

The Impact of Crossover and Mutation Operators on a GA Solution for the Capacitated Vehicle Routing Problem

Hazan Dağlayan, Murat Karakaya*

Department of Computer Engineering, Atilim University, İncek, 06836, Ankara, Turkey

Copyright ©2016 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract The Vehicle Routing Problem (VRP) is one of the well-known NP hard problems requiring excessive time to be exactly solved. Therefore, for solving this type of problems, some researchers implemented meta-heuristics such as Genetic Algorithm (GA). In this paper, we study the Capacitated VRP (CVRP) which has some constraints on the capacities of the vehicles used in VRP. The goal of this study is to observe the impact of the selected operators of GA on the quality of the generated solutions. Therefore, we propose 6 different GAs by mixing and combining 3 crossover and 5 mutation operators. We observed the performance of these solutions by applying them over 10 CVRP benchmarks.

Keywords Capacitated Vehicle Routing Problem, Genetic Algorithm, Vehicle Routing Problem

1 Introduction

The Vehicle Routing Problem (VRP) is first introduced by Dantzig and Ramser in 1959 [1]. The problem deals with delivering goods to customers from different depots provided that customer locations and demands are known and the route starts and finishes at the same depot. The aim is to find an optimum route with the minimum cost.

There are many variants of the VRP according to the considered constraints: Capacitated VRP (CVRP) [2–4], Multiple Depot VRP (MDVRP) [5, 6], VRP with Time Windows (VRPTW) and VRP with Pick-Up and Delivering (PDVRP) [7] which are the most popular problems.

In CVRP, the goods are delivered to customers with a capacitated vehicle. Therefore, the vehicle should be reloaded at the depot when the goods are delivered. As a result, the vehicle may visit the depot more than once. CVRP is originally proposed for network problems, but then it helps to find solution to travelling, customer' demands and location problems [8] as well.

The proposed solutions to the problem can be classified under three headings: exact solutions, heuristic methods, and

meta-heuristic methods. While exact solutions provide the optimal solutions, meta-heuristic and heuristic does not guarantee it. However, for the problems with considerable number of customers exact solutions may be infeasible [9]. To find solutions for such problems, heuristic and meta-heuristic methods are proposed. Meta-heuristic methods can provide some good solutions exploiting wisely heuristic methods. Some popular meta-heuristics are Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSA) among others [8]. GA is a meta-heuristic method that imitates evolutionary process in the nature. In essence, GA modifies the population of individual solutions repeatedly with the operations such as crossover and mutations to reach an optimized solution. Due to its success and easy implementation, GA is applied to different optimization problems of different research areas [10].

In this paper, we investigate the quality of the solutions generated by GA when different crossover and mutation operators are implemented for solving CVRP. We opt to implement, mix, and compare 3 number of crossover and 5 number of mutation operators proposed in two GA solutions in [11] and [12]. Our aim is to observe the impact of the selected operators on the quality of the solution. As GA is a stochastic search method, random and heuristic choices can affect the success of the process.

The remainder of this paper is organized as follows. A detailed presentation of CVRP is given in Section 2. GA and its operators are described in Section 3. In Section 4 and Section 5, we propose GA implementations using different mutation and crossover operators are presented. The results of these GA implementations are given in Section 6. Finally, in Section 7, we provide conclusions.

2 Capacitated Vehicle Routing Problem

Capacitated Vehicle Routing Problem (CVRP) can be defined as a graph $G = (V, A)$ where $V = 0, 1, 2, \dots, n$ is the vertex set and $A = \{(i, j) \mid i, j \in V\}$ is the arc set.

As seen in Fig. 1, Vertex 0 is the central depot while vertices $1, 2, \dots, n$ correspond to n customers. Each customer has demands d_i and the vehicle has a delivery capacity C . A solution for CVRP consists of k routes $\{R_1, R_2, \dots, R_k\}$ satisfying the following equation

$$\sum_{i \in R_j} d_i \leq C \quad (1)$$

In the CVRP, the aim is to minimize the total cost as described in (2) where the cost c_{ij} is associated with each arc $(i, j) \in A$.

$$TotalCost = \sum_{i,j \in V} c_{ij} \quad (2)$$

Each cost c_{ij} represents the distance between customer i and customer j .

3 Genetic Algorithm

GA belongs to class of evolutionary algorithms. GA begins with some possible problem solutions which is called as population. The of each individual (solutions) in the population is the solution cost. Optimization problem is to find an individual which minimize or maximize this fitness value. New generations are created with crossover and mutation operators applied on selected individual in the population. With a selection method, two individuals are chosen from the population as parents, and two offsprings are created applying crossover and mutation operators on these parents. When an optimum route satisfying a threshold is obtained or generation number is reached to the limit, the termination condition is met and GA outputs the solution with maximum or minimum fitness value.

3.1 Initialization

Initialization is the first process of the GA meta-heuristic. It may be usually randomized, while sometimes the solution of a heuristic method may be joined to the random population [13]. However, improper selection of initial population may lead to stack to local maximum or minimum points. That is, some solution may be so dominant that other individual (chromosomes) cannot create better offsprings (solution).

In most of the solution for CVRP, after random initialization, a greedy approach is used to handle capacity limitations [11]. Thus, it is assumed that the vehicle visits the first customer and serves the customer according to customer demand and vehicle capacity. If the vehicle has enough delivery for the next customer, it will visit. Otherwise, the vehicle returns to the depot. For instance, considering Fig. 1, if the order of a possible solution (chromosome) is as 1 4 5 6 3 7 2, the demand of the customers is 3 2 1 3 3 2 4, and the capacity of vehicle is 6, the chromosome consist of 3 separate tours initiating and ending at the starting depot 0 as seen in Fig. 1.

3.2 Encoding Schema

In this paper, we represent each city with a unique number, and except the depot no number will be repeated in a chromosome. For example, if we have 10 cities the individual

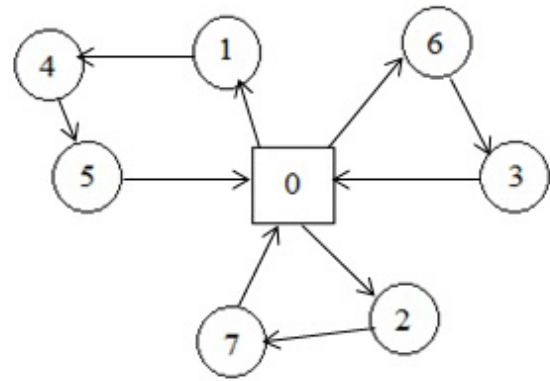


Figure 1. Representation of a CVRP example

consists of 10 or more numbers (with depot 0) from 1 to 10 with the order randomly permutation. The individual may be like that 3 2 8 0 1 7 6 0 4 5 9 0 1 0, where zero indicates the depot [14].

3.3 Fitness Value

Fitness value is calculated after a new offspring is created. Each individual (chromosome) has a fitness value. It helps to find the best individual. If the individual has the maximum fitness value, it will be the best individual of the population. In CVRP, if the total distance is the minimum for an individual (solution), it means that it has the maximum fitness value. Therefore, usually the fitness value is formulated as [8, 15]

$$f_i = \frac{100}{TotalDist_i} \quad f_i = \frac{TotalDist_i}{\sum TotalDist_i} \quad (3)$$

where f_i is the fitness value and $TotalDist_i$ is the total distance of the i^{th} individual.

3.4 Selection Operator

The selection operator is used to choose the parents from the current population to create new individual for the next generation by applying crossover or mutation operators.

Some well-known selection operators are tournament selections, roulette wheel selection, and steady-state approach. The tournament selection helps to find the best individual of tournament size number individuals [4, 8, 11]. For instance, if the tournament size is 2, which is called binary tournament size, two individuals are selected randomly, and the one with the better fitness value is the winner and selected as the parent.

The Roulette Wheel Selection is proposed to select the chromosomes which have greater fitness values. If the fitness value is greater, the probability of being selected will be greater [7, 8].

The steady-state approach ensures that two parents and two offsprings are compared; it selects the best two chromosomes of them [2].

3.5 Crossover Operator

Crossover is the main process of the GA meta-heuristic to vary the chromosomes from one generation to the next.

Selected two parents consist of two offsprings implementing some steps according to type of crossover. There are various types of crossover operators such as Partially Mapped Crossover (PMX), Route Better Adjustment Recombination, Order Crossover, Pereira Crossover, etc.

PMX is one of the well-known crossover operators. Two cut points are selected randomly and offspring inherits the genes between these two cut points with the same order. Then, a mapping table is constructed between two parents. According to second parent's position, if the gene is not located between cut points, it is placed in the first child's chromosome. However, if the gene is already copied, it is ignored.

The following example explains how PMX operates. Assume that the chromosomes of selected parents are as follows: P_1 is (3 5 2 1 4 8 7 5) and P_2 is (2 4 5 6 7 3 1 8). Then, two cut points randomly generated and showed as black.

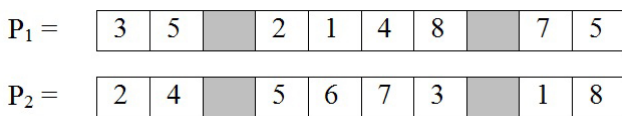


Figure 2. The example of parent for CVRP

Exchanging genes between the cut points:

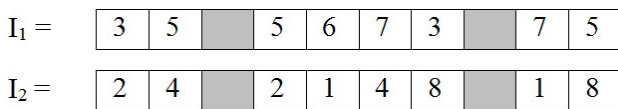


Figure 3. The example of interim offsprings for CVRP

Mapping will be as:

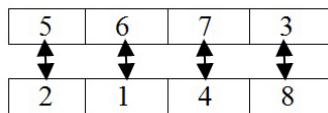


Figure 4. Mapping table for CVRP

After mapping, new generations will be as:

Best Route Better Adjustment Recombination (BRBAX) is a constructed crossover for this problem [11]. This operator aims to transmit the best routes to the new generations which have less distance and satisfy the capacity constraint. The procedure is as follows. As a first step, one of the parents is selected and its m routes are sorted and $m/2$ routes which have minimum distance are selected and placed as the first genes of the child's chromosome. Then, in the second step, the other parent's genes are inherited to the child according to order of parent's chromosome. Figure 6 provides an example operation of the BRBAX crossover.

The crossover suggested by Pereira is another operator which may be used in CVRP [12]. Firstly, randomly a sub-route is selected from second parent. After, the next gene to the first gene of the sub-route is chosen, the sub-route is inserted; therefore it is satisfied shorter distance probably. In the following example, assume that customer 6 is closer to customer 9. (9, 1, 10) sub-route is selected randomly and it is put on near 6.

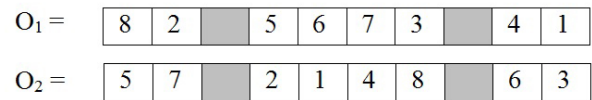


Figure 5. The example of offsprings for CVRP

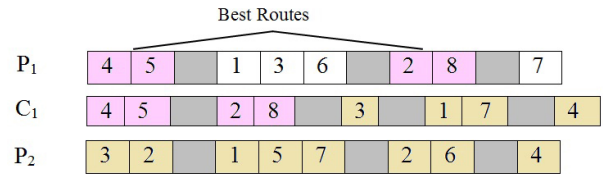


Figure 6. The example of offsprings for BRBAX

3.6 Mutation Operator

Mutation operator is used to escape the local optimum points. When mutation is not used, the local points can be seen optimum solution. Swap, insertion, displacement and inversion mutations are well-known mutation types. Details of these mutation operators are provided below.

Swap: Two points are selected randomly from the chromosome, and swaps them. The genes can be same or different routes [11, 12].

Insertion: A gene is selected randomly and it is removed from the position and it is reinserted a new position [12].

Displacement: A sub-route is selected and then it is reinserted a new position. It may be added a route or it may be a new route [12].

Inversion: The points between two positions which are selected randomly are reversed [16]

Scramble: A sub-route is selected randomly and the genes are mixed [17].

All mutations mentioned above may be local and global, this means that they may be in a route or in a chromosome.

3.7 Elitism

The best individual may be preserved by without changing the result of the crossovers and mutations. Therefore, it will be inherited to next generations without any changes, it is called elitism.

4 Selected Solutions

We have selected two solutions to implement their GA methods by mixing a permutation of crossover and mutation operators.

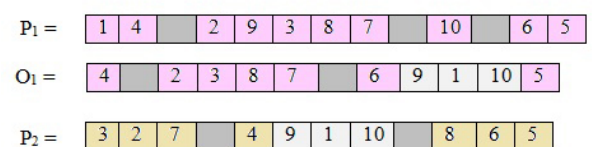


Figure 7. The example of offsprings for Pereira Crossover

4.1 Solution A

We selected Graglias solution as the first solution [11]. In Graglias solution, an initial population is created randomly. The population goes into a cycle and then it continues with crossover, mutation, selection and evaluation. Two good chromosomes are chosen by tournament selection. And then with a probability of crossover PMX or BRBAX are applied. The next process is to choose mutation with a probability. Three types mutation is tried: swap, insertion and combination of them. Also elitist strategy takes part in the algorithm. The procedure goes on until the max generation is reached.

4.2 Solution B

The solution proposed by Pereira is selected as the second solution [12]. The solution of Pereira starts with creating an initial population with the following algorithm: 1. Build a set with a random permutation of all the demands 2. While there are elements in D Repeat: 2.1. Create one new route with K demands. K is a random value between 1 and the current number of elements from D 2.2. Remove first K elements from D and assign them to the new route. Fitness function of the algorithm is the total distance of the routes, and the aim is minimizing the function. Until the maximum generation number is reached, the genetic operators will be applied. The crossover which is described above in crossover part is used and swap, insertion, inversion and displacement mutations are used in this algorithm. The elitist strategy also is included.

The common flowchart of the algorithms are given on Fig. 8

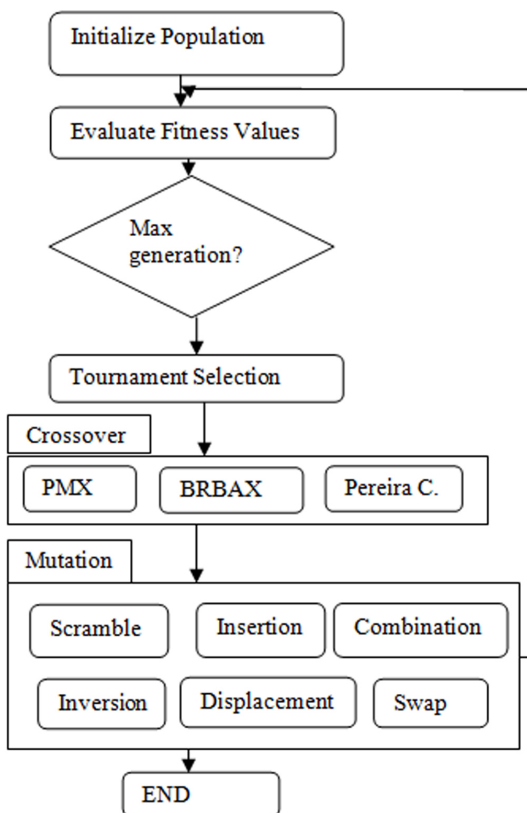


Figure 8. The flowchart of algorithms

5 Simulation Experiments and Results

The algorithms described in the previous section are coded in MATLAB. 10 different CVRP benchmark files are used for testing the algorithms. We used 6 different combinations of crossover and mutation operators. The crossover and mutations are presented on Table 2 for each algorithm. The algorithms are repeated 10 times and their average values are given on Table 3. The benchmark files nodes change between 32 and 101. The default values of GA are given on Table 1. The values of crossover and mutation probability are determined according to the empirical experiments in which the best performance values are observed when these probability values are tested. In each algorithm, the default values are constant to compare the algorithms results.

Table 1. GA parameters and their values

Parameter	Default Value
Population Size	200
Crossover Probability	0.9
Mutation Probability	0.15
Generation number	5000

To compare the algorithms, we utilise the error formula given in 4 based on the best known value and obtained value.

$$Error = \frac{|BestKnown| - |GA|}{|BestKnown|} \tag{4}$$

The first and second algorithm (A_1, A_2) have the same properties with Graglia’s algorithm [11]. In the first algorithm, BRBAX crossover is applied whereas PMX is used in the second algorithm. Pereira’s method is tested in the third algorithm. We obtained better results with Pereira’s algorithm. Since Pereira’s algorithm has more mutation types, scramble mutation is also inserted the mutation types which we used. Finally, we tested all crossover types with these mutations.

We obtained the best results from the algorithm A_4 which is consist of Pereira’s crossover and 5 mutations (swap, insertion, inversion, displacement and scramble). Since the mutation types are common on A_4, A_5 and A_6 , we have the information that Pereira’s crossover is the best type crossover.

The values in Table 4 gives the average error obtained by (4) for each algorithms. If we compare the algorithms A_3 and A_4 , it is obtained that the increase the number of mutation type decreases the error value. Moreover, comparing the algorithm A_1 and A_5 supports this idea.

When algorithms A_4 and A_6 are compared, Pereira’s crossover gives better results than PMX crossover. Besides, the average error value of PMX crossover are less than BRBAX crossover with the help of algorithms A_5 and A_6 .

6 Conclusion

In this paper, we compare the performance of 6 GAs constructed from combining of 3 crossover and 5 mutation operators for solving CVRP. As GA depends on stochastic operators, the resulting performance is expected to be affected by the selected operators. However, due to complexity and the randomness of these processes, to analyze the possible impact of these operators on the solution quality is very hard. Therefore, in these study, we conducted extensive simulation

Table 2. Properties of Algorithms

GA	Crossover	Mutation	Note
A ₁	BRBAX	swap, insertion	Graglia's method
A ₂	PMX	swap, insertion, combination of them	Graglia's method
A ₃	Pereira's crossover	swap, insertion, inversion, displacement	Pereira's method
A ₄	Pereira's crossover	swap, insertion, inversion, displacement, scrabble	
A ₅	BRBAX	swap, insertion, inversion, displacement, scrabble	
A ₆	PMX	swap, insertion, inversion, displacement, scrabble	

Table 3. Comparisons of Algorithms

Benchmarks	Algorithms (Average)						Best Known
	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	
A-n32-k5	889.8	829.4	828.7	787.1	788.3	787.2	784
E-n33-k4	909.7	837.9	859.1	837.7	869.1	846.7	835
A-n37-k6	973.2	984	1002.4	970.6	999.5	976.3	949
A-n45-k7	1274.2	1193.9	1203.8	1186.7	1206.8	1203.8	1146
vrpnc1	619.2	546.8	569.9	539.5	561.7	558.5	524.6
vrpnc6	578.6	590.5	567.4	557.2	608.6	569.3	555.4
A-n53-k7	1177.1	1090.1	1048.1	1046.8	1079.6	1052.1	1010
A-n69-k9	1348.1	1220.8	1226.2	1219.7	1250.7	1243.7	1168
vrpnc2	996.9	902.8	888.0	886.3	907.1	896.5	835.2
vrpnc3	1247.8	955.4	906.3	863.4	1042.1	965.5	826.1

Table 4. Average error values of the proposed algorithms compared to best-known solutions for 10 benchmarks

Algorithm	Average error value
A ₁	0.160
A ₂	0.058
A ₃	0.054
A ₄	0.028
A ₅	0.080
A ₆	0.052

tests to observe the effect of crossover and mutation operators on the generated solutions. We compared the generated results with the best-known values. We observed that selecting suitable crossover and mutation operators considerably affects the generated solution quality. Among the considered methods, the results shows that Pereiras crossover with higher number of different types of mutations outperforms the other combinations of operators. Thus, we can conclude that designing better crossover and mutation operators leads better genetic algorithms.

REFERENCES

- [1] G. B. Dantzig, J. H. Ramser. The truck dispatching problem, *Management science* 6.1, 80-91, 1959.
- [2] M. Filipec, D. Skrlec, S. Krajcar. An efficient implementation of genetic algorithms for constrained vehicle routing problem, *Systems, Man, and Cybernetics*, 1998, 1998 IEEE International Conference on. Vol. 3. IEEE, 1998.
- [3] S. Wink, T. Bck, M. Emmerich. A meta-genetic algorithm for solving the capacitated vehicle routing problem, 2012 IEEE Congress on Evolutionary Computation, IEEE, 2012.
- [4] H. Nazif, L. S. Lee. Optimised crossover genetic algorithm for capacitated vehicle routing problem, *Applied Mathematical Modelling* 36.5, 2110-2117, 2012.
- [5] M. Filipec, D. Skrlec, S. Krajcar. Darwin meets computers: New approach to multiple depot capacitated vehicle routing problem, *Systems, Man, and Cybernetics*, 1997, Computational Cybernetics and Simulation., 1997 IEEE International Conference on. Vol. 1. IEEE, 1997.
- [6] M. Skok, D. Skrlec, S. Krajcar. The genetic algorithm method for multiple depot capacitated vehicle routing problem solving. *Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, 2000, Proceedings, Fourth International Conference on. Vol. 2. IEEE, 2000.
- [7] A. S. Tasan, M. Gen. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries, *Computers & Industrial Engineering* 62.3, 755-761, 2012.
- [8] M. A. Mohammed, M. S. Ahmad, S. A. Mostafa. Using genetic algorithm in implementing capacitated vehicle routing problem, *Computer & Information Science (ICCIS)*, 2012 International Conference on. Vol. 1. IEEE, 2012.
- [9] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms, *European Journal of Operational Research* 59.3, 345-358, 1992.
- [10] D. Beasley, D. R. Bull, R. R. Martin. An overview of genetic algorithms: Part 2, research topics, *University computing* 15.4, 170-181, 1993.
- [11] P. Graglia, N. Stark, H. Alfonso, C. Salto. A comparison of recombination operators for capacitate vehicle routing problem. In *XIV Congreso Argentino de Ciencias de la Computacion*, 2008.
- [12] F.B. Pereira, J. Tavares, P. Machado, E. Costa. GVR: a new genetic representation for the vehicle routing problem, *Irish Conference on Artificial Intelligence and Cognitive Science*, Springer Berlin Heidelberg, 2002.
- [13] E. Khaji, A. S. Mohammadi. A Heuristic Method to Generate Better Initial Population for Evolutionary Methods, arXiv preprint arXiv:1406.4518, 2014.
- [14] N. Kumar, R. K. Karambir. A comparative analysis of pmx, cx and ox crossover operators for solving traveling salesman problem, *International journal of Latest Research in science and technology* 1, 2012.
- [15] C. REN. Applying genetic algorithm for capacitated vehicle routing problem, *2nd International Conference on Electronic & Mechanical Engineering and Information Technology*, Atlantis Press, 2012.

- [16] K. Puljić. An Evolutionary Algorithm Based on Repeated Mutations for Solving the Capacitated Vehicle Routing Problem, CIT. Journal of Computing and Information Technology 20.1, 51-58, 2012.
- [17] Y. Y. Leu, L. A. Matheson, L. P. Rees. Assembly Line Balancing Using Genetic Algorithms with Heuristic Generated Initial Populations and Multiple Evaluation Criteria. Decision Sciences 25.4, 581-605, 1994.