R., 1999. A Hybrid Genetic Algorithms, Simulated Annealing and Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. In: *Practical Handbook of Genetic Algorithms, Complex Structures*, Chambers, L. (Ed.). Vol. 3, CRC Press, New York, ISBN-10: 0849325390, pp: 347-381.
- Wren, A., 1971. *Computers in Transport Planning and Operation*. 1st Edn., Ian Allan Shepperton, London, ISBN: 711002355.