Optimized Crossover Genetic Algorithm for Vehicle Routing Problem with Time Windows

H. Nazif and L.S. Lee
Department of Mathematics, Faculty of Science, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Abstract: Problem statement: In this study, we considered the application of a genetic algorithm to vehicle routing problem with time windows where a set of vehicles with limits on capacity and travel time are available to service a set of customers with demands and earliest and latest time for serving. The objective is to find routes for the vehicles to service all the customers at a minimal cost without violating the capacity and travel time constraints of the vehicles and the time window constraints set by the customers. Approach: We proposed a genetic algorithm using an optimized crossover operator designed by a complete undirected bipartite graph that finds an optimal set of delivery routes satisfying the requirements and giving minimal total cost. Various techniques have also been introduced into the proposed algorithm to further enhance the solutions quality. Results: We tested our algorithm with benchmark instances and compared it with some other heuristics in the literature. The results showed that the proposed algorithm is competitive in terms of the quality of the solutions found. Conclusion/Recommendations: This study presented a genetic algorithm for solving vehicle routing problem with time windows using an optimized crossover operator. From the results, it can be concluded that the proposed algorithm is competitive when compared with other heuristics in the literature.

Key words: Genetic algorithm, vehicle routing problem, time windows

INTRODUCTION

The Vehicle Routing Problem with Time Windows (VRPTW) which is an extension of Vehicle Routing Problems (VRPs) arises in a wide array of practical decision making problems. Instances of the VRPTW occur in rail distribution, school bus routing, mail and newspaper delivery, airline and railway fleet routing and etc. In general the VRPTW is defined as follows: given a set of identical vehicles \( V = \{1,2,\ldots,K\} \), a central depot node, a set of customer nodes \( C = \{0,1,2,\ldots,N\} \) and a directed network connecting the depot and customers. Each arc in the network represents a connection between two nodes and also indicates the direction it travels. The depot is denoted as customer 0, which uses K independent delivery vehicles, with delivery capacity \( q_k \), \( k = 1,2,\ldots,K \), to service demands \( m_i \) from \( n \) customers, \( i = 1,2,\ldots,N \). The time window constraint is denoted by a predefined time interval \([a_i,b_i]\) for each customer i, respectively. The vehicles must arrive at the customers not later than the latest arrival time, if vehicles arrive earlier than the earliest arrival time, waiting occurs. Each customer also imposes a service time \( s_i \), taking consideration of the loading/unloading time of goods. A non-negative cost (distance or travel time) matrix \( C = (c_{ij}) \) between customers i and j is defined on the network. A solution for the VRPTW would be a partition \( R_1,R_2,\ldots,R_K \), representing the routes of the vehicles, each route \( R_k \) is a permutation of the customers in C specifying the order of visiting them, starting and ending at the depot. The cost of the problem solution is the sum of the costs of its routes \( R_k \), defined as follows:

\[
\text{Cost} = \sum_{k=1}^{K} \text{Cost}(R_k)
\]

The VRPTW consists in determining a set of a maximum of K routes (i) of minimum total cost (Eq. 1); (ii) starting and ending at the depot denoted with customer 0 and such that (iii) each customer is visited exactly once by exactly one vehicle; subject to the restrictions (iv) the total demand of any route \( R_k \) does not exceed \( q_k \); (v) each route \( R_k \) must be completed within a total route time, which is essentially the time
window of the depot; (vi) the vehicles must visit the customers within predefined time windows.


Blanton and Wainwright (1993) were the first to apply a Genetic Algorithm (GA) to VRPTW. They hybridized a GA with a greedy heuristic. GAs have been proposed for the problem by Thangiah (1995); Potvin and Bengio (1996); Berger et al. (1998; 1999); Berger and Barkaoui (2000; 2004); Bräysy et al. (2000) and Tan et al. (2001a). Gehring and Homberger (2002) proposed a parallelization of a two-phase metaheuristic for solving VRPTW.

In this study, we consider the vehicle routing problem with time windows and propose an Optimized Crossover Genetic Algorithm (OCGA) for this problem.

**MATERIALS AND METHODS**

**Optimized Crossover Genetic Algorithm (OCGA):**

GA, originally developed by Holland and Holland (1975) is an adaptive heuristic search method that mimics evolution through natural selection. It works by combining selection, crossover and mutation operations. The selection pressure drives the population toward better solutions while crossover uses genes of selected parents to produce offspring that will form the next generation. Mutation is used to escape from local minima.

Optimized crossover was proposed by Aggarwal et al. (1997) for the independent set problem. They applied optimized crossover within genetic algorithm and produced two new children: The Optimum child (O-child) and Exploratory child (E-child). The O-child is constructed in such a way that has the best objective function value from a feasible set of children, while the E-child is constructed so as to maintain the diversity of the search space.

In the remainder, we describe the proposed optimized crossover and discuss how various steps of the GA are implemented for the proposed algorithm. The general framework of OCGA can be shown as follows:

**Algorithm OCGA:**

begin
  Initialize population (randomly generated);
  Fitness evaluation;
  repeat
    Selection (probabilistic binary tournament selection);
    Optimized crossover;
    Swap node (if the optimized crossover is not applied);
    Mutation (inversion and swap sequence);
    Fitness evaluation;
    Elitism replacement with filtration;
  until the end condition is satisfied;
  return the fittest solution found;
end

**Individual representation and selection mechanism:**

The representation of a solution we use here is an integer string of length N, where N is the number of customers, applied by Tan et al. (2001b). Each gene in the string is the integer node number assigned to that customer originally. The sequence of the genes in the string is the order of visiting these customers. For example, if we have the following solution represented in Fig. 1.

![Fig. 1](https://example.com/fig1.png)

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

The coded integer string is then:

2 → 3 → 1 → 4 → 5 → 6 → 10 → 8 → 9 → 7 → 11 → 12
Fig. 1: A solution to a vehicle routing problem

This representation is unique and one string can only be decoded to one solution. It is a 1-to-1 relation. The last customer visited in route i is linked with the first customer visited in route i+1 to form a string of all the routes involved. Note that we do not put any bit in the string to indicate the end of a route now, because such delimiters in a string greatly restrain the validity of children produced by optimized crossover operator later. To decode the string into route configurations, the gene values are inserted into the new routes sequentially. There is a chance that we may not get back the original routes after decoding, but it is generally assumed that minimizing the number of routes helps in minimizing the total travel cost, therefore, packing a route to its maximum capability implies a potential good solution as a result (Tan et al., 2001b).

After choosing the representation, we uniformly randomly generate an initial population using a random number generator. Moreover we use a probabilistic binary tournament selection scheme to select individuals from the population to be the parents for the OCGA with a given selection probability \( p_s = 0.75 \). In other words, we give a 75% chance for the fitter individual to be selected as the parent compared to the less fit individual which only has a 25% chance to be selected.

**Optimized crossover:** We propose an optimized crossover operator within a GA for the VRPTW. The proposed optimized crossover using a complete undirected bipartite graph finds the two new children which are called O-child and E-child. We will now explain the optimized crossover strategy on determining the O-child and E-child for the VRPTW.

**Step 1:** Given two parents \( P_1 \) and \( P_2 \). Construct a complete undirected bipartite graph \( G = (U \cup V, E_1 \cup E_2) \) where \( U = \{u_1, u_2, ..., u_n\} \) representing customers, \( V = \{v_1, v_2, ..., v_n\} \) representing nodes, \( E_1 \) and \( E_2 \) representing the arc sets in the graph in which, \( \{u_j, v_i\} \in E_1 \) if and only if, customers \( j \) of parent \( P_1 \) is located at node \( i \) and \( \{u_j, v_i\} \in E_2 \) if and only if, customers \( j \) of parent \( P_2 \) is located at node \( i \).

**Step 2:** Determine all perfect matchings in graph \( G \). Suppose that there are \( k \) cycles corresponding to different locations of customers in the two parents. There will be exactly \( 2^k \) perfect matchings in graph \( G \). Note that each perfect matching representing a temporary offspring.

**Step 3:** Select a temporary offspring with the least objective function value as O-child.

**Step 4:** Generate E-child using \((P_1 \cup P_2) \setminus \text{O-child} \cup (P_1 \cap P_2)\).

We restricted the temporary offspring in graph \( G \) in every case to \( 2^k \) even if the cycles are more than 5. In addition, the proposed optimized crossover is applied based on a crossover probability, \( p_c = 0.75 \).

**Swap node:** In this study, we use a swap node operator instead of reproduction to produce two new offspring when the optimized crossover is not applied to the parents. This operator could be regarded as a giant mutation where the elements in the parent are randomly reassigned. The process of swap node operator is as follows: two nodes from a parent are randomly selected and swapped. It is repeated for the second parent to create a second offspring. An example of swap node operator is given in Fig. 2.

**Mutation:** The mutation operator used in the proposed algorithm consists of applying with equal probability two different mutation operators called inversion and swap sequence operators. The inversion operator reverses the visiting order of the customers between two randomly selected points, while the swap sequence operator consists of randomly selecting two sub-strings of customers and exchanging them. The examples of the two mutation operators are given in Fig. 3.

**Elitism replacement with filtration:** Elitism replacement scheme is applied in this study as follows: Both parent and offspring population are combined into a single population and sorted in a non-increasing order of their associated fitness value. Then, the first half of the combined population is selected as the individuals of the new population for the next generation. In order to avoid premature convergence and to add diversity to the new population, we propose the filtration strategy in which, identical individuals are identified from the new population and, they are removed and replaced by uniformly randomly generated new individuals. As the filtration strategy requires a certain amount of computational time, the procedure will only be invoked after every 50 generations.

**Fig. 2:** An example of swap node
Each customer \( i \) has a time window \([a_i, b_i]\), which represents the time interval to arrive in that customer. Different capacity constraints are considered for the vehicle in each class of instance, as well as the demands from the customers.

We compare our proposed algorithm against the few works have considered Solomon's benchmark problems minimizing Travel Distance (TD).

### RESULTS

We present the results of the computational experiments of our proposed algorithm on VRPTW. The proposed algorithm is coded in C language and implemented on a Pentium 4, 2.0 GHz computer with 2.0 GB RAM.

The OCGA has been tested on all six classes of the benchmark of Solomon (1987) which are 100-customer problem sets. The classes R1 and R2 have customers randomly disposed, while the classes C1 and C2 have customers clustered. The classes RC1 and RC2 contain a subset of customers randomly disposed and the other part clustered. The distance between two customers is the simple Euclidean distance. One unit of time is necessary to run one unit of distance by one vehicle.

Fig. 3: Examples of the two mutation operators

### DISCUSSION

The results are presented in Table 1 together with the best known solution in the literature. The numerical results are computed after making 50 independent runs for statistical significance. The objective function applied is only the total TD. Results in Table 1 represent the best known solution is found or improved (better TD) by OCGA in 32 problems. It is clear to see that the proposed algorithm continues to be competitive in terms of total TD.

Following, Table 2 shows the results by problem class, including results from other relevant works in the literature. We have selected the heuristic methods proposed by Taillard et al. (1997); Rousseau et al. (1999); Gehring and Homberger (2002); Tan et al. (2001b) and Berger and Barkaoui (2004). We compute the percentage Deviation (D) in terms of travel distance between our best solution (sol) and the published best (best) for each problem class with the following equation:

\[
D = \left( \frac{\text{best sol} - \text{sol}}{\text{best}} \right) \times 100\% \tag{2}
\]

We present the results of the computational experiments of our proposed algorithm on VRPTW. The proposed algorithm is coded in C language and implemented on a Pentium 4, 2.0 GHz computer with 2.0 GB RAM.

The OCGA has been tested on all six classes of the benchmark of Solomon (1987) which are 100-customer problem sets. The classes R1 and R2 have customers randomly disposed, while the classes C1 and C2 have customers clustered. The classes RC1 and RC2 contain a subset of customers randomly disposed and the other part clustered. The distance between two customers is the simple Euclidean distance. One unit of time is necessary to run one unit of distance by one vehicle. Each customer \( i \) has a time window \([a_i, b_i]\), which represents the time interval to arrive in that customer. Different capacity constraints are considered for the vehicle in each class of instance, as well as the demands from the customers.

We compare our proposed algorithm against the few works have considered Solomon's benchmark problems minimizing Travel Distance (TD).

### RESULTS

We present the results of the computational experiments of our proposed algorithm on VRPTW. The proposed algorithm is coded in C language and implemented on a Pentium 4, 2.0 GHz computer with 2.0 GB RAM.

The OCGA has been tested on all six classes of the benchmark of Solomon (1987) which are 100-customer problem sets. The classes R1 and R2 have customers randomly disposed, while the classes C1 and C2 have customers clustered. The classes RC1 and RC2 contain a subset of customers randomly disposed and the other part clustered. The distance between two customers is the simple Euclidean distance. One unit of time is necessary to run one unit of distance by one vehicle. Each customer \( i \) has a time window \([a_i, b_i]\), which represents the time interval to arrive in that customer. Different capacity constraints are considered for the vehicle in each class of instance, as well as the demands from the customers.

We compare our proposed algorithm against the few works have considered Solomon's benchmark problems minimizing Travel Distance (TD).

### DISCUSSION

The results are presented in Table 1 together with the best known solution in the literature. The numerical results are computed after making 50 independent runs for statistical significance. The objective function applied is only the total TD. Results in Table 1 represent the best known solution is found or improved (better TD) by OCGA in 32 problems. It is clear to see that the proposed algorithm continues to be competitive in terms of total TD.

Following, Table 2 shows the results by problem class, including results from other relevant works in the literature. We have selected the heuristic methods proposed by Taillard et al. (1997); Rousseau et al. (1999); Gehring and Homberger (2002); Tan et al. (2001b) and Berger and Barkaoui (2004). We compute the percentage Deviation (D) in terms of travel distance between our best solution (sol) and the published best (best) for each problem class with the following equation:

\[
D = \left( \frac{\text{best sol} - \text{sol}}{\text{best}} \right) \times 100\% \tag{2}
\]
CONCLUSION

This study presents a genetic algorithm that uses an optimized crossover operator to solve the vehicle routing problem with time windows. The proposed algorithm has been tested against the best known solutions reported in the literature, using 56 Solomon’s problems with 100 customers. The computational results showed that the proposed algorithm is competitive in terms of the quality of the solutions found (in terms of total travel distance). As for future work, it may be interesting to test OCGA with additional benchmarks of VRPTW.

REFERENCES


