J. Civil Eng. Mater.App. 2020 (June); 4(2): 103-114

Journal of Civil Engineering and Materials Application

Journal home page: http://jcema.com

Received: 10 January 2020 • Accepted: 19 April 2020

Research

doi: 10.22034/jcema.2020.224281.1019

Developing genetic algorithm to solve Vehicle Routing Problem with Simultaneous Pickup and Delivery

Amir Masoud Rahimi [1]*, Vahid Rajabi [2], Ehsan Ramezani Khansari [3]

[1] Department of Civil Engineering, Faculty of Engineering, University of Zanjan, Zanjan, Iran.
[2] Transportation Engineering, Imam Khomeini International University, Qazvin, Qazvin, Iran.
[3] Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran.

*Correspondence should be addressed to Amir Masoud Rahimi, Department of Civil Engineering, Faculty of Engineering, University of Zanjan, Zanjan, Iran. Tel: +989123000222; Fax: +98xxxxxxxx; Email: amrahimi@znu.ac.ir .

ABSTRACT

One of the well-known and highly used extensions of vehicle routing problem (VRP) is Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD), in which delivery and pickup for each customer is carried out simultaneously. In this study, it is attempted to present an optimal method for solving VRPSPD using genetic algorithm. In this method, genetic algorithm is improved by modifying genetic parameters and presenting efficient and proper operators. Three Randomized, nearest neighbor and Cheapest Insertion algorithms are utilized to create the initial population. Given the different structure used in each of these methods, the initial solutions are varied and include all feasible regions. In addition, by making modifications in these methods, the initial population was tried to be created through higher quality solutions to help genetic algorithm reach a better future generation. Also, 4 algorithms were invented for mutation operators, which prevented convergence in local optimums and helped finding better solutions by comparing the results. The proposed algorithm is executed on 40 different standard examples. After comparing the results by this algorithm and the best solutions by other algorithms, improvement is observed in 3 of the examples.

Keywords: NP-hard problems, Vehicle Routing Problem, genetic algorithm

1. INTRODUCTION

Finding the optimum route for delivering orders and goods in cities leads to reduction in costs such as fuel and trip time span. One of them is when a vehicle with goods visits specific costumer and picks up and delivers them simultaneously. This sort of freight transportation has been extended today. Calculating the route with lowest cost can be considered as an optimization problem in mathematics. And many deterministic solutions have been provided. By increasing the number of costumers, vehicles and routs, the complexity of problem and its solving time by deterministic methods will increase exponentially. Then deterministic methods lose their efficiency. Therefore in the recent decades, using Meta-heuristic methods for solving such a complicated problems in an acceptable time and need of processing source was the goal of many studies. Min defined new restriction for VRP in 1989, later known as Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD) [1]. It can be said VRP is based on the traveling salesman problem, which is well known for decades[2, 3]. A long time after Min, Salhi and Nagy (1999) solved this problem using a heuristic method [4]. In the first step, they solved the Capacitated VRP problem, only considering the customers who pick up goods. Next, they added the customers who deliver goods on the route. The functions of VRPSPD in reverse logistics problem were highlighted for the first time by Dethloff [5]. Tang Montane and Galvao defined 3 more types of VRPDP problem, and solved them using Wheel heuristic, Minimum Spanning Tree and Cheapest Insertion heuristic [6]. Salhi and Nagy developed their previous studies. Using more node operators, they invented the heuristic, which helped correcting the solution [7]. Chen and Wu proposed an Insertion method and a hybrid meta-heuristic algorithm based on the record-to-record method, including taboo list and improving algorithms [8]. Another method to solve VRPSPD is an algorithm presented by Tang and Galvao. They proposed a taboo search meta-heuristic algorithm including further penalty steps[9]. Bianchessi and Righini applied a method, developed by combining local search and taboo search algorithms, on Dethloff's examples, which yielded better result[10]. Zachariadis, et al. proposed a hybrid framework based on two known heuristic algorithms, i.e. taboo search and local search[11]. Gajpal and Abad applied an improved ant colony optimization algorithm to solve VRPSPD [12]. Catay also developed an ant colony algorithm, equipped with a saving function and pheromone update method[13]. Zachariadis presented a meta-heuristic for VRPSPD, which, using an efficient local search method, explores a neighbor solution [14]. This paper devoted to basic GA which its functionality is demonstrated in many scientific and engineering problems [15, 16], but there are some other papers that applied a combination of GA and other algorithm such as ant colony[17–19], artificial neural network [20, 21], simulated annealing [22, 23] and Particle Swarm Optimization [24–27]. In this study, a genetic heuristic algorithm is proposed for solving Vehicle Routing Problem with Simultaneous Pickup and Delivery.

2. MATERIALS AND METHODS

Vehicle Routing Problem with Simultaneous Pickup and Delivery by a fleet of identical vehicles include: a set of customers with specific coordinates who have asked for simultaneous pickup and delivery of goods, and all the fleet are fed by one warehouse. Vehicle Routing Problems can be presented by a G= (V,E) graph, in which V={0, ..., n}is the set of nodes and E the set of arcs between the nodes. The objective function of this problem is to minimize the past distance by vehicle or the total route, presented as follows:

$$X_{ijk} = \begin{cases} 1 & If\ vehicle\ k\ goes\ from\ node\ i\ to\ node\ j\ directly (\forall i,j \in N;\ \forall k \in K) \\ 0 & Otherwise \end{cases}$$

The problem constraints are modeled as follows:

$$\min \quad \sum_{i\epsilon N}\sum_{j\epsilon N}\sum_{k\epsilon K} c_{ij}x_{ijk} \qquad (1)$$

$$s.t. \quad \sum_{i\epsilon N}\sum_{k\epsilon K} x_{ijk} = 1 \qquad \forall j\epsilon N_0 \qquad (2)$$

$$\sum_{j\epsilon N} x_{ijk} = \sum_{j\epsilon N} x_{ijk} \qquad \forall i\epsilon N, \forall k\epsilon K \qquad (3)$$

$$\sum_{j\epsilon N_0} x_{0jk} \le 1 \qquad \forall k\epsilon K \qquad (4)$$

$$U_i - U_j + |N|\sum_{k\epsilon K} x_{ijk} \le |N| - 1 \qquad \forall i,j\epsilon N_0, i \ne j \qquad (5)$$

$$1 \le U_i \le |N| \qquad \forall i\epsilon N_0 \qquad (6)$$

$$L'_k = \sum_{i\epsilon N}\sum_{j\epsilon N_0} d_j\ x_{ijk} \qquad \forall k\epsilon K \qquad (7)$$

$$L_i \ge L'_k - d_i + p_i - M(1 - x_{0ik}) \qquad \forall i\epsilon N_0, \forall k\epsilon K \qquad (8)$$

$$L_j \ge L_i - d_j + p_j - M\left(1 - \sum_{k\epsilon K} x_{ijk}\right) \qquad \forall i,j\epsilon N_0, i \ne j \qquad (9)$$

$$0 \le L'_k \le Q \qquad \forall k\epsilon K \qquad (10)$$

$$0 \le L_j \le Q \qquad \forall j\epsilon N_0 \qquad (11)$$

$$x_{ijk}\epsilon\{0,1\} \qquad \forall i,j\epsilon N, \forall k\epsilon K \qquad (12)$$

The problem parameters include: N: set of all nodes (0,…,n), N_0: set of all customers (1,…,n), K: set of vehicles (1,…,k),c_{ij}: cost (distance) of an arc (i, j) (i, jϵN), d_i: delivery demand of customer i (iϵ N_0), P_i: pickup demand of customer i (iϵ N_0), Q: vehicle capacity, M: large positive number, L'_k: load of vehicle *k* when leaving the depot (∀k∈K), L_i: load of vehicle after having serviced customer i (∀i∈N_0), U_i: variable used to prohibit sub tours (∀i∈ N_0), In this formulation, the objective function (1) is to minimize the total cost (distance). Constraints (2) ensure that each customer must be serviced exactly once. Constraints (3) imply that if a vehicle arrives at a customer, then the same vehicle must also leave. Constraint (4) guaranties that a vehicle can be used at most one route. While constraint (5) eliminates sub tours and constraint (6) bounds additional variables, constraint (7) denotes the initial load of vehicles. Constraints (8) and (9) describe the fluctuating load on vehicle after first customer and any customers, respectively. Constraints (10) and (11) state that the vehicle loads do not exceed the vehicle capacity. Finally, constraints (12) impose binary conditions on the variables[5].

2.1. THE PROPOSED ALGORITHM

Vehicle Routing Problems are complex ones in nature, categorized under NP-hard problems. Due to calculation complexity, artificial intelligence (AI) is utilized to solve such problems. In this study, genetic algorithm is applied to solve the Vehicle Routing Problem.

Genetic algorithm is a heuristic search method, inspired by biological evolution theory, which helps to improve solutions in complex problems with discrete solutions. Genetic algorithm is an optimization tool, inspired by natural evolution and Darwin's "survival of the fittest principle". This algorithm improves the solutions for optimization problems, using genetic operators such as selection, crossover and mutation. The presented solution in this study includes the three phases of creation, crossover and mutation. These 3 phases will be elaborated later. Genetic algorithm requires a data structure influenced by VRPSPD for assessment mechanism. Thus, for addressing genes, instead of binary representation, integers will be used, which represent customers. Using integers will reduce unacceptable chromosomes. An outline of the proposed algorithm's phases is presented in Figure 1. The first step is to create the preliminary population. 3 different methods are used for this matter. The next step is assessment function, and based on the assessed value in crossover, the chromosomes select their couple to reproduce. In the mutation phase, 4 methods are applied to improve the chromosomes, which will be elaborated later.

2.1.1. Creating the Initial population

The initial population is of high importance to genetic algorithm. The better the initial population, the faster genetic algorithm can reach the better generation, and the variety in producing the initial solution helps genetic algorithm to explore the whole feasible region searching for the optimum solution. In this proposed method, the first step is to form a set of good solutions as the first generation population for the genetic algorithm. Therefore, 3 methods of solution generation were used, and the reached solutions of each method consists the initial population. Considering the different structure used in each method, the initial produced solutions are varied and include all feasible regions. Also, given the changes made in these methods, the initial population is tried to be produced by quality solutions, in order to help genetic algorithm reach a superior future generation. In complex problems, heuristic methods help present good and satisfying solutions. In most of genetic algorithm methods presented to solve travelling salesman problem (TSP), the permutation of chromosomes is used as a heuristic method, in which each chromosome displays the route that vehicle must pass. This coding method enjoys higher popularity, since order and sequence is observed in it. However, in some routing problems the travelled distance by each vehicle might be divided into a number of sub-tours, due to some limitations of the problem such as the vehicle's capacity, which is called Iterated Tour Partitioning (ITP). The following is the necessary modifications for coordinating three methods of Randomized, Nearest Neighbor and Cheapest Insertion heuristic applied in generating the solution to VRPSPD.

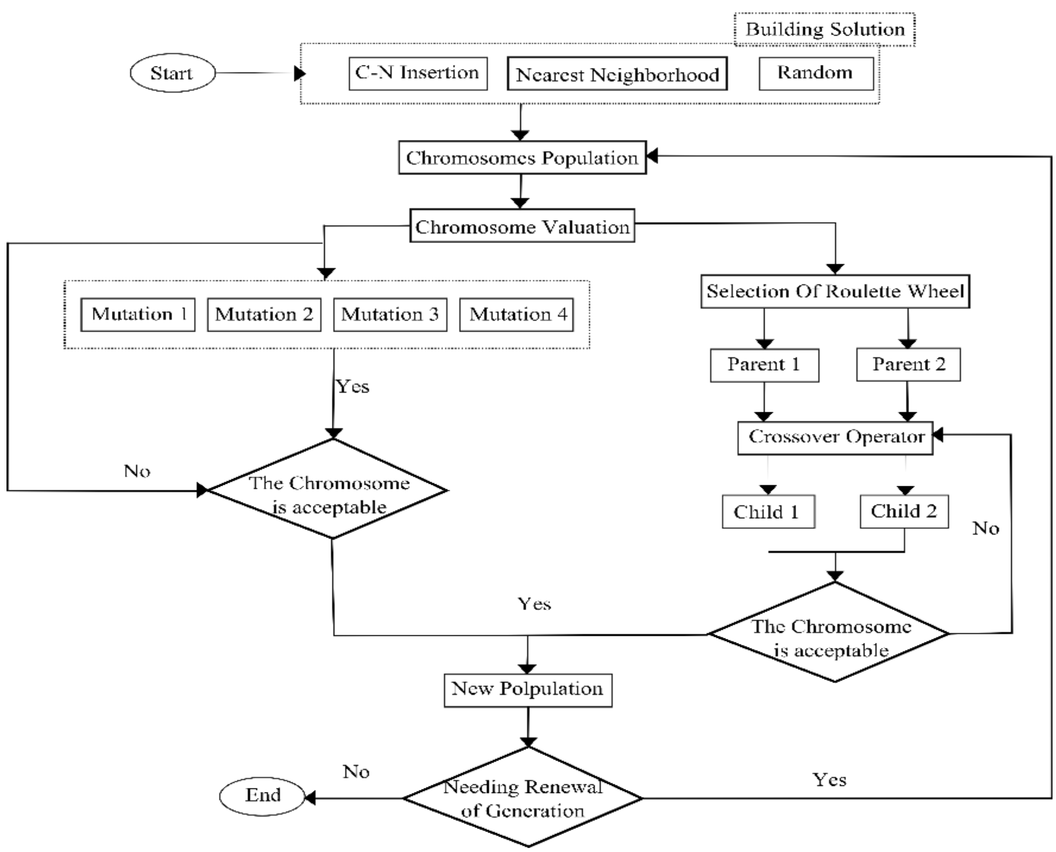


Figure 1. Proposed algorithm flowchart

2.1.2. Randomized Algorithm

In Randomized method, the points are chosen at random and the sub-tours are created respectively. To do so, the warehouse is selected as the first point; the next point is selected at random and added to the path. This goes on until the vehicle's capacity is full. Then, the next sub-tour begins. This process continues until there is no point left. In VRPSPD problems, each customer picks up and delivers some goods. Therefore, each vehicle, given its capacity, can give service to a limited number of customers. When its capacity is full, the vehicle returns the warehouse and starts a new trip. These trips continue until the vehicle has travelled through all points. Since the points are selected at random, the reached solutions are not of high quality. Acceptable solutions are always presented though, which is one of the advantages of this method.

2.1.3. Nearest Neighbor Algorithm

Nearest Neighbor Algorithm is probably the most famous heuristic for solving VRP. In this method, the sub-tours are created in the first place. For this purpose, the first customer is the nearest customer to the warehouse. Next, the nearest customer to the first customer is chosen, provided that picked up and delivered goods do not exceed the capacity of vehicle. The next customers are chosen in the same manner and the tour is completed. The process continues until the capacity of vehicle is full. At the end of the tour, the vehicle returns to warehouse and the next sub-tour begins, until the tour is completed and there is no customer left. To find the first customer after leaving the warehouse, probability function is proposed. Based on the distance between customers, this function gives a probability to each customer, so the nearest customer enjoys the highest probability to be chosen. The probability to choose each customer (fi) is reached through the following equation.

$$p_i = \left(\frac{1}{d_i}\right)^{2.5} \tag{13}$$

$$f_i = \frac{p_i}{\sum p_i} \tag{14}$$

d_i: distance of customer i to warehouse

Figure 2 shows the nearest neighbor algorithm.

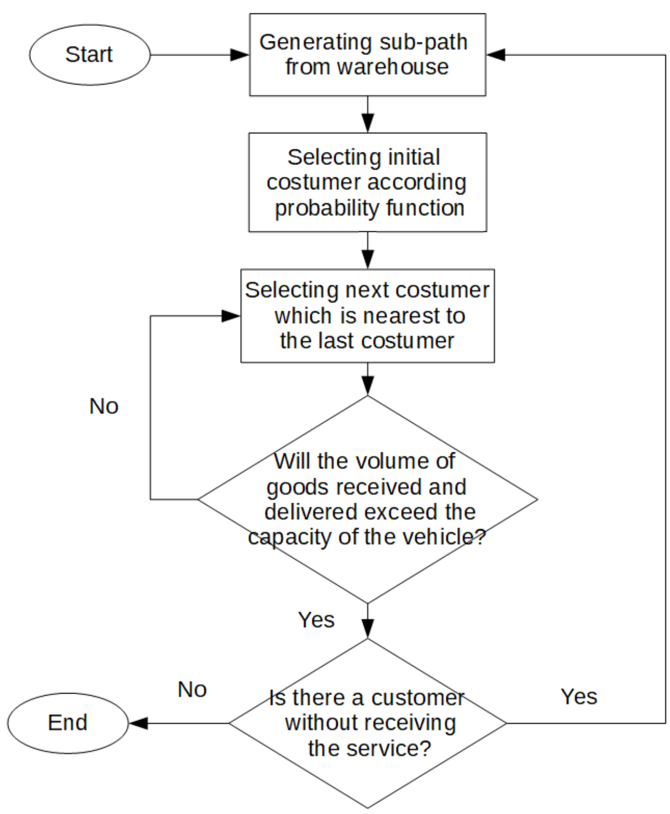


Figure 2. Nearest neighbor algorithm

2.1.4. Cheapest Insertion Algorithm

In the Insertion method, in order to create a tour, first an initial tour is created, then a link from the tour is chosen at random, the nearest customer to the link is specified and added to the tour. This process continues until service is provided to all customers. Like the previous methods, sub-tours are created. Then the whole path is created by combining the sub-tours. In this study, creating the initial tour is proposed as follows. The initial tour is created as d_0 -› n_i -› d_0, d_0 being the warehouse and ni the customer chosen based on probability function. According to the distance between customers and warehouse, this function gives probability to each customer. So the furthest customer has the highest chance to be selected. The probability of each customer to be chosen (f_i) is reached through this equation.

$$p_i = (d_i)^{2.5} \quad (15)$$

$$f_i = \frac{p_i}{\sum p_i} \quad (16)$$

d_i: the distance between customer i and warehouse

After creating the initial route, one link is selected at random and the nearest customer to the link is specified. This customer is added to the initial rout, so that it creates the shortest length. For instance, if we consider the selected link as T:{ i , j}, customer k is selected so that c(T,k) value is minimal.

$$c(T,k) = c(i,k) + c(k,j) - c(i,j) \quad (17)$$

Next customer is selected in the same manner, until the vehicle is full. When the sub-tour is completed, the next sub-tour is created for unserved customers, until all customers are served once. Figure 3 shows the explained.

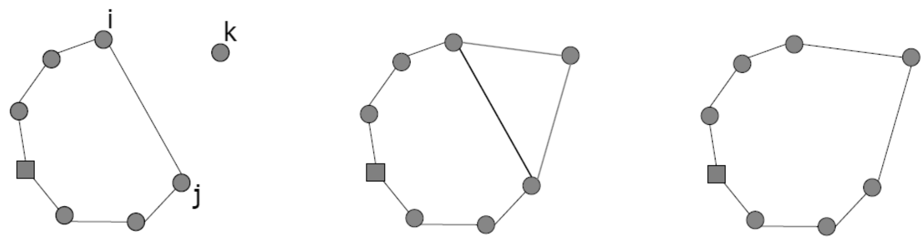


Figure 3. The procedure of adding nearest customer to the path

2.1.5. Fitness Function

This function gives each chromosome a fitness value. In the proposed algorithm, instead of using the fitness function as the objective function, the total traveling distance (tour length) is used. Therefore, the less the fitness value of chromosomes, the higher value they have.

2.1.6. Selection of Roulette Wheel

Chromosomes are selected for reproduction based on fitness function using roulette wheel. In this method, chromosomes are computed using fitness function. So if f_k is the fitness function of the k^{th} chromosome, the corresponding survival probability would be:

$$p_k = f_k \Big/ \sum_{i=1}^{n} f_i \qquad \text{, n=pop size} \quad (18)$$

Now chromosomes are arranged according to p_k, and q_k, which is the cumulative value of p_k, is reached as follows:

$$q_k = \sum_{i}^{k} p_i \quad (19)$$

In this method, first a random number between 1 and 0 is chosen to select the chromosome. Then the chromosome is selected based on the interval in which the randomly chosen number is placed. In the proposed method, each chromosome represents a tour. Therefore, the value of each chromosome is calculated based on its corresponding path [28]. Figure 4 shows the pseudo code of roulette wheel selection.

```
For i to PopSize
        fi = Value of each chromosome from Fitness function
End

For k to PopSize
        Pk = fk – sum(f)
End

For k to NumberofCrossOver
        x=rand()  //Return random number uniformly distributed in [0,1]
        pt = qk - x
        select(k)=pt
End
```

Figure 4. The pseudo code of roulette wheel selection

2.1.7. Crossover Operator

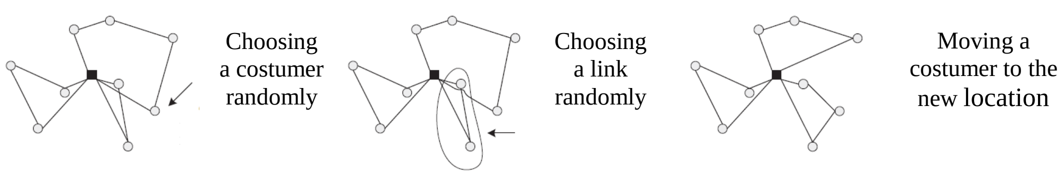
This operator considers two chromosomes as parents and exchanges specific genes, and as a result, new children are produced. The purpose of this operator is to create better generation from a set of the best of previous generation. In fact, crossover operator searches the feasible regions keeping the existing information in chromosomes.

2.1.8. Sequential Method

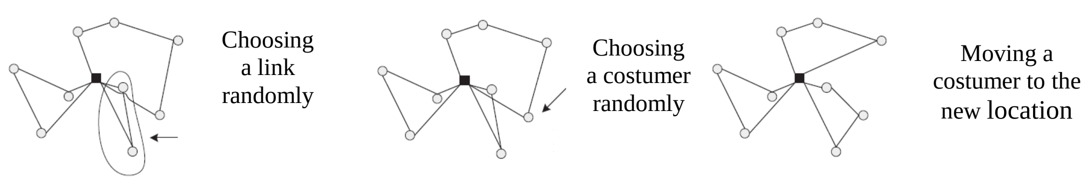
In this method, first introduced by Davis (1985), two random numbers are selected as cut points. Then, the genes between these two numbers are kept unchanged in parent chromosomes and the rest of the genes are switched [29].

2.1.9. Mutation Operators

Mutation through creating random changes in chromosomes' genes helps improving the generation. Mutation helps searching the intact area, and the most important objective of causing mutation is to prevent convergence to locally optimal solution. Here, 4 mutation operators are introduced to create new features in the current generation. These operators allow genetic algorithm to search the new area and help to improve chromosomes. *Mutation type 1:* in this operator, one customer is selected at random. Then, a link is chosen at random, and the customer is moved to a new location and the new tour is created. (Figure 5) *Mutation type 2:* in this mutation, first one link is selected at random. Then the customer is selected so as to be able to be located on the link. (Figure 5) *Mutation type 3:* this operator is similar to operator type 1, except that two customers are selected consecutively instead of one. (Figure 5) *Mutation type 4:* in this mutation, two sub tours are selected randomly. Then one customer is selected from each of the sub tours. In the end, the customers are exchanged and a new tour is created. (Figure 5)



Mutation type 1



Mutation type 2

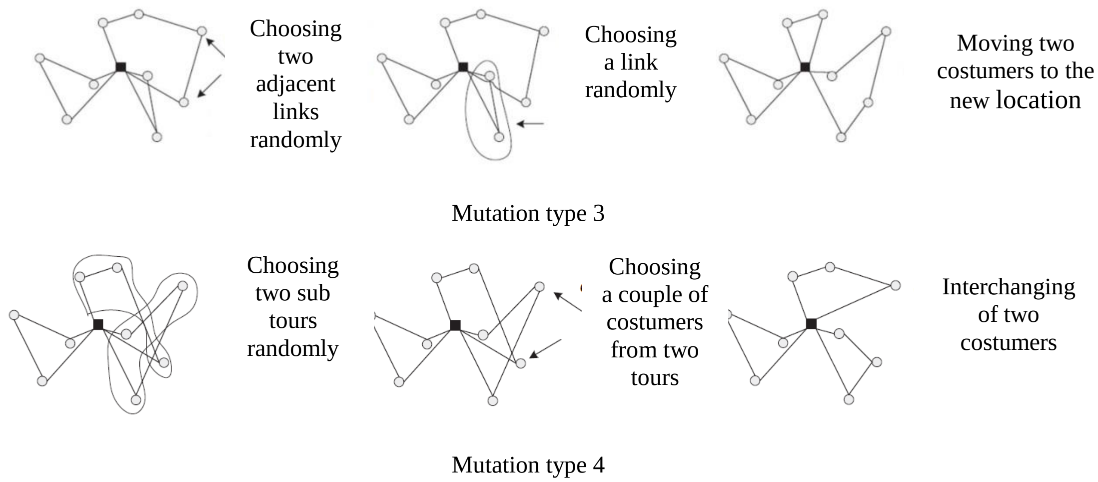


Figure 5. Mutation Types

3. RESULTS AND DISCUSSION

In this study, Dethloff (2001) standard examples are used to test the proposed algorithm. These examples include 50 customers and two scenarios based on location. In SCA scenario, customers' coordinates are distributed evenly in 0-100 interval. In CON scenario, half of the customers are distributed the same way as SCA scenario, and the other half are distributed evenly in 100/3-200/3 interval. The delivered goods value (D_j) by customers is distributed evenly in 0-100 interval. The picked up goods value (P_j) is reached from the following equivalent using a random number (R_j) from 0 to 1.

$$P_j = (0.5 + r_j)D_j \tag{20}$$

Examples are developed for different vehicle capacities with the minimum vehicle (μ). The capacity for these examples can be reached through the following equivalent. (μ is chosen as 3 or 8).

$$C = \sum_{s\epsilon J} D_s/\mu \tag{21}$$

In order to run the algorithm, MATLAB software and a PC with Intel Core™ i3 (2.27GHZ) processor and 4GB RAM is used. The results of the proposed algorithm on Dethloff's examples using the world's best methods until 2019 are presented in table 1 & 2. In these tables, by the number of nodes the number of customers, and by μ, vehicles is meant. On the second column, the results reached by Dethloff [5] are presented. As the number of vehicles is not known, its value is not presented. Also, on the fourth column, the results of Tang and Galvao [9], on the sixth column the results of Catay [13] and on the eighth column the results of Zachariadis [14] are presented. The last results are from artificial bee colony algorithm [30].

In tables below, green color means the result of proposed genetic algorithm is better or equal to that of the mentioned algorithm. For example in problem SCA3-1, there is no result better than the proposed algorithm. It is obvious that no algorithm can yield best solution, thus the results proposed genetic algorithm are better than all mentioned algorithm in some problems. For a better view, figure 6 & 7 depict the results of table 1 & 2.

Table 1. Comparison between the proposed algorithm and other meta-heuristic methods- Dethloff problems, SCA scenario

Problem	*Nods*	*D 2001* [5]	μ	*T,G 2006* [9]	μ	*C 2010* [13]	μ	*Z,K 2011* [14]	μ	*S,E 2019* [30]	μ	*Proposed Genetic Algorithm*	μ
SCA3-0	50	689	-	640.55	4	636.1	4	636.06	4	640.55	4	*640.55*	4
SCA3-1	50	765.6	-	697.84	4	700.1	4	697.84	4	697.84	4	*697.84*	4
SCA3-2	50	742.8	-	659.34	4	659.3	4	659.34	4	659.30	4	*665.71*	4
SCA3-3	50	737.2	-	680.04	4	680	4	680.04	4	683.11	4	*684.1*	4

SCA3-4	50	747.1	-	690.5	4	690.5	4	690.5	4	692.57	4	692.57	4
SCA3-5	50	784.4	-	659.9	4	670.1	4	659.9	4	659.90	4	661.07	4
SCA3-6	50	720.4	-	653.81	4	651.1	4	651.09	4	651.09	4	654.47	4
SCA3-7	50	707.9	-	659.17	4	666.1	4	659.17	4	666.54	4	666.54	4
SCA3-8	50	807.2	-	719.47	4	719.5	4	719.47	4	723.44	4	720.57	4
SCA3-9	50	764.1	-	681	4	681	4	681	4	685.16	4	684.66	4
SCA8-0	50	1132.9	-	981.47	9	961.6	9	961.5	9	961.50	9	963.02	9
SCA8-1	50	1150.9	-	1077.44	9	1063	9	1050.2	9	1060.63	9	1069.1	9
SCA8-2	50	1100.8	-	1050.98	10	1040.6	9	1039.64	9	1045.12	9	1059.1	9
SCA8-3	50	1115.6	-	983.34	9	985.9	9	983.34	9	983.34	9	997.75	9
SCA8-4	50	1235.4	-	1073.46	9	1071	9	1065.49	9	1072.39	9	1079.1	9
SCA8-5	50	1231.6	-	1047.24	9	1054.3	9	1027.08	9	1027.08	9	1047.24	9
SCA8-6	50	1062.5	-	995.59	9	972.5	9	971.82	9	980.71	9	997.7	9
SCA8-7	50	1217.4	-	1068.56	10	1059.7	9	1052.17	9	1059.28	9	1119.3	9
SCA8-8	50	1231.6	-	1080.58	9	1082.7	9	1071.18	9	1080.02	9	1093.5	9
SCA8-9	50	1185.6	-	1084.8	9	1081.4	9	1060.5	9	1060.50	9	1066.1	9

Table 2. Comparison between the proposed algorithm and other meta-heuristic methods- Dethloff problems, CON scenario

Problem	Nods	D 2001 [5]	μ	T,G 2006 [9]	μ	C 2010 [13]	μ	Z,K 2011 [11]	μ	S,E 2019 [30]	μ	*Proposed Genetic Algorithm*	μ
CON3-0	50	672.4	-	631.39	4	616.50	4	616.52	4	616.50	4	621.22	4
CON3-1	50	570.6	-	554.47	4	555.6	4	554.47	4	554.47	4	566.4	4
CON3-2	50	534.8	-	522.86	4	521.4	4	519.26	4	523.47	4	524.14	4
CON3-3	50	656.9	-	591.19	4	591.2	4	591.19	4	595.46	4	608.01	4
CON3-4	50	640.2	-	591.12	4	589.3	4	589.32	4	591.37	4	600	4
CON3-5	50	604.7	-	563.7	4	563.7	4	563.7	4	563.70	4	570.64	4
CON3-6	50	521.3	-	506.19	4	499.2	4	500.8	4	502.63	4	504.97	4
CON3-7	50	602.8	-	577.68	4	577.5	4	576.48	4	580.87	4	582.15	4
CON3-8	50	556.2	-	523.05	4	523.1	4	523.05	4	523.94	4	534.47	4
CON3-9	50	612.8	-	580.05	4	578.2	4	580.05	4	578.25	4	591.63	4
CON8-0	50	967.3	-	860.48	9	858.9	9	857.17	9	864.52	9	873.96	9
CON8-1	50	828.7	-	740.85	9	740.9	9	740.85	9	745.91	9	736.78	9
CON8-2	50	770.2	-	723.32	9	714.3	9	713.44	9	712.89	9	717.05	9
CON8-3	50	906.7	-	811.23	10	812.3	10	811.07	10	816.38	10	812.88	10
CON8-4	50	876.8	-	772.25	9	770.1	9	772.25	9	774.90	9	779	9

CON8-5	50	866.9	-	756.91	9	766.6	9	756.91	9	758.33	9	*751.55*	9
CON8-6	50	749.1	-	678.92	9	697.2	9	678.92	9	683.21	9	*675.16*	9
CON8-7	50	929.8	-	814.5	9	814.8	9	811.96	9	811.96	9	*820.8*	9
CON8-8	50	833.1	-	775.59	9	771.3	9	767.53	9	771.19	9	*776.67*	9
CON8-9	50	877.3	-	809	9	815.1	9	809	9	809.00	9	*812.55*	9

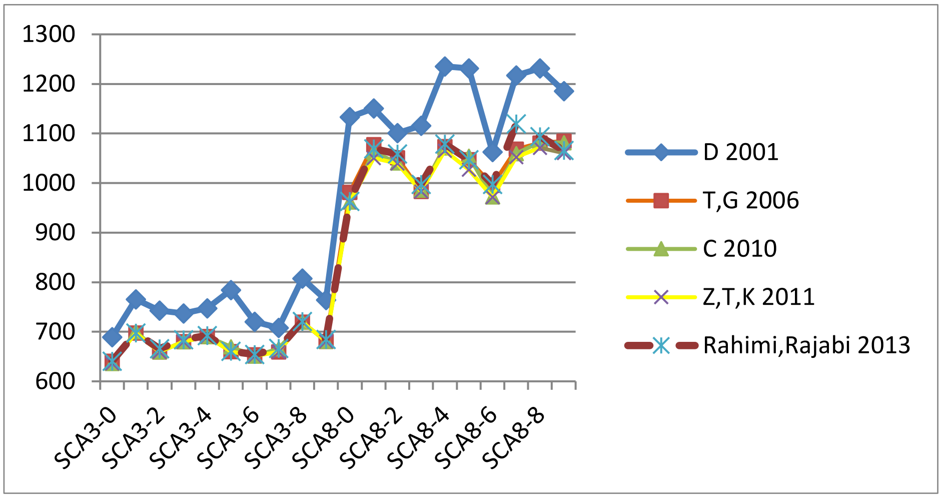


Figure 6. Graph comparing the results of the proposed algorithm and other algorithms for SCA scenario

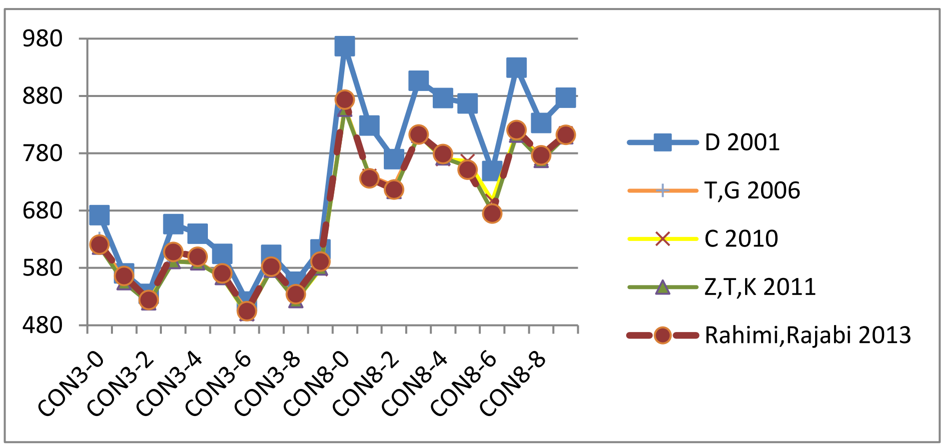


Figure 7. Graph comparing the results of the proposed algorithm and other algorithms for CON scenario

It is obvious that by increasing the number of points and difficulty of the problems, the error of heuristic methods increases. As can be seen, the error of proposed algorithm is ascending, as well as other algorithms. Table 3 summarizes tables 1 & 2, here green color indicates that the result of proposed genetic algorithm were better than best solution by mentioned algorithms. It means that the proposed algorithm could improve the solution and provided a new better one with lower cost. The negative gap percentage shows that how much the proposed genetic algorithm improved the current solution and vice versa. The formula for calculating gap is as follows (eq. 25):

$$Gap\% = [(Proposed\ solution - Best\ solution)/(\frac{Proposed\ solution + Best\ solution}{2})] * 100 \qquad (25)$$

Table 3. The best possible solutions for Dethloff problems

Problem	Best Solution	Proposed Genetic Algorithm	Gap%	Problem	Best Solution	Proposed Genetic Algorithm	Gap%
SCA3-0	636.06	640.55	0.70	CON3-0	616.52	621.22	0.76
SCA3-1	697.84	697.84	0.00	CON3-1	554.47	566.4	2.13
SCA3-2	659.3	665.71	0.97	CON3-2	518	524.14	0.93
SCA3-3	680	684.1	0.60	CON3-3	591.19	608.01	2.80
SCA3-4	690.5	692.57	0.30	CON3-4	588.79	600	1.80
SCA3-5	659.9	661.07	0.18	CON3-5	563.7	570.64	1.22
SCA3-6	651.09	654.47	0.52	CON3-6	499.05	504.97	1.15
SCA3-7	659.17	666.6	1.11	CON3-7	576.48	582.15	0.98
SCA3-8	719.47	720.57	0.15	CON3-8	523.05	534.47	2.16
SCA3-9	681	684.66	0.53	CON3-9	578.25	591.63	2.30
SCA8-0	961.5	963.02	0.16	CON8-0	857.17	873.96	1.93
SCA8-1	1050.2	1069.1	1.77	CON8-1	740.85	736.78	***-0.55***
SCA8-2	1039.64	1059.1	1.85	CON8-2	712.89	717.05	0.58
SCA8-3	983.34	997.75	1.45	CON8-3	811.07	812.88	0.22
SCA8-4	1065.49	1079.1	1.27	CON8-4	772.25	779	1.15
SCA8-5	1027.08	1047.2	1.94	CON8-5	754.88	751.55	***-0.71***
SCA8-6	971.82	997.7	2.62	CON8-6	678.92	675.16	***-0.55***
SCA8-7	1052.17	1119.3	6.16	CON8-7	811.96	820.8	1.08
SCA8-8	1071.18	1093.5	2.05	CON8-8	767.53	776.67	1.18
SCA8-9	1060.5	1066.1	0.53	CON8-9	809	812.55	0.44
Average of differences %			1.24	Average of differences %			1.05

Although the proposed genetic algorithm ran by a weaker CPU and programming language, in comparison to other researches such as Simsir and Ekmichi [30], it was able to improve the solutions for three problems and As randomness is one of the main parts of meta-heuristic algorithms, it is not logical to expect the proposed algorithm could improve all solutions, but its average gap with best solutions was 1.15%, which can be acceptable. The heuristic algorithms are based on random selections, so there should be no specific pattern or trend in the accuracy of solutions. In other words, the proposed algorithm should not provide a better solution than other algorithms in a specific part of the problems. The comparison of the proposed algorithm with other algorithms shows that the algorithm has performed better than other algorithms in some problems randomly. Therefore, the proposed algorithm cannot be dedicated to a specific area of problems and the random section is working properly. On the other hand, it is observed that the differences between the proposed algorithm and the best available solution, in SCA and CON problems, are not significantly different, less than 5%. It indicates that the proposed algorithm has the same efficiency in different kinds of problems.

4. CONCLUSION

In this article, to solve Vehicle routing problem with simultaneous pickup and delivery, hybrid meta-heuristic is presented, in which a genetic algorithm with new features is applied. In this genetic algorithm, three Randomized, Nearest Neighbor and Cheapest Insertion algorithms were combined and applied. Next, the reached results were used for genetic algorithm initial population. In each of these algorithms, heuristics were used to reach a better solution, one crossover operator and 4 mutation operators were introduced, which make changes and mutations in populations randomly and simultaneously. In order to evaluate and validate the proposed algorithm, Dethloff's standard examples introduced to solve VRPSPD were used. The proposed algorithm was solved for all problems and optimal solutions were reached. The reached results from other algorithms to solve Dethloff's problems were gathered for the purpose of comparison. The proposed algorithm is the first meta-heuristic genetic algorithm presented to solve Dethloff's standard problems. The proposed algorithm has been able to find a better solution for 3 of the examples than other presented algorithms. The graph in figures (3) and (4) indicate how close the results by the proposed algorithm are to those of other algorithms. The gap between the best existing solutions is displayed in table (3). Comparing the results with the best existing solutions, we will observe that the proposed algorithm in 3 examples, CON8-1, CON8-5, CON8-6, yielded 0.55, 0.71 and 0.55 percent better solutions respectively than the best existing solutions.

FUNDING/SUPPORT

Not mentioned any Funding/Support by authors.

ACKNOWLEDGMENT

Not mentioned by authors.

AUTHORS CONTRIBUTION

This work was carried out in collaboration among all authors.

CONFLICT OF INTEREST

The author (s) declared no potential conflicts of interests with respect to the authorship and/or publication of this paper.

5. REFERENCES

[1] Min H. The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transportation Research Part A: General. 1989 Sep 1;23(5):377-86. [View at Google Scholar] ; [View at Publisher].

[2] Fu C, Zhang L, Wang X, Qiao L. Solving TSP problem with improved genetic algorithm. InAIP Conference Proceedings 2018 May 23 (Vol. 1967, No. 1, p. 040057). AIP Publishing LLC. [View at Google Scholar] ; [View at Publisher].

[3] Hussain A, Muhammad YS, Nauman Sajid M, Hussain I, Mohamd Shoukry A, Gani S. Genetic algorithm for traveling salesman problem with modified cycle crossover operator. Computational intelligence and neuroscience. 2017 Oct 25;2017. [View at Google Scholar] ; [View at Publisher].

[4] Salhi S, Nagy G. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. Journal of the operational Research Society. 1999 Oct 1;50(10):1034-42. [View at Google Scholar] ; [View at Publisher].

[5] Dethloff J. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. OR-Spektrum. 2001 Feb 1;23(1):79-96. [View at Google Scholar] ; [View at Publisher].

[6] Montané FA, Galvão RD. Vehicle routing problems with simultaneous pick-up and delivery service. Opsearch. 2002 Feb 1;39(1):19-33. [View at Google Scholar] ; [View at Publisher].

[7] Nagy G, Salhi S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. European journal of operational research. 2005 Apr 1;162(1):126-41. [View at Google Scholar] ; [View at Publisher].

[8] Chen JF, Wu TH. Vehicle routing problem with simultaneous deliveries and pickups. Journal of the Operational Research Society. 2006 May 1;57(5):579-87. [View at Google Scholar] ; [View at Publisher].

[9] Montané FA, Galvao RD. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Computers & Operations Research. 2006 Mar 1;33(3):595-619. [View at Google Scholar] ; [View at Publisher].

[10] Bianchessi N, Righini G. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. Computers & Operations Research. 2007 Feb 1;34(2):578-94. [View at Google Scholar] ; [View at Publisher].

[11] Zachariadis EE, Tarantilis CD, Kiranoudis CT. A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. Expert Systems with

applications. 2009 Mar 1;36(2):1070-81. [View at Google Scholar] ; [View at Publisher].

[12] Gajpal Y, Abad P. An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. Computers & Operations Research. 2009 Dec 1;36(12):3215-23. [View at Google Scholar] ; [View at Publisher].

[13] Çatay B. A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. Expert Systems with Applications. 2010 Oct 1;37(10):6809-17. [View at Google Scholar] ; [View at Publisher].

[14] Zachariadis EE, Kiranoudis CT. A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. Expert Systems with Applications. 2011 Mar 1;38(3):2717-26. [View at Google Scholar] ; [View at Publisher].

[15] Mukhopadhyay DM, Balitanas MO, Farkhod A, Jeon SH, Bhattacharyya D. Genetic algorithm: A tutorial review. International journal of grid and distributed computing. 2009 Sep 1;2(3):25-32. [View at Google Scholar] ; [View at Publisher].

[16] Mirjalili S, Dong JS, Sadiq AS, Faris H. Genetic algorithm: Theory, literature review, and application in image reconstruction. InNature-Inspired Optimizers 2020 (pp. 69-85). Springer, Cham. [View at Google Scholar] ; [View at Publisher].

[17] Donis-Díaz CA, Bello R, Kacprzyk J. Using ant colony optimization and genetic algorithms for the linguistic summarization of creep data. InIntelligent Systems' 2014 2015 (pp. 81-92). Springer, Cham. [View at Google Scholar] ; [View at Publisher].

[18] Schroeders U, Wilhelm O, Olaru G. Meta-heuristics in short scale construction: Ant colony optimization and genetic algorithm. PLoS One. 2016 Nov 28;11(11):e0167110. [View at Google Scholar] ; [View at Publisher].

[19] Zhao F, Yao Z, Luan J, Song X. A novel fused optimization algorithm of genetic algorithm and ant colony optimization. Mathematical Problems in Engineering. 2016 Jan 1;2016. [View at Google Scholar] ; [View at Publisher].

[20] Asadi AA, Naserasadi A, Asadi ZA. A new hybrid algorithm for traveler salesman problem based on genetic algorithms and artificial neural networks. International Journal of Computer Applications. 2011;975:8887. [View at Google Scholar] ; [View at Publisher].

[21] Kartheeswaran S, Durairaj DD. A hybrid genetic algorithm and back-propagation artificial neural network based simulation system for medical image reconstruction in noise-added magnetic resonance imaging data. In2015 Online International Conference on Green Engineering and Technologies (IC-GET) 2015 Nov 27 (pp. 1-6). IEEE. [View at Google Scholar] ; [View at Publisher].

[22] Fatyanosa TN, Sihananto AN, Alfarisy GA, Burhan MS, Mahmudy WF. Hybrid genetic algorithm and simulated annealing for function optimization. Journal of Information Technology and Computer Science. 2017 Feb 8;1(2):82-97. [View at Google Scholar] ; [View at Publisher].

[23] Chen PH, Shahandashti SM. Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. Automation in Construction. 2009 Jul 1;18(4):434-43. [View at Google Scholar] ; [View at Publisher].

[24] Pomar LA, Pulido EC, Roa JD. A Hybrid Genetic Algorithm and Particle Swarm Optimization for Flow Shop Scheduling Problems. InWorkshop on Engineering Applications 2017 Sep 27 (pp. 601-612). Springer, Cham. [View at Google Scholar] ; [View at Publisher].

[25] Kuo RJ, Han YS. A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem–A case study on supply chain model. Applied Mathematical Modelling. 2011 Aug 1;35(8):3905-17. [View at Google Scholar] ; [View at Publisher].

[26] Cherki I, Chaker A, Djidar Z, Khalfallah N, Benzergua F. A Sequential Hybridization of Genetic Algorithm and Particle Swarm Optimization for the Optimal Reactive Power Flow. Sustainability. 2019 Jan;11(14):3862. [View at Google Scholar] ; [View at Publisher].

[27] DengWei L. Optimization design based on hierarchic genetic algorithm and particles swarm algorithm. Journal of Algorithms & Computational Technology. 2018 Sep;12(3):217-22. [View at Google Scholar] ; [View at Publisher].

[28] Melanie M. An introduction to genetic algorithms. Cambridge, Massachusetts London, England, Fifth Print 1999; 3: 62–75. [View at Publisher].

[29] Davis L. Applying adaptive algorithms to epistatic domains. InIJCAI 1985 Aug 18 (Vol. 85, pp. 162-164). [View at Google Scholar] ; [View at Publisher].

[30] Simsir F, Ekmekci D. A metaheuristic solution approach to capacitied vehicle routing and network optimization. Engineering Science and Technology, an International Journal. 2019 Jun 1;22(3):727-35. [View at Google Scholar] ; [View at Publisher].