TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 1217 – 1235
© TÜBİTAK
doi:10.3906/elk-1902-122

A population based simulated annealing algorithm for capacitated vehicle routing problem

İlhan İLHAN*
Department of Mechatronic Engineering, Faculty of Engineering, Necmettin Erbakan University, Konya, Turkey

Received: 18.02.2019 • **Accepted/Published Online:** 06.03.2020 • **Final Version:** 08.05.2020

Abstract: The Vehicle Routing Problem (VRP) is one of the most discussed and researched topics nowadays. The VRP is briefly defined as the problem of identifying the best route to reduce distribution costs and improve the quality of service provided to customers. The Capacitated VRP (CVRP) is one of the most commonly researched among the VRP types. Therefore, the CVRP was studied in this paper and a new population based simulated annealing algorithm was proposed. In the algorithm, three different route development operators were used, which are exchange, insertion and reversion operators. It was tested on 63 well-known benchmark instances in the literature. The results showed that the optimum routes could be determined for the 23 instances.

Key words: Capacitated vehicle routing problem, best route, route enhancing, simulated annealing

1. Introduction

The Vehicle Routing Problem (VRP) was first described by Dantzig and Ramser [1]. According to their definition, a vehicle fleet with the same or different capacities in a central depot is requested to serve a set of customers, each with a different location and different demands. In this process, it is aimed to determine the most suitable route in terms of total travel distance, duration and cost. From the date of its first definition, the new constraints have been added to the VRP and the different types have been introduced. The Capacitated VRP (CVRP) is one of them and the numerous researches have been carried out on it up to now [2].

Exact methods were used by some researchers to solve CVRP [3]. These methods include algorithms such as branch and bound [4], branch and cut [5], branch and price [6]. They use a divide and conquer strategy to separate the solution space into subproblems and then optimize each subproblem individually. The CVRP is an NP-hard problem. It has high computational complexity. The time consumption of exact methods is very high. Therefore, they cannot be applied to large-scale CVRP problems [3]. Also, they cannot consistently solve the CVRP problems with more than 50 demands [7]. When the scale of the CVRP problem is too large, heuristic and in particular metaheuristic methods are used more often than exact methods. Metaheuristic methods have two categories as single solution algorithms and population based algorithms [3]. The first includes algorithms such as greedy heuristic [8], simulated annealing [9], tabu search [10]. The second contains algorithms such as genetic algorithms, ant colony optimization, artificial bee colony, artificial immune systems, differential evolution algorithms. More and more studies are currently conducted on population based metaheuristic algorithms [3].

*Correspondence: ilhan@erbakan.edu.tr

In particular, those using new local search strategies and operators produce very good results in a reasonable time [11].

Many studies were conducted using metaheuristic algorithms to solve the CVRP [12–19]. In the first study, an improved ant colony optimization method with a new strategy and mutation process was proposed to update the increased pheromone [12]. The method used the ant-weighted strategy. It was tested on 14 different benchmark instances. The results were compared with those of other methods. In the second study, a new method based on simulated annealing algorithm was developed [13]. The method included the combination of random and deterministic operators based on problem knowledge. 24 different benchmark instances were used to test the method. In another study, a novel algorithm named the Optimized Crossover Genetic Algorithm (OCGA) was proposed using genetic algorithm with the optimized crossover operator [14]. In the algorithm, first, two parents were selected. Then, two children were produced by a mechanism designed by a non-directional binary graph. The OCGA was tested using common benchmark instances. A different study was conducted by Zhang et al. [15] using the artificial bee colony algorithm which is one of common heuristic methods. The method was named the Routing Directed Artificial Bee Colony (RABC). Many improvements were made on diversified and concentrated search capability of the traditional artificial bee colony algorithm. Teoh et al. [16] proposed a new method called the Improved Differential Evolution Algorithm with Local Search (DELS). The differential evolution algorithm was used in the method. A new local search procedure was developed to discover the new search regions and to improve the solutions found. The test of both RABC [15] and DELS [16] was performed using benchmark instances. Ewbank et al. [17] used an unsupervised fuzzy clustering method to solve the CVRP. Their method was tested on 85 benchmark instances. The different fuzziness parameter (m) values were used for each instance. Thus, the relationship between customers' demands and distances to the depot was shown with this parameter. It was shown that the optimum values could be reached with an average error of 5% in terms of total travel distance. Mohammed et al. [18] proposed a new method using the K-Nearest Neighbor Algorithm (KNNA). The method was designed not to require a large database to record the population. Thus, the running time of the algorithm was shortened. 20 benchmark instances were used to test the algorithm. The running times ranging from 0.85 seconds to 8.27 seconds were obtained. Faiz et al. [19] developed a new method called the Perturbation Based Variable Neighborhood Search with Adaptive Selection Mechanism (PVNS-ASM). In the method, the Perturbation Based Variable Neighborhood Search approach was combined with Adaptive Selection Mechanism. The perturbation scheme selection method was preferred instead of stochastic selection method. The PVNS-ASM [19] was tested on 21 benchmark instances and achieved more successful results in comparison with other methods.

In this study, a population based metaheuristic algorithm was proposed to solve the CVRP, inspired from the studies mentioned above. The simulated annealing algorithm was used as the metaheuristic algorithm because of its greater ability to prevent trapping to the local minimum in comparison with other methods. Three different local search operators were used to develop routes. The operators are exchange, insertion, and reversion operators. It was decided randomly which of them would be used in any iteration. Thus, a three-probability stochastic approach was adopted during the creation of a new solution about the current solution. A stochastic approach was also used to determine the number of points in route development operators. The proposed algorithm was tested on 63 benchmark instances with different numbers of demand points and vehicles, which are the well-known instances in the literature. The results were recorded together with the running times for each instance.

2. Capacitated Vehicle Routing Problem

In the Travelling Salesman Problem (TSP), a salesman or vehicle is requested to start from a certain location, to visit only once to all other locations in the system and to return to the starting location again [20]. Also, the total tour distance is aimed to be minimum. In fact, the VRP is a complex variant of the Multiple Travelling Salesman Problem (m-TSP), which includes multiple salesmen. In the m-TSP, each salesman needs to create a different route. In addition, the VRP was obtained by adding the constraints that the salesmen could carry a certain amount of load and the demand on each point could be different [21]. The VRP is divided into different subgroups depending on whether it includes one or more of the constraints such as environment and route status, time, cost, capacity of vehicles [22, 23]. One of the subgroups is the CVRP. Its mathematical model is given below [24].

The decision variable:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle k travels from customer i to j} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The objective function:

$$Minimize \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijk} \tag{2}$$

Subject to:

$$\sum_{k \in V} \sum_{i \in N} x_{ijk} = 1, \qquad \forall j \in N \tag{3}$$

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \qquad \forall i \in N \tag{4}$$

$$\sum_{i \in N} m_i \sum_{j \in N} x_{ijk} \leq q, \qquad \forall k \in V \tag{5}$$

$$\sum_{j \in N} x_{0jk} = 1, \qquad \forall k \in V \tag{6}$$

$$\sum_{i \in N} x_{i0k} = 1, \qquad \forall k \in V \tag{7}$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \qquad \forall h \in N, \forall k \in V \tag{8}$$

$$x_{ijk} \in \{0, 1\}, \qquad i \neq j, \forall i, j \in N, \forall k \in V \tag{9}$$

The equations were obtained for a system with V vehicles with equal capacity and N demand points. m_i refers to the demand of point i. x_{ijk} is the binary decision variable and given in (1). It equals to 1 if vehicle k travels from customer i to j and 0 otherwise. The objective function of the model is given in (2). It is tried to be minimized of the d_{ij} (distance from point i to j) costs of the connections between vehicles k

and (i, j). The provision of service by exactly one vehicle to each point is provided by (3) and (4). The routes should not exceed the vehicle capacity value q. This situation is included in the model with (5). Establishing of both incoming and outgoing connection to the depot for each vehicle is provided by (6) and (7). The constraint regulating the flow is expressed with (8). The vehicle coming to any point of demand should leave to another point of demand. A variable only takes the integer 0 or 1. This constraint is provided by (9).

In the mathematical model of the CVRP given in [24], there is a constraint given by the equation $\sum_{i\in N}\sum_{j\in N} d_{ij}x_{ijk} \leq D_k$, where $\forall k \in V$ and D_k is the maximum allowed travel distance by vehicle k. The constraint shows the limit of the total length of each route. It was not used in the methods KNNA [18] and PVNS-ASM [19]. It was not also considered in Ewbank et al.'s method [17]. Since the results of the proposed method were compared with the results of these methods, the constraint showing the limit of the total length of a route was also not taken into consideration in this study. Therefore, it was not included in the mathematical model given above.

3. Method

3.1. The population based simulated annealing algorithm

The simulated annealing algorithm is a random search algorithm proposed by Kirkpatrick et al. [9]. It was developed based on the similarity between the metallurgical annealing process and the research of the minimum value in a more general system. The simulated annealing algorithm starts the research with a single randomly generated solution. Therefore, its performance largely depends on the starting point. If the quality of the starting point is poor, the result is unsatisfactory. On the other hand, it is not appropriate to use a single solution to explore the entire research space [25]. It also requires a long calculation time to find a reasonable solution. These drawbacks are revealed when the problem dimension is high and there are many local minima [25]. In order to overcome them, a population based simulated annealing algorithm was proposed and used to solve the CVRP. The proposed algorithm does not contain memory, unlike the population based simulated annealing (PSA) proposed in [25]. The PSA algorithm generates the new candidate solution using the current solution and its two interesting elite experiences. In the algorithm proposed in this study, the new candidate solutions (routes) are generated by using exchange, insertion, and reversion operators. It was decided randomly which of them would be used in any iteration.

At the beginning of the proposed algorithm, a start temperature T_{start} and an end temperature T_{end} are determined and an initial population x containing random solutions is generated. The population size should be given as input to the algorithm. The fitness values ($f(x)$) of the solutions constituting the population are calculated by using the objective function. The objective function for the CVRP is given in (2). It represents the sum of the distances covered. In Figure 1, the calculation of the distance covered (fitness value) is shown on an instance in detail. Figure 1a gives information belonging to the instance consisting of twelve customers and one depot. The distances between the customers and the depot are calculated as in Figure 1b. A randomly generated sample solution is shown in Figure 1c. According to the solution, the route information of the vehicle 1 is {0, 12, 1, 7, 0}. For this route, the distance to be covered by the vehicle and the amount of load to be carried is 101.7 and 46, respectively. The route information of other vehicles, the distances covered, and the load amounts are also given under the sample solution. Considering that the capacity of the vehicles is 60, the given sample solution is a valid solution. The graphical representation of this solution and total distance (fitness value) are given in Figure 1d.

		Depot	Customers											
Symbols		0	1	2	3	4	5	6	7	8	9	10	11	12
Demands		-	19	21	15	13	17	22	16	14	16	18	10	11
Coordinates	x	52	96	50	49	13	29	58	84	14	2	3	5	98
	y	46	44	5	8	7	89	30	39	24	39	82	10	52

No of Trucks: 4 Capacity of trucks: 60

(a)

Sample solution														
12	1	7	0	9	11	6	0	2	3	4	0	5	10	8

Route 1 = {0, 12, 1, 7, 0} = 101.7
Capacity 1 = 46

Route 2 = {0, 9, 11, 6, 0} = 160.2
Capacity 2 = 48

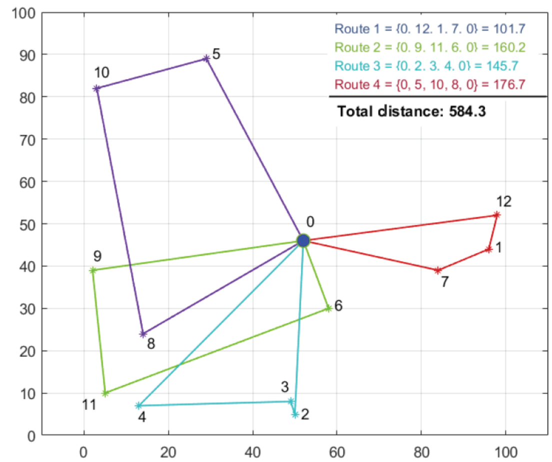
Route 3 = {0, 2, 3, 4, 0} = 145.7
Capacity 3 = 49

Route 4 = {0, 5, 10, 8, 0} = 176.7
Capacity 4 = 49

(c)

		Depot	Customers											
		0	1	2	3	4	5	6	7	8	9	10	11	12
Depot	0	-	44,7	47,0	44,1	59,5	43,6	22,8	34,5	47,2	51,7	57,5	63,0	46,0
Customers	1	44,7	-	60,3	59,2	90,9	80,7	40,5	13,0	84,4	94,1	100,5	97,1	8,2
	2	47,0	60,3	-	3,2	37,1	86,6	26,2	48,1	40,7	58,8	90,2	45,3	67,2
	3	44,1	59,2	3,2	-	36,0	83,4	23,8	46,8	38,5	56,3	87,1	44,0	65,9
	4	59,5	90,9	37,1	36,0	-	83,5	50,5	77,9	17,0	33,8	75,7	8,5	96,2
	5	43,6	80,7	86,6	83,4	83,5	-	65,7	74,3	66,7	56,8	26,9	82,6	78,3
	6	22,8	40,5	26,2	23,8	50,5	65,7	-	27,5	44,4	56,7	75,7	56,6	45,7
	7	34,5	13,0	48,1	46,8	77,9	74,3	27,5	-	71,6	82,0	91,7	84,2	19,1
	8	47,2	84,4	40,7	38,5	17,0	66,7	44,4	71,6	-	19,2	59,0	16,6	88,5
	9	51,7	94,1	58,8	56,3	33,8	56,8	56,7	82,0	19,2	-	43,0	29,2	96,9
	10	57,5	100,5	90,2	87,1	75,7	26,9	75,7	91,7	59,0	43,0	-	72,0	99,6
	11	63,0	97,1	45,3	44,0	8,5	82,6	56,6	84,2	16,6	29,2	72,0	-	102,0
	12	46,0	8,2	67,2	65,9	96,2	78,3	45,7	19,1	88,5	96,9	99,6	102,0	-

(b)



(d)

Figure 1. a) The instance consisting of twelve customers and one depot b) The distances between customers and depot c) The sample solution d) The graphical expression of sample solution and total distance.

If the amount of load to be carried by a vehicle exceeds the vehicle capacity, the constraint given by (5) is violated. In this case, the new solution is invalid. The death penalty procedure used for constraint processing rejects the solution. The relevant trial step is skipped and the next trial step starts. The death penalty method used in the proposed algorithm is quite simple and effective. Unlike static, dynamic or adaptive penalty methods, it does not require extra computation [26]. It accelerates the proposed algorithm. Thus, the algorithm runs stably on different sized instances.

After the fitness values of the solutions constituting the population are calculated, the process of finding new and better solutions about current solutions starts. In this process, any of exchange, insertion, and reversion route development operators are used. If the generated new solution is better than the previous solution, it is considered as the current solution. If it is not better than the previous solution, then a value w is calculated using the equation $w = e^{\frac{-\triangle f}{T}}$, where $\triangle f$ is the difference between the fitness values of the new solution and the previous solution and T is the temperature. If this value is greater than the randomly generated value r between 0 and 1, the generated new solution is considered as the current solution. Otherwise the route development process is ignored. These processes are repeated for all solutions constituting the population up to the number of trials given as input to the algorithm. The best fitness value and the route solution giving this value are determined for each iteration and recorded. After each iteration, the temperature T should be reduced depending on the rate calculated according to the start temperature T_{start} and the end temperature T_{end}. In the last iteration, the temperature T decreases to the end temperature T_{end} given as input to the algorithm. The result of the proposed population based simulated annealing algorithm is the best fitness value obtained during the iterations and the route solution giving this value.

3.2. The route development operators

Exchange, insertion and reversion operators were used by Yu and Lin [27] to solve the location-routing problem with simultaneous pickup and delivery (LRPSPD). They conducted various experiments on four LRPSPD instances to analyze the performance of these operators. They examined the convergence of any solution by applying each operator alone and randomly applying all the operators. A faster convergence was achieved by random application of exchange, insertion and reversion operators. At the same time, the better solutions were obtained. For this reason, Yu and Lin [27] decided with 1/3 probability which of them will be used in any iteration. In this study, one of exchange, insertion and reversion operators was used randomly to find a new and better solution about the current solution. Similarly, it was decided with 1/3 probability which of these operators will be used. However, exchange and insertion operators are slightly different from those proposed by Yu and Lin [27]. The number of points to be added or replaced can be 1 or 2. Thus, the proposed method gained the ability to obtain the faster convergence and the better solution.

3.2.1. The exchange operator

Figure 2a shows a sample solution, the route information of this solution and the amount of load to be carried for each route. The implementation of the exchange operator is given in Figures 2b and 2c. As can be seen from Figure 2b, the points 3 and 9 are randomly determined exchange points. The exchange numbers of the points are 1 and 2, respectively. The exchange numbers (1 or 2) are also determined randomly in the algorithm. Figure 2c shows the new solution obtained after the exchange operator was implemented. According to this solution, the routes 1 and 3 have changed and the amount of load to be carried by the vehicles following these routes is given below the routes. Since the new solution is a suitable solution, the amount of load to be carried by the vehicles should be less than or equal to the capacity of the vehicles. In order that the new solution becomes a better solution than the current solution, the distance to be covered should be shorter. Considering that the vehicle capacities are 60, it is not a suitable solution because of the amount of load to be carried by the vehicle following the route 1. Therefore, the exchange operator should be reapplied on the current solution for randomly redetermined exchange points and numbers.

3.2.2. The insertion operator

The implementation of the insertion operator on the sample solution (Figure 2a) is shown in Figures 2d and 2e. As can be seen from Figure 2d, the points 5 and 14 are randomly determined insertion points. The number of insertions (1 or 2) is also identified randomly in the algorithm. Figure 2d shows the new solution obtained after the insertion operator was implemented. According to this solution, the routes 2 and 4 have changed and the amount of load to be carried by the vehicles following these routes is given below the routes. Considering the vehicle capacities, it is not a suitable solution because of the amount of load to be carried by the vehicle following the route 2. Therefore, the insertion operator should be reapplied on the current solution for randomly redetermined insertion points and numbers.

3.2.3. The reversion operator

The implementation of the reversion operator on the sample solution (Figure 2a) is shown in Figures 2f and 2g. As can be seen from Figure 2f, the points 6 and 14 are randomly determined reversion points. The new solution is obtained by reversing the section between these reversion points. Figure 2g shows the new solution obtained after the reversion operator was implemented. According to this solution, the routes 2, 3 and 4 have changed and the amount of load to be carried by the vehicles following these routes is given below the routes. Considering vehicle capacities, the new solution is a suitable solution. It is also a better solution. Because the total distance for the sample solution given in Figure 2a is 584.3. The total distance for the new solution is 555.9. In this case, the new solution is considered as the current solution and the execution of the route development procedure is continued until the number of trials is reached.

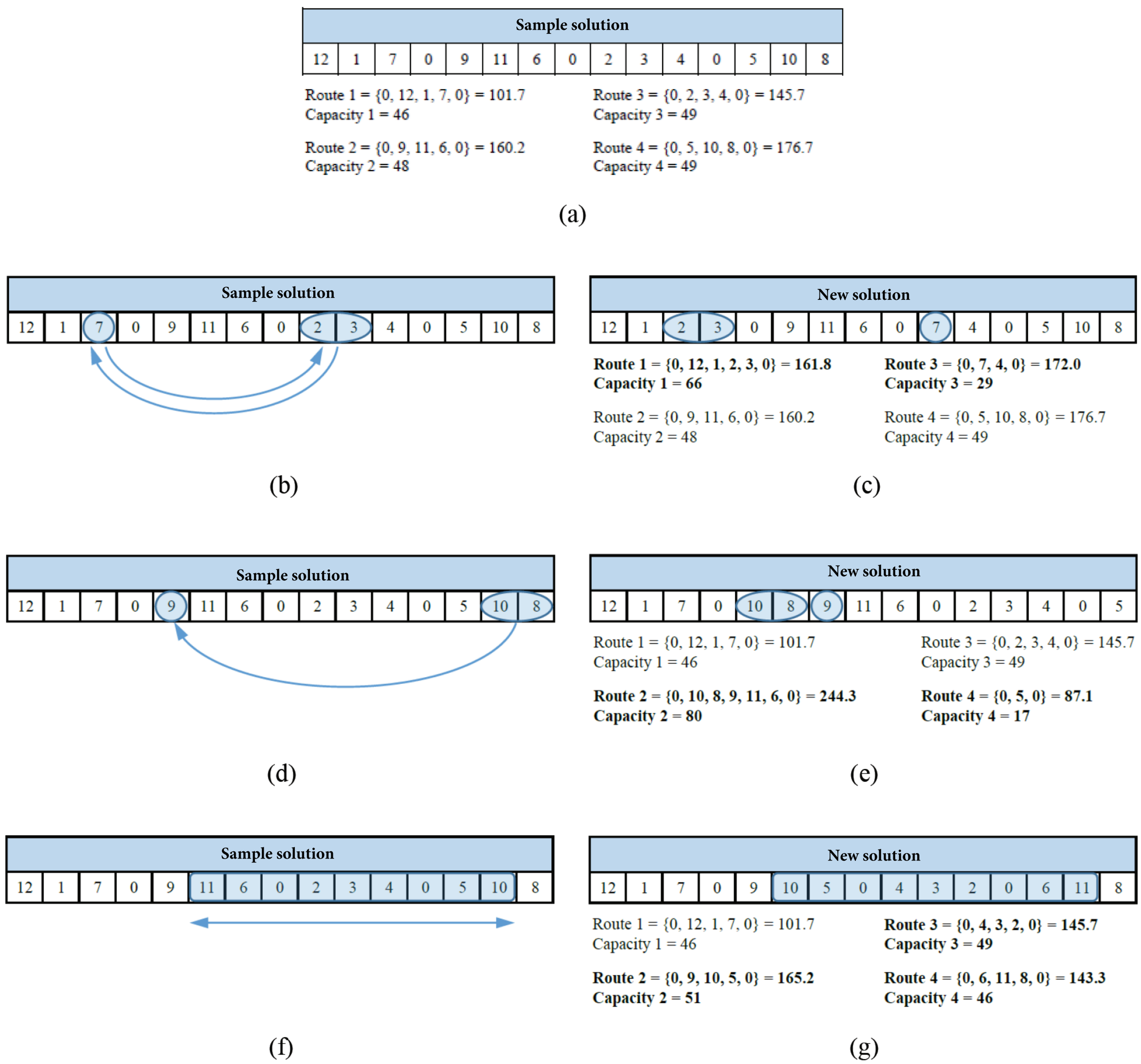


Figure 2. a) The sample solution b) Exchange points c) A new solution obtained for the exchange operator d) Insertion points e) A new solution obtained for the insertion operator f) Reversion points g) A new solution obtained for the reversion operator.

In the literature, a population based simulated annealing algorithm was proposed for the solution of the traveling tournament problem (TTP) which is a combinatorial optimization problem such as the CVRP [28].

This algorithm includes features such as both macro-intensification and macro-diversification. It is arranged in the form of a series of waves consisting of many simulated annealing runs. Each wave is followed by a macro-intensification and a macro-diversification. A macro-diversification is used to produce the best solution. A macro-intensification is used to find better solutions than the best solution available. In [28], the local search strategy is based on macro-intensification and macro-diversification operators. In this study, the local search strategy is built on exchange, insertion and reversion operators. These operators are applied with 1/3 probability to each solution in the population in any iteration. Thus, a new and better solution is tried to be found.

3.3. The pseudo-code and the interface

The pseudo-code of the proposed algorithm is given in Figure 3. As can be seen from the figure, the parameter values are primarily given as input to the algorithm. The number, locations and demand values of the customers, the number and capacities of the vehicles and the location of the depot are taken from the selected instance. The distances between the customers and the depot are calculated according to the location information in the instance. All this information is used by the proposed algorithm and the best route and the fitness value are produced as output.

```
Input:  I_n: The number of iterations; P_n: The number of solutions in population;
        T_n: The number of trials; T_start: The start temperature; T_end: The end temperature;
        IN: The instance;
Output: S_best: The best solution; f(S_best): The best fitness value
Begin
D ← Read_Instance(IN)
T ← T_start
Frac ← (T_end/T_start)^(1.0/(I_n-1.0))
x ← Create_Initial_Population(D, P_n)
f ← Compute_ Fitness_Value(D, x)
S_best ← x(1)
f(S_best) ← f(1)
for i ← 1 to I_n
  foreach s in x
    for j ← 1 to T_n
        s_new ← Develop_New_Route(s)
        if isInvalid(s_new)
            Death_Penalty(s_new)
        end if
        f(s_new) ← Compute_ Fitness_Value(s_new)
        if f(s_new)<f(s)
            s ← s_new
            f(s) ← f(s_new)
        else
            w = e^(-(f(s_new)-f(s))/T)
            if random(0,1)<w
                s ← s_new
                f(s) ← f(s_new)
            end if
        end if
    end for
    if f(s) < f(S_best)
        S_best ← s
        f(S_best) ← f(s)
    end if
  end foreach
  T ← Frac * T
end for
return(S_best, f(S_best))
End
```

Figure 3. The pseudo-code of the proposed algorithm.

The pseudo-code was coded in MATLAB and a new interface was developed. The interface is given in Figure 4. The parameter values for the proposed algorithm are entered via this interface. In addition, the best route obtained for each iteration can be followed via it and the fitness value of the route can be seen.

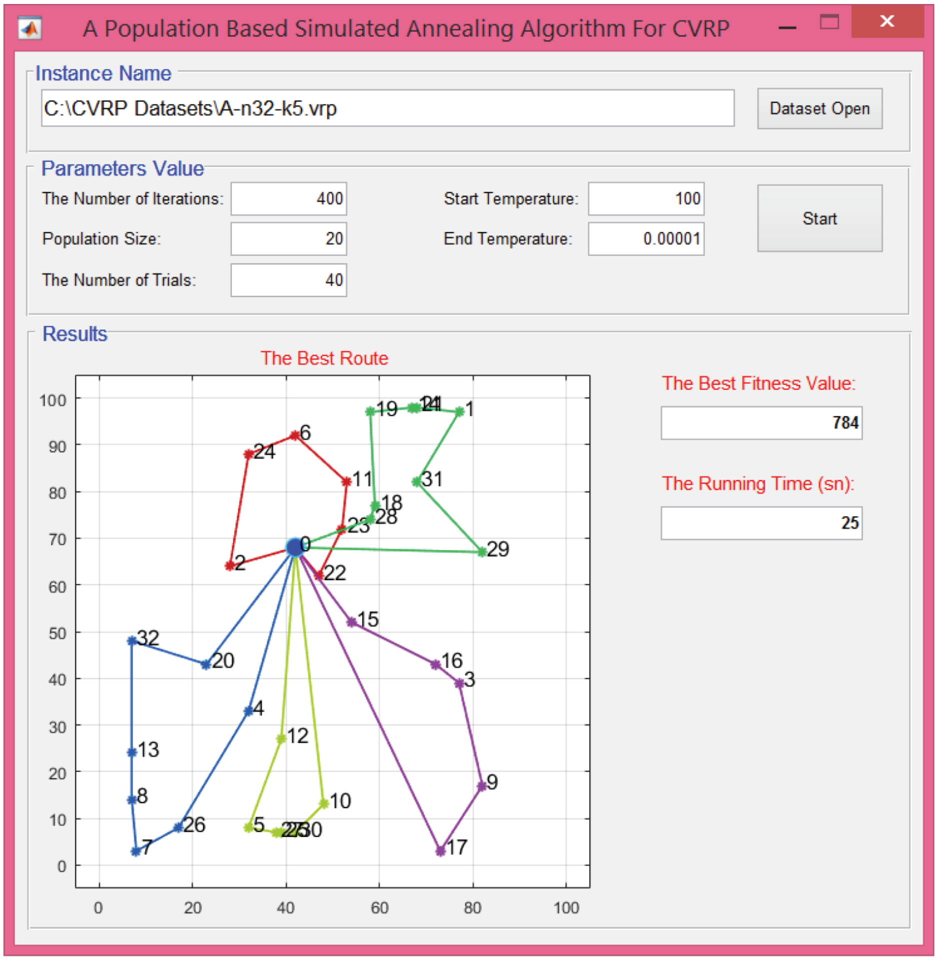


Figure 4. The interface of the proposed algorithm.

4. Experimental study

4.1. Test datasets

The proposed method was tested on well-known benchmark instances in the literature. A total of 63 instances were used, 50 of the datasets A, B and P proposed by Augerat et al. [29], 10 of the dataset E proposed by Christofides and Eilon [30] and 3 of the dataset F proposed by Fisher [31]. The instances have a depot, the number of demand points between 15 and 134, and the number of vehicles between 2 and 15. The tightness ratio of the instances is between 0.75 and 0.99. The tightness ratio is defined as the ratio of the sum of the demands of the customers to the sum of the capacities of the vehicles [31]. The B-n45-k6 has the highest tightness ratio while the E-n23-k3 has the lowest tightness ratio among the instances.

4.2. Parameter settings and sensitivity analysis

The proposed algorithm was run on a notebook with Windows 8.1 operating system. The notebook has an Intel Core i7 processor and 8 GB RAM. Since the parameter values play an important role in the quality of the solutions, a few preliminary experiments were carried out with the different parameter settings given follow:

The number of iterations (I_n) = {200, 300, 400, 500, 600};

The number of solutions in population (P_n) = {20, 50, 100};

The number of trials (T_n) = {30, 40, 50};

The start temperature (T_{start}) = {10, 100, 500};

The end temperature (T_{end}) = $\{10^{-3}, 10^{-4}, 10^{-5}\}$.

Each parameter combination was tested three times. According to the test results, when I_n = 400, P_n = 20, T_n = 40, T_{start} = 100 and T_{end} = 10^{-5}, it was observed that the proposed algorithm obtained the best results in terms of balance between the running time and the fitness value. Therefore, these parameter settings were used in the final analysis.

In order to analyze the performance of the proposed algorithm and the route development operators, a few experiments were performed on 8 instances with different number of demand points and vehicles. Four different scenarios were used in the experiments. The scenarios were that exchange, insertion and reversion route development operators were implemented alone and all of them were implemented randomly. The experiments showed that the random use of three different route development operators produced better solutions and provided faster convergence (Table 1).

4.3. Experimental results

The proposed algorithm was run five times on each instance using the parameter values and the route development operators. The fitness values and the running times were averaged separately for each instance. The averaged values were recorded as the results of the experiments. The results are given as detailed in Tables 2–4. Table 2 shows the results of the proposed method on 14 instances of the dataset A. The table also contains the results of the methods KNNA [18] and PVNS-ASM [19] together with those of Ewbank et al.'s method [17]. As can be seen from the table, the best-known optimal value was obtained by the method KNNA [18] and Ewbank et al.'s method [17] for no instance. The proposed method has almost the same result with the method PVNS-ASM [19] in terms of the number of the best-known optimal values. On the other hand, the proposed method run faster than the method PVNS-ASM [19]. The running time of the proposed method increases slightly according to the size of the instance. However, the running time of the method PVNS-ASM [19] increases in proportion to the size of the instance.

Table 3 shows the results of the proposed method on 16 instances of the dataset B. The table also contains the results of the method PVNS-ASM [19] and Ewbank et al.'s method [17]. As can be seen from the table, the number of the best-known optimal values obtained by Ewbank et al.'s method [17] is only 1. The best-known optimal values were obtained for 9 instances by both the method PVNS-ASM [19] and the proposed method. As on dataset A, the proposed method run faster than the method PVNS-ASM [19] on this dataset. The greater the size of the instances, the greater the difference in the average running time.

Table 4 shows the results of the proposed method on 33 instances of the datasets E, F and P. The table also contains the results of Ewbank et al.'s method [17]. As can be seen from the table, the best-known optimal value was obtained by Ewbank et al.'s method [17] for no instance. The number of best-known optimal values obtained by the proposed method is 6. The average percent error value of Ewbank et al.'s method [17] is 4.1% for 33 instances of these datasets. The average percent error value of the proposed method is 2.1%.

Table 1. The performance analysis of different route development operators.

Instance	**NODP**	**NOV**	**Tightness**	**BKO**	**All**		**Exchange**		**Insertion**		**Reversion**	
					Iteration	**Result**	**Iteration**	**Result**	**Iteration**	**Result**	**Iteration**	**Result**
A-n32-k5	31	5	0.82	784	**231**	**784**	300	835	303	794	287	843
A-n60-k9	59	9	0.92	1354	**242**	**1380**	390	1402	396	1406	400	1391
B-n38-k6	37	6	0.85	805	**171**	**820**	261	828	202	852	212	835
B-n68-k9	67	9	0.93	1272	**371**	**1272**	395	1283	400	1277	399	1281
E-n23-k3	22	3	0.75	569	**185**	**569**	221	587	212	576	223	576
E-n76-k14	75	14	0.97	1021	**391**	**1046**	400	1067	400	1081	400	1078
P-n16-k8	15	8	0.88	450	**161**	**450**	232	480	258	477	351	469
P-n101-k4	100	4	0.91	681	**396**	**705**	398	815	376	798	383	749

NODP: The Number of Demand Point, NOV: The Number of Vehicle, BKO: The Best Known Optimal

Table 2. The comparison of results obtained for Augerat et al.'s [29] dataset A.

Instance	NODP	NOV	Tightness	BKO	Ewbank et al. [17]	KNNA [18]		PVNS-ASM [19]		The Proposed Method								
					Error	Error	Time	Error	Time	Result	Error	Time	Max. Result	Min. Result	Max. Time	Min. Time	Count	Total Time
A-n32-k5	31	5	0.82	784	3.6	1.7	**0.85**	**0.0**	31	784	**0.0**	25	785	784	27	23	3	126
A-n33-k5	32	5	0.89	661	2.9	21.8	**0.84**	**0.0**	33	661	**0.0**	26	663	661	28	25	4	131
A-n33-k6	32	6	0.90	742	2.0	17.1	**0.86**	**0.0**	32	750	1.1	25	755	748	27	24	0	125
A-n37-k5	36	5	0.81	669	3.6	32.0	**1.14**	**0.0**	59	669	**0.0**	24	670	669	25	23	4	122
A-n37-k6	36	6	0.95	949	2.9	5.9	**1.17**	**0.2**	23	972	2.4	25	980	963	28	24	0	126
A-n39-k6	38	6	0.88	831	1.7	17.6	**1.32**	**0.0**	99	831	**0.0**	28	831	831	30	26	5	142
A-n45-k6	44	6	0.99	944	1.3	19.6	**1.86**	**0.42**	96	958	1.5	26	963	952	28	24	0	131
A-n45-k7	44	7	0.91	1146	4.6	3.4	**1.87**	**0.0**	81	1146	**0.0**	29	1147	1146	31	28	3	146
A-n46-k7	45	7	0.86	914	2.4	26.3	**1.96**	**0.0**	92	939	2.7	28	947	935	31	26	0	142
A-n48-k7	47	7	0.89	1073	2.3	12.1	**2.28**	**0.0**	47	1073	**0.0**	29	1073	1073	31	27	5	146
A-n55-k9	54	9	0.93	1073	3.7	18.9	**3.04**	**0.0**	101	1073	**0.0**	32	1075	1073	33	31	4	162
A-n60-k9	59	9	0.92	1354	3.6	2.0	**3.77**	**0.37**	74	1380	1.9	33	1387	1370	36	30	0	167
A-n65-k9	64	9	0.97	1174	4.8	32.1	**5.63**	0.51	84	1174	**0.0**	30	1174	1174	32	29	5	150
A-n80-k10	79	10	0.94	1763	6.5	1.3	**8.28**	**1.19**	186	1837	2.1	31	1850	1820	33	30	0	157
Average					**3.3**	**15.1**	**2.5**	**0.2**	**74**		**0.8**	**28**						

NODP: The Number of Demand Point, NOV: The Number of Vehicle, BKO: The Best Known Optimal, Error: The Percent Error, Time: The Running Time in Seconds, Result: The Fitness Value, Max. Result and Max. Time: The Maximum Values in Five Runs, Min. Result and Min. Time: The Minimum Values in Five Runs, Count: The Number of Obtaining The Best Known Solution in Five Runs, Total Time: The Total Time For Five Runs.

Table 3. The comparison of results obtained for Augerat et al.'s [29] dataset B (This dataset has not been processed by the method KNNA [18])

Instance	NODP	NOV	Tightness	BKO	Ewbank et al. [17]	PVNS-ASM [19]		The Proposed Method									
					Error	Error	Time	Result	Error	Time	Max. Result	Min. Result	Max. Time	Min. Time	Count	Total Time
B-n31-k5	30	5	0.82	672	0.6	**0.0**	28	672	**0.0**	**26**	674	672	27	25	4	129
B-n34-k5	33	5	0.91	788	1.8	**0.0**	46	788	**0.0**	**26**	788	788	26	25	5	128
B-n38-k6	37	6	0.85	805	3.8	**0.0**	41	820	1.8	**27**	827	815	28	26	0	135
B-n39-k5	38	5	0.88	549	3.5	**0.0**	81	549	**0.0**	**27**	550	549	27	26	3	133
B-n41-k6	40	6	0.95	829	2.7	**0.12**	46	831	0.2	**26**	834	830	28	24	0	130
B-n43-k6	42	6	0.87	742	3.4	**0.0**	57	742	**0.0**	**27**	743	742	30	25	4	137
B-n44-k7	43	7	0.92	909	2.2	**0.0**	41	937	3.1	**28**	940	934	31	24	0	138
B-n45-k5	44	4	0.97	751	3.0	**0.0**	130	751	**0.0**	**27**	752	751	30	24	3	136
B-n45-k6	44	6	0.99	678	4.6	0.29	63	678	**0.0**	**27**	678	678	28	24	5	134
B-n50-k7	49	7	0.87	741	7.2	**0.0**	80	750	1.2	**28**	755	747	31	26	0	142
B-n50-k8	49	8	0.92	1312	4.0	**0.1**	51	1358	3.5	**28**	1370	1349	29	27	0	139
B-n56-k7	55	7	0.88	707	2.1	**0.0**	126	707	**0.0**	**30**	707	707	32	27	5	148
B-n66-k9	65	9	0.96	1316	2.2	0.3	105	1316	**0.0**	**35**	1318	1316	37	32	4	173
B-n67-k10	66	10	0.91	1032	3.9	**1.0**	92	1062	2.9	**33**	1068	1055	38	30	0	167
B-n68-k9	67	9	0.93	1272	3.5	1.2	150	1272	**0.0**	**33**	1274	1272	35	30	4	164
B-n78-k10	77	10	0.94	1221	**0.0**	1.72	196	1250	2.4	**32**	1257	1245	33	30	0	158
Average					**3.0**	**0.3**	**83**		**0.9**	**29**						

NODP: The Number of Demand Point, NOV: The Number of Vehicle, BKO: The Best Known Optimal, Error: The Percent Error, Time: The Running Time in Seconds, Result: The Fitness Value, Max. Result and Max. Time: The Maximum Values in Five Runs, Min. Result and Min. Time: The Minimum Values in Five Runs, Count: The Number of Obtaining The Best Known Solution in Five Runs, Total Time: The Total Time For Five Runs.

Table 4. The comparison of results obtained for Christofides and Eilon's [30] dataset E, Fisher's [31] dataset F and Augerat et al.'s [29] dataset P (These datasets have not been processed by the methods KNNA [18] and PVNS-ASM [19])

Instance	NODP	NOV	Tightness	BKO	Ewbank et al. [17]	The Proposed Method								
					Error	Result	Error	Time	Max. Result	Min. Result	Max. Time	Min. Time	Count	Total Time
E-n22-k4	21	4	0.94	375	**0.1**	381	1.6	22	385	379	24	21	0	111
E-n23-k3	22	3	0.75	569	0.1	569	**0.0**	21	570	569	21	20	4	104
E-n30-k3	29	3	0.94	534	8.0	549	**2.8**	25	552	545	26	25	0	127
E-n33-k4	32	4	0.92	835	1.2	835	**0.0**	24	835	835	25	22	5	118
E-n51-k5	50	5	0.97	521	0.7	521	**0.0**	26	522	521	29	23	4	131
E-n76-k7	75	7	0.89	682	5.8	697	**2.2**	29	700	693	30	28	0	146
E-n76-k8	75	8	0.95	735	5.4	753	**2.4**	30	760	750	32	29	0	152
E-n76-k10	75	10	0.97	830	5.4	830	**0.0**	34	832	830	38	31	4	171
E-n76-k14	75	14	0.97	1021	7.6	1046	**2.4**	36	1054	1039	38	36	0	182
E-n101-k8	100	8	0.91	815	5.9	833	**2.2**	31	845	825	34	28	0	157
Average					**4.0**		**1.4**	**28**						
F-n45-k4	44	4	0.90	724	8.5	756	**4.4**	35	760	750	37	34	0	177
F-n72-k4	71	4	0.96	237	**1.2**	241	1.7	41	243	240	42	40	0	207
F-n135-k7	134	7	0.95	1162	2.1	1182	**1.7**	40	1188	1175	43	38	0	201
Average					**4.0**		**2.6**	**39**						
P-n16-k8	15	8	0.88	450	0.2	450	**0.0**	39	451	450	42	36	4	197
P-n19-k2	18	2	0.97	212	3.7	218	**2.8**	22	222	214	22	21	0	108
P-n20-k2	19	2	0.97	216	**1.1**	219	1.4	33	222	217	34	32	0	163
P-n21-k2	20	2	0.93	211	4.3	217	**2.8**	35	219	216	38	32	0	173
P-n22-k2	21	2	0.96	216	0.9	216	**0.0**	32	217	216	33	31	4	161
P-n22-k8	21	8	0.94	603	4.6	619	**2.7**	39	625	610	42	36	0	197
P-n40-k5	39	5	0.88	458	2.3	468	**2.2**	39	472	463	41	38	0	197
P-n45-k5	44	5	0.92	510	**0.5**	517	1.4	41	521	515	43	38	0	203

Table 4. (Continued).

Instance	**NODP**	**NOV**	**Tightness**	**BKO**	**Ewbank et al. [17]**	**The Proposed Method**								
					Error	**Result**	**Error**	**Time**	**Max. Result**	**Min. Result**	**Max. Time**	**Min. Time**	**Count**	**Total Time**
P-n50-k7	49	7	0.91	554	4.5	561	**1.3**	42	563	560	44	40	0	211
P-n50-k10	49	10	0.95	696	5.1	716	**2.9**	32	720	709	36	29	0	162
P-n51-k10	50	10	0.97	741	6.0	769	**3.8**	34	776	759	35	32	0	169
P-n55-k7	54	7	0.88	568	12.5	581	**2.3**	42	586	575	44	41	0	211
P-n55-k10	54	10	0.91	694	7.2	721	**3.9**	42	723	720	45	39	0	209
P-n60-k10	59	10	0.95	744	10.2	784	**5.4**	43	790	770	45	41	0	214
P-n60-k15	59	15	0.95	968	12.3	989	**2.2**	30	996	983	32	28	0	149
P-n65-k10	64	10	0.94	792	4.1	809	**2.2**	37	815	801	39	35	0	185
P-n70-k10	69	10	0.97	827	5.9	842	**1.8**	35	846	836	38	31	0	174
P-n76-k4	75	4	0.97	593	2.8	602	**1.5**	36	606	598	37	35	0	179
P-n76-k5	75	5	0.97	627	**1.6**	638	1.8	37	642	633	39	36	0	187
P-n101-k4	100	4	0.91	681	4.2	705	**3.6**	40	711	698	42	39	0	201
Average					**4.4**		**2.3**	**37**						

NODP: The Number of Demand Point, NOV: The Number of Vehicle, BKO: The Best Known Optimal, Error: The Percent Error, Time: The Running Time in Seconds, Result: The Fitness Value, Max. Result and Max. Time: The Maximum Values in Five Runs, Min. Result and Min. Time: The Minimum Values in Five Runs, Count: The Number of Obtaining The Best Known Solution in Five Runs, Total Time: The Total Time For Five Runs.

As can be seen from Tables 2–4, the running time of the proposed metaheuristic method increases slightly according to the size of the instance. This case can be explained by three different approaches used in the method. These approaches:

1. The constraint processing approach: The death penalty method is a popular constraint processing method used in optimization algorithms [26]. In this method, a solution is rejected when it violates a constraint. No extra calculations are needed to estimate its degree of invalidity. This reduces the computational complexity of the algorithm and gives it speed [26]. At the same time, it allows the algorithm to be affected as little as possible by the instances properties (the number of demand points and the vehicles, the tightness ratio).

2. The creation of the initial population: The solutions in the initial population were randomly generated based on the number of demand points and vehicles. However, it was checked in the same procedure whether a solution produced was a valid solution. If the solution was an invalid solution, a new solution was randomly generated instead. This generation process continued until the number of solutions equaled the population size. The procedure used to create the initial population is given in Figure 5. This procedure ensured the first iteration of the algorithm starts with valid solutions. Thus, it was provided that the change in the running time of the algorithm is less according to the size of the instances.

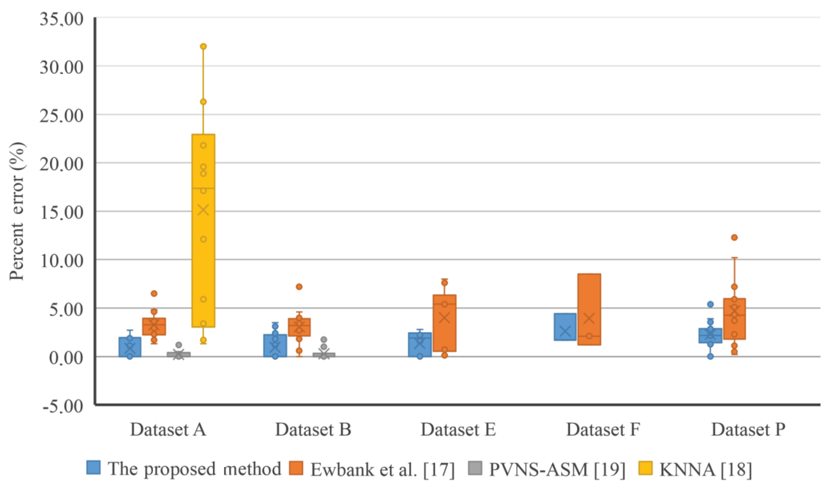
```
Input:   D: The read instance; P_n: The number of solutions in population;
Output:  x: The initial population

    Create_Initial_Population(D, P_n)
    Begin
    i ← 1
    while i<=P_n
        x(i) ← randperm(D.NumberOfDemandPoints+D.NumberOfVehicles-1)
        positions ← find(x(i)>D.NumberOfDemandPoints)
        x(i, positions) ← 0
        if isInvalid(x(i))
            continue
        end if
        i ← i+1
    end while
    return(x)
    End
```

Figure 5. The procedure for creating the initial population.

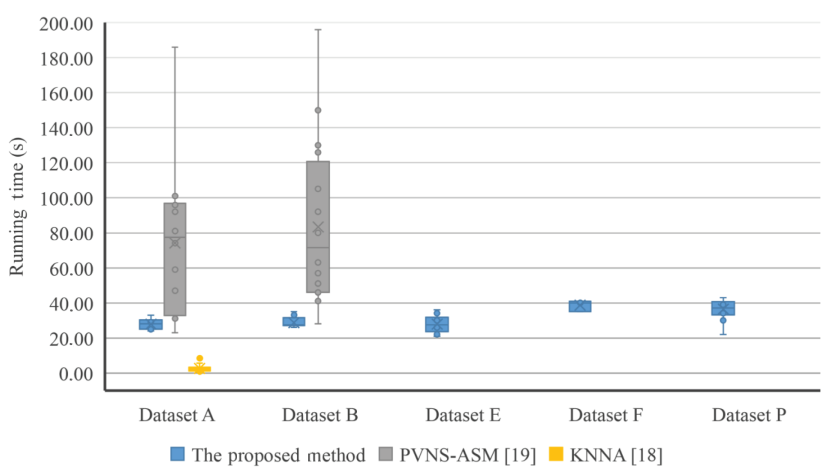
3. The route development operators: Exchange, insertion and reversion operators are used to produce a new solution about the current solution. It was decided with 1/3 probability which of these operators will be applied on the current solution. Figure 2 illustrates how these operators are applied to a solution. As can be seen from the figure, the time consumption of the exchange and insertion operators is independent of the length of the solution (the number of demand points and vehicles). The time consumption of the reversion operator depends on the randomly determined reversion points. Both the selection of the reversion operator and the determination of the reversion points are performed with a certain probability. Therefore, the effect of this operator on the time consumption of the proposed algorithm remains limited.

The fitness value of a new solution generated in the proposed algorithm is calculated as shown in Figures 1c and 1d. The length of the generated solution string is directly dependent on the size of the instance (Figures 1a and 1b). As the size of the instance (the number of demand points and vehicles) increases, the length of the solution string increases. The longer the solution string, the longer it takes to calculate its fitness value. This time directly affects the running time of the algorithm. However, this effect is the same as the values given in detail in Tables 2–4.

Figure 6a shows as a boxplot the percent error values for the datasets A, B, E, F and P. Figure 6b gives as a boxplot the running times. The running times were not processed by Ewbank et al.'s method [17] for these datasets. Therefore, Ewbank et al.'s method [17] is not included in Figure 6b. In terms of the percent error value, the method PVNS-ASM [19] has a more stable feature. However, the same method has a very unstable feature in terms of running time. The opposite is the case for the method KNNA [18]. The method KNNA [18] has a more stable feature in terms of running time. However, the same method has a very unstable feature in terms of the percent error value. Considering the percent error value and running time together, the proposed method in this study is a more stable than the other methods compared.



(a)



(b)

Figure 6. a) The boxplot of the percent errors b) The boxplot of the running times.

5. Conclusion and future research

The CVRP, one of the most discussed and researched topics nowadays, was studied in this paper. A novel population based simulated annealing algorithm is proposed to solve the CVRP. In this algorithm, three different route development operators, which are exchange, insertion and reversion operators, were used. It was decided randomly which of these operators to use in any iteration. The proposed algorithm was tested on 63 well-known benchmark instances in the literature. The results were recorded together with the running times for each instance. They showed that the proposed algorithm could determine optimum routes for the 23 instances. In addition, considering the percent error value and running time together, it was determined that the proposed method was a more stable than the other methods compared.

Future research may focus on the use of different metaheuristic algorithms, such as gray wolf, harmony search, firefly optimization algorithm to solve the CVRP. Even, a new route development operator can be researched. A new hybrid method that combines two metaheuristics can be developed.

References

[1] Dantzig GB, Ramser JH. The truck dispatching problem. Management Science 1959; 6 (1): 80-91. doi: 10.1287/mnsc.6.1.80

[2] Toth P, Vigo D. Vehicle Routing: Problems, Methods, and Applications. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2015.

[3] Jourdan L, Basseur M, Talbi EG. Hybridizing exact methods and metaheuristics: a taxonomy. European Journal of Operational Research 2009; 199 (3): 620-629. doi: 10.1016/j.ejor2007.07.035

[4] Lawler EL, Wood DE. Branch-and-bound methods: a survey. Operations Research 1966; 14 (4): 699-719. doi: 10.1287/opre.14.4.699

[5] Gomory RE. Outline of an algorithm for integer solutions to linear programs. Bulletin of the American Mathematical Society 1958; 64 (5): 275-278.

[6] Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MW, Vance PH. Branch-and-price: column generation for solving huge integer programs. Operations Research 1998; 46 (3): 316-329. doi: 10.1287/opre.46.3.316

[7] Toth P, Vigo D. The Vehicle Routing Problem. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2002.

[8] Papadimitriou CH, Steiglitz K. Combinatorial Optimization: Algorithms and Complexity. USA: Prentice-Hall, 1982.

[9] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983; 220 (4598): 671-680. doi: 10.1126/science.220.4598.671

[10] Glover F. Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 1986; 13: 533-549. doi: 10.1016/0305-0548(86)90048-1

[11] Baghel M, Agrawal S, Silakari Sanjay. Survey of metaheuristic algorithms for combinatorial optimization. International Journal of Computer Applications 2012; 58 (19): 21-31. doi: 10.5120/9391-3813

[12] Yu B, Yang ZZ, Yao B. An improved ant colony optimization for vehicle routing problem. European Journal of Operational Research 2009; 196 (1): 171-176. doi:10.1016/j.ejor.2008.02.028

[13] Harmanani HM., Azar D, Helal N, Keirouz W. A simulated annealing algorithm for the capacitated vehicle routing problem. In: 26th International Conference on Computers and Their Applications; New Orleans, Louisiana, USA; 2011. pp. 96-101.

[14] Nazif H, Lee LS. Optimised crossover genetic algorithm for capacitated vehicle routing problem. Applied Mathematical Modelling 2012; 36 (5): 2110-2117. doi: 10.1016/j.apm.2011.08.010

[15] Zang SZ, Lee CKM. An improved artificial bee colony algorithm for the capacitated vehicle routing problem. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics; Kowloon, China; 2015. pp. 2124-2128.

[16] Teoh BE, Ponnambalam SG, Kanagaraj G. Differential evolution algorithm with local search for capacitated vehicle routing problem. International Journal of Bio-Inspired Computation 2015; 7 (5): 321-342. doi: 10.1504/IJBIC.2015.072260

[17] Ewbank H, Wanke P, Hadi-Vencheh A. An unsupervised fuzzy clustering approach to the capacitated vehicle routing problem. Neural Computing and Applications 2016; 27 (4): 857-867. doi: 10.1007/s00521-015-1901-4

[18] Mohammed MA, Ghania MKA, Hamedc RI, Mostafad SA, Ibrahime DA et al. Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution. Journal of Computational Science 2017; 21: 232-240. doi: 10.1016/j.jocs.2017.04.012

[19] Faiz A, Subiyanto S, Arief UM. An efficient meta-heuristic algorithm for solving capacitated vehicle routing problem. International Journal of Advances in Intelligent Informatics 2018; 4 (3): 212-225. doi: 10.1016/j.cor.2004.07.009

[20] Menger K. Das botenproblem. Ergebnisse Eines Mathematischen Kolloquiums 1932; 2 (4): 11-12 (in German).

[21] Ballou RH. Business logistics management: Planning, organizing and controlling the supply chain. USA: Prentice Hall, 1999.

[22] Toth P, Vigo D. Exact solution of the vehicle routing problem. In: Teodor GC, Gilbert L (Editors). Fleet Management and Logistics. New York, USA: Springer, 1998, pp. 1-31.

[23] Dell AM, Rignihi G, Salani M. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. Transportation Science 2006; 40 (2): 235-247. doi: 10.1287/trsc.l050.0118

[24] Chen A, Yang G, Wu Z. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University-Science A 2006; 7 (4): 607-614. doi: 10.1631/jzus.2006.A0607

[25] Askarzadeh A, Santos CL, Klein CE, Mariani VC. A population-based simulated annealing algorithm for global optimization. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC); Budapest, Hungary; 2016. pp. 4626-4633.

[26] Coello C, Carlos A. Constraint-handling techniques used with evolutionary algorithms. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion; Denver, Colorado, USA; 2016. pp. 563-587.

[27] Yu VF, Lin SY. Solving the location-routing problem with simultaneous pickup and delivery by simulated annealing. International Journal of Production Research, 2016; 54 (2): 526-549. doi: 10.1080/00207543.2015.1085655

[28] Hentenryck P, Vergados Y. Population-based simulated annealing for traveling tournaments. In: Proceedings of the National Conference on Artificial Intelligence; Vancouver, British Columbia; 2007. pp. 1-20.

[29] Augerat PH, Belenguer JM, Benavent E, Corberan A, Naddef D et al. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report 949-M, Grenoble, France: 1995.

[30] Christofides N, Eilon S. An algorithm for the vehicle-dispatching problem. Journal of the Operational Research Society 1969; 20 (3): 309-318. doi: 10.1057/jors.1969.75

[31] Fisher ML. Optimal solution of vehicle routing problems using minimum K-trees. Operational Research 1994; 42: 626-642. doi: 10.1287/opre.42.4.626