Int j simul model 18 (2019) 4, 574-585
ISSN 1726-4529 Original scientific paper

COMPARISON OF DIFFERENT CLUSTERING ALGORITHMS VIA GENETIC ALGORITHM FOR VRPTW

Gocken, T.[#] & Yaktubay, M.
Department of Industrial Engineering, Adana Alparslan Turkes Science and Technology University, Adana, Turkey
E-Mail: tgocken@atu.edu.tr ([#] Corresponding author)

Abstract
In this paper, Vehicle Routing Problem with Time Windows (VRPTW) with known customer demands, a central depot and a set of vehicles with limited capacity, is considered. The objectives are both to minimize the total distance and the total waiting time of the vehicles while capacity and time windows constraints are secured. The applied solution techniques consist of three steps: clustering, routing and optimizing. By using K-means, Centroid-based heuristic, DBSCAN and SNN clustering algorithms in the initial population generation phase of genetic algorithm, the customers are divided into feasible clusters. Then feasible routes are constructed for each cluster. Lastly, the feasible route solutions are taken as the initial population and genetic algorithm is utilized for the optimization. A set of well-known benchmark data is used to compare the obtained results. According to the results of the study it is observed that using K-means clustering algorithm in generating the initial population of the genetic algorithm is more effective for the handled problem.
(Received in April 2019, accepted in October 2019. This paper was with the authors 1 month for 1 revision.)

Key Words: Vehicle Routing with Time Windows, Genetic Algorithm, Clustering, Multi-Objective Optimization, K-means Clustering Algorithm

1. INTRODUCTION

Vehicle Routing Problem with Time Windows (VRPTW) which is a type of classical Vehicle Routing Problem (VRP) handles a transportation issue that is comprised in the logistics management which is a substantial component of the supply chain management. VRPTW searches optimum routes for a fleet of vehicles making delivery from a depot to the customers in a specified time interval. Route optimization has a significant importance in logistics management owing to the effect on the customer satisfaction by fast delivery and lower cost. In accordance with this concept, VRPTW is concentrated on minimizing total travelled distance and usage of the vehicles, which are directly related with economic and environmental sustainability.

Many approaches that have utilized exact, heuristic and metaheuristic algorithms have been developed heretofore to solve VRPTW which is a hard-combinatorial problem. If the size of the customer set is small, the exact algorithms can be utilizable; else if the set is getting larger it is not viable to use these algorithms due to the high solution duration. For this reason, the solution approaches of the problem via heuristic and metaheuristic strategies are growing in the literature recently. Genetic Algorithm (GA) is frequently used in the solution of the VRP and VRPTW. It is a useful tool to produce good solutions to optimization and search problems. GA iteratively improves a set of solutions by mimicking the biological evolution in natural selection. Because GA is a versatile and effective approach, it is prone to the hybridization. It aims to increase the probability of getting the best solution of an optimization problem and to decrease the time of searching. In the literature of the VRP and VRPTW, there are plenty of studies that applied GA incorporated with other heuristics [1-7].

In the literature, there are also studies that utilized clustering algorithms in the routing problem. Díaz-Parra et al. [8] used K-means clustering algorithm and neighbourhood techniques to produce a non-random initial population for GA and solved a VRPTW. Comert et al. [9] proposed a hierarchical approach for solving VRPTW in a real life situation. At first, the customers are clustered by K-means, K-medoids and DBSCAN clustering algorithms separately. Then, the routes are formed with an exact method. Shin and Han [10] proposed a 3-phase centroid-based heuristic algorithm for VRP. Their algorithm constructs clusters first, then adjusts the clusters, at last establishes the routes. They presented that, their proposed algorithm achieves better results than the sweep algorithm. López-Santana et al. [11] considered a problem of scheduling and routing in a courier service. Before the construction of the routes, the customers are clustered by a centroid-based algorithm and a sweep algorithm. Min et al. [12] proposed a three-stage algorithm for split delivery VRP. First, maximum-minimum distance method is employed to cluster customers. Second, load-demand adjustment is employed. Third, a tabu search algorithm is used to optimise the routes. Zhang [13] used a density based clustering algorithm, i.e. improved DBSCAN, in the solution approach for a VRP. Erdogan and Miller-Hooks [14] proposed a solution approach for a green VRP. They developed a density based clustering algorithm in a part of their approach. It has been built based on DBSCAN algorithm.

In this study, the effect of the different clustering algorithms on the GA performance is analysed by using them at the initial population generation step. K-means, centroid-based heuristic, DBSCAN and SNN clustering algorithms are used at the initial population generation step. According to the best of our knowledge, there is not a comparable study measuring the effect of these partitioning and density based clustering algorithms on GA for solving multi-objective vehicle routing problem with capacity and time windows constraints. The algorithms have been tested on Solomon's well-known benchmarking problems of VRPTW literature.

2. PROBLEM DEFINITION

VRP is the problem of forming optimal routes to the vehicles that will serve a customer set. The information of the customers and the depot(s) (e.g. numbers, demand quantities, geographic data) are known before starting the solution of the problem. The vehicles that serve the customers are assumed to be composed of a homogeneous fleet with a capacity limitation. Each vehicle begins on the route from the depot and returns to the depot at the end of the route. The requirements of each customer must be met in one single vehicle at a time. The total demand of the customers which served by the same vehicle should not be in excess of the capacity limit of the vehicle.

VRP with only one restriction, i.e. vehicle capacity restriction, refers to the Capacitated Vehicle Routing Problem (CVRP). The components which all are found in CVRP, i.e. a homogeneous fleet with specific number of vehicles with the same capacity and characteristics, customers with known demands and locations, a warehouse with a known geographical location, are exist also in VRPTW. The constraint that makes VRPTW different and challenging from other VRP varieties is that there is a specific time interval (e_i, l_i) at which service can start for each customer. This time interval is the time window constraint of each relevant customer. The time window of the depot signifies the maximum traveling time of the individual routes. Making delivery or providing services to each customer takes as long as service duration s_i. At the end of the service duration, the vehicle drives to the next customer or the depot [5].

In this study, minimization of the total distance and minimization of the total waiting time of the vehicles are determined as the objective functions and the problem is modelled as a multi-objective optimization problem.

The mathematical model of the problem is presented in the study of Gocken et al. [5].

3. PROPOSED APPROACH

The applied solution techniques in our study consist of three steps: clustering, routing and optimizing. The customers are clustered as groups, taking into consideration of the vehicle capacity constraint. To get initial feasible solutions the grouped customers are routed according to the time window constraint. The solutions are optimized via GA beginning with these initial solutions.

By using K-means, Centroid-based heuristic, DBSCAN and SNN clustering algorithms in the initial population generation phase of GA, the customers are divided into feasible clusters. Then feasible routes are constructed for each cluster. Lastly, the feasible route solutions are taken as the initial population and GA is utilized for the optimization.

3.1 Clustering phase

(1) K-means

K-means is a partitioning clustering algorithm and developed by MacQueen in [15]. K-means algorithm partitions the set of objects into *K* clusters. The name of the algorithm comes from the parameter *K*, the number of clusters. The steps of K-means algorithm are shown in Table I.

Table I: Steps of K-means algorithm.

Parameter: *k*
1: Select *k* random points as centroids i.e. initial cluster centres.
2: repeat
3: Assign each point to the closest centroid's cluster.
4: Check the feasibility of capacity constraint.
5: Recalculate the centroid of each cluster.
6: until Centroids do not change.

(2) Centroid-based heuristic

Shin and Han [10] proposed a 3-phase Centroid-based heuristic algorithm to solve the CVRP. The phases are cluster construction, cluster adjustment and route establishment. In this study, cluster construction and cluster adjustment phases are utilized for clustering the customers. The steps of Centroid-based heuristic algorithm are shown in Table II. The steps 1-11 represent cluster construction phase and 12-16 represent cluster adjustment phase.

Table II: Steps of Centroid-based heuristic algorithm.

1: Select the farthest point from the depot as the first customer of the first cluster.
2: repeat
3: The first point of the cluster is the centroid of the cluster.
4: repeat
5: Find the closest un-clustered point.
6: Check the feasibility of capacity constraint.
7: Assign the closest un-clustered point to the cluster.
8: Recalculate the centroid of the cluster.
9: until The vehicle capacity do not exceed.
10: Select an arbitrary un-clustered point as the first customer of the next cluster.
11: until There is no un-clustered point.
12: repeat
13: if A point is closer a centroid of a cluster than its own, **and** there is available capacity for that point
14: Move that point to the cluster of with the closest centroid.
15: Recalculate the centroids of the clusters.
16: until Centroids do not change.

(3) DBSCAN

Density based clustering algorithm named as Density based Spatial Clustering of Application with Noise (DBSCAN) proposed by Ester et al. [16]. The density of a point is defined as the number of points, that is, neighbours, located within a specified radius (*Eps*) from that point. The main idea is to form the clusters that have at least *MinPts* points within the *Eps* neighbourhood. *Eps* neighbourhood of a point p, denoted by $N_{Eps}(p)$, is the class of the points that have at most *Eps* distance from p. The points which have at least *MinPts* neighbours in its N_{Eps} class are defined as core points. The core points are directly density-reachable to their neighbours in N_{Eps} class. The points that are non-core points but are in the N_{Eps} class of a core point are defined as border points. A border point is the point which is density-connected to a border point in the same cluster. The steps of DBSCAN algorithm are shown in Table III.

Table III: Steps of DBSCAN algorithm.

Parameters: *Eps*, *MinPts*
1: Search each point's *Eps* neighbourhood (N_{Eps}).
2: Determine the core points which have at least *MinPts* neighbours in *Eps* neighbourhood.
3: repeat
4: Select an arbitrary un-clustered core point as the first customer of the cluster.
5: **repeat**
6: Assign the points (which are directly density-reachable from the core point) in *Eps* neighbourhood of the core point to the cluster.
7: Check the feasibility of capacity constraint.
8: **if** The vehicle capacity is not full,
9: Assign the points (which are density-reachable from the core point) in *Eps* neighbourhood of the core points, which are already taken to the cluster, to the cluster.
10: **until** The vehicle capacity do not exceed, if it exceeds discard the points beginning from the farthest one and return to the previous status.
11: until There is no un-clustered point.

(4) Shared Nearest Neighbour

Shared Nearest Neighbour (SNN) clustering algorithm proposed by Ertoz et al. [17]. SNN algorithm is a density based clustering algorithm as DBSCAN algorithm. The main difference between them is the density concept. In SNN algorithm, the density is related not only the distance between points, but also the number of the nearest points that they share. The steps of SNN algorithm are shown in Table IV.

Table IV: Steps of SNN algorithm.

Parameters: *k*, *Eps*, *T*, *MinPts*
1: Define each point's NN( ) class by searching nearest *k* neighbours.
2: Revise all NN( ) classes by looking at whether the points are mutually in each other's classes.
3: Calculate the SNN similarity of all points between the points and their neighbours in NN( ) class by looking at the number of shared neighbours.
4: Search each point's *Eps* neighbourhood ($N_{Eps}(\,)$).
5: Calculate the SNN density of all points by looking at the number of neighbours with SNN similarity at least *T* in $N_{Eps}(\,)$.
6: Determine the core points which have at least *MinPts* neighbours in SNN density.
7: repeat
8: Select an arbitrary un-clustered core point as the first customer of the cluster.
9: **repeat**
10: Assign the points (which are directly density-reachable from the core point) in *Eps* neighbourhood of the core point to the cluster.
11: Check the feasibility of capacity constraint.

if The vehicle capacity is not full,
    Assign the points (which are density-reachable from the core point) in *Eps* neighbourhood of the core points, which are already taken to the cluster, to the cluster.
until The vehicle capacity do not exceed.
until There is no un-clustered point.

3.2 Routing phase

The routing process occurs with the same rules for all algorithms. It begins with the inclusion of the depot in the clusters. At first, the depot is added and then the customer who is nearest to the depot in one cluster is assigned to the route. The distances between the non-routed customers and the last customer assigned to the route are computed. The closest one that provides the time constraint is assigned to the route. Thus, the routes are built for the customers at each cluster. If any non-routed customer exists, those who do not violate the limitations are assigned to a route. In the lack of a feasible location, a new route is formed. The practicable solutions are achieved via these improvements.

3.3 Random-based initial algorithm

For the observation of the effectiveness of the clustering algorithms at the generation of the initial population, a random-based algorithm is used for the comparison. Taking the time window constraint into account, the steps of the Random-based algorithm are indicated in Table V. The algorithm constructs the routes while selecting the customers.

Table V: Steps of Random-based algorithm.

Select the closest point to the depot as the first visited location of the first route.
repeat
    repeat
        Select an arbitrary non-routed point.
        Check the feasibility of capacity constraint.
        Check the feasibility of time window constraint.
        Assign it to the route.
    until Any of the constraints are violated.
    Form a new route beginning with the last point that violates any constraints in the previous route.
until There is no non-routed point.

3.4 Optimizing phase

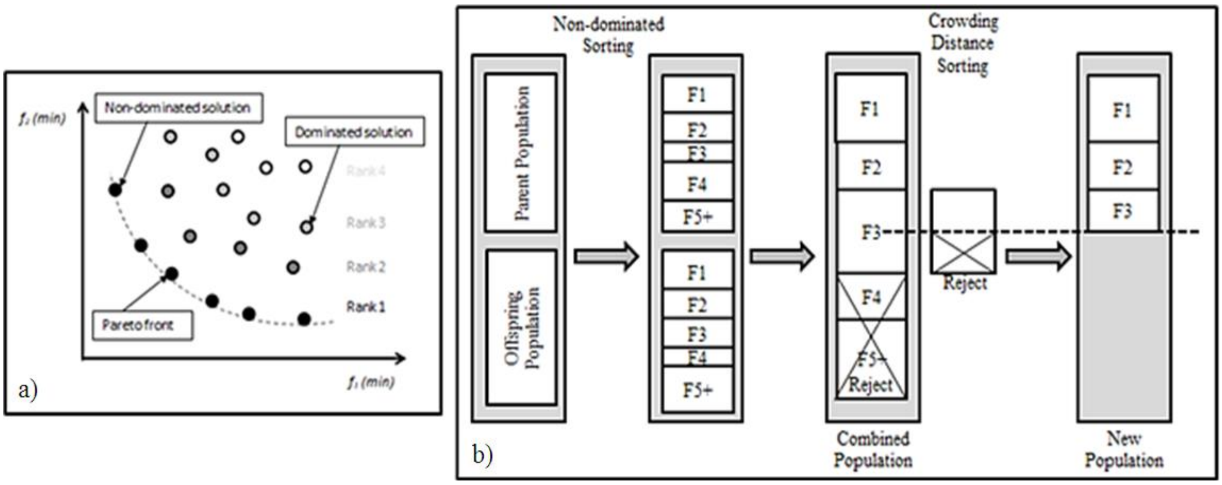

Figure 1: a) Pareto front and ranking scheme, b) Elitist selection procedure of NSGA-II.

After generating the initial population, GA is used for the optimization. NSGA-II (Non-dominated Sorting Genetic Algorithm II) which is one of the multi objective optimization techniques is used in the evaluation, ranking and selection of the individuals at GA steps. DEAP Library is employed to perform NSGA-II [18, 19]. The steps of the proposed GA are shown in Table VI.

Table VI: Steps of Genetic algorithm.

Parameters: *PopulationSize*, $P_{crossover}$, $P_{mutation}$, *GenerationNumber*
1: Choose one of the initial clustering algorithms from K-means, Centroid-based heuristic, DBSCAN, SNN and Random-based.
2: Generate the initial population as much as *PopulationSize* using Permutation encoding with a separator, -1, for specifying the routes in a single array.
3: Evaluate and order the dominance of the individuals via optimal Pareto fronts with NSGA-II.*
4: Calculate the domination counts of each individual and determine all the Pareto fronts as seen in Fig. 1 a.
5: **if** The individuals are on the same front,
6: Calculate the crowding distance values of the individuals and rank them in descending order.
7: Set the iteration is 0.
8: repeat
9: Select the individuals as parents.
10: Rank the dominance of the individuals and pair the individuals in consecutive form.
11: Perform the two-point crossover operation with the rate of $P_{crossover}$.
12: Determine two cut points where after the separator -1.
13: Exchange the contents among the cut points between the parent pair.
14: **if** There is a repeated customer in the array,
15: Keep the customer in the first place where it is seen and deleted from where it is repeated.
16: **if** There is a non-routed customer,
17: Add the customer to a place that makes the solution feasible. In the absence of such a place, a new route is generated.
18: Perform the mutation operation with the rate of $P_{mutation}$.
19: Pick a random route from the produced offspring and then a random customer from the picked route.
20: Extract that customer from that route and positioned in the best space to improve the objective function value.
21: Evaluate and order the dominance of the individuals via optimal Pareto fronts with NSGA-II with steps 4, 5 and 6.
22: Apply replacement strategy of NSGA-II with elitism for evolving next generation as seen in Fig. 1 b.*
23: Increase the iteration by 1.
24: until The iteration is equal to *GenerationNumber*.

* For additional information of NSGA-II see [20, 5].

4. NUMERICAL RESULTS

The generated algorithms that utilize different clustering algorithms or a random-based algorithm are tested on C1, C2, R1 and R2 classes of Solomon's benchmark problems [21]. All problem instances have one depot, 100 customers and a fleet of homogeneous vehicles. Geographic positions of the customers and the depots are given by (x, y) coordinates. The route length between them is computed by Euclidean distance and the obtained value is in the unit of distance. It is assumed that the travelling of 1 unit distance takes 1 unit time. Customer demand quantities and service durations are available in the data set. The earliest and latest arrival times (i.e., time windows) of each customer are given. The problem classes differ according to the geographical data features and scheduling horizon type. In C category classes, the customers are clustered. In R category classes, the customers are distributed randomly and uniform. The problems in C1 and R1 classes have a short scheduling horizon; i.e., the time window of the depot is narrow; and low vehicle capacity. On the contrary, the problems in C2 and R2 classes have a long scheduling horizon; i.e., the time window of the depot is wide; and high vehicle capacity. The problem instances are available at [22].

The algorithms are coded in the Python 3.6. For testing the implementation, a computer with i5 processor technology, 3.00 GHz processor speed and 8 GB RAM capacity is used.

The parameter K which specifies the number of clusters in the K-means algorithm is calculated by adding 1 to the ratio of the total demand to the vehicle capacity. While forming the clusters by using K-means and Centroid-based clustering algorithms maximum iteration number is set to 10. The parameters of the DBSCAN algorithm are predetermined to be $Eps = 12$ and $MinPts = 4$. The parameters of the SNN algorithm are predetermined to be $k = 40$, $Eps = 12$, $T = 1$ and $MinPts = 5$. The parameter values used in the GAs are as follows: $P_{crossover}$ is always 0,7 and $P_{mutation}$ is 0,2. Population size is 200 (at DBSCAN it is 300) and the number of generations is 200 (at SNN it is 300). The probability parameters are decided based on the values used in the literature. Other parameters used in GA and clustering algorithms are determined after preliminary runs of the problem.

For each problem set 5 independent tests are run. Algorithms gave the final solution populations as a result. Table VII is formed using the best distance values and Table VIII is formed using the best waiting time values on the Pareto optimal fronts of the final populations. These alternate Pareto solutions are comparable and decision maker can decide which solution is more preferable based on the preferences. The results are compared between the generated algorithms.

Table VII shows that K-means algorithm has attained better performance than other algorithms for 14 results which are shown with bold values according to the minimum travelled distance and minimum required total vehicle numbers. The following better performance belongs to CBased algorithm with 9 best results. Even so, K-means and CBased algorithms are equally competitive according to the difference between results, especially in the short scheduling horizon problems. The results reveal that DBSCAN and SNN (i.e., with density based clustering algorithms) are not as efficient as the algorithms with partitioning clustering algorithms; they show poor performances up to 7 % according to the average total distance values of the classes. Random based algorithm is generally not as good as the other algorithms in particular in the C1 and C2 class problems. It shows poor performance up to 30 % according to the average total distance values of the classes. It also gives up to 65 % worse result for problem basis than K-means algorithm, in C208. However, RN algorithm results reveal that the algorithm is not as poor on random distributed customers as that of clustered customers. It reaches 4 best results in R1 and 3 best results in R2 class problems according to the total distance values, but it always requires more vehicles in total of the classes.

From Table VIII it is seen that the RN algorithm reached 33 best results out of 40 instance results in minimizing waiting time values (29 of them have 0 waiting time). It reveals better results in R1 and R2 class problems. However, it has not compensated for the travelled distance values and the required total vehicle number. According to the total of the travelled distance values, K-means algorithm achieves the best results in particular in the C1 and C2 class problems. Also, K-means algorithm reached 29 instance results with 0 waiting time. CBased algorithm performs approximately close to K-means algorithm on total travelled distance and waiting time values. DBSCAN and SNN algorithm results are not as efficient as the partitioning clustering algorithms. Moreover, required total vehicle numbers are minimum in K-means algorithm (with 332) and maximum in RN algorithm (with 343).

Figs. 2 and 3 display the class averages of the algorithms results according to the travelled distance and waiting time values. It enables the comparison of the data in Table VII and Table VIII. It is seen that a decrement in waiting time values causes an increment in travelled distance values. This fact can be observed in all problem instances and at all algorithms. In multi-objective optimization, an improvement of one of the objectives may cause deterioration of the other. Only in R1 class, the success of reducing waiting time values is not as good as of other classes. In general, the minimum averages of the results yielded by K-means algorithm. Particularly in C1 and C2 classes, K-means algorithm reveals better results than the other algorithms. In R1 and R2 classes, the averages of RN algorithm results on total distance and waiting time values might be seen the minimum, but for the instance problem basis it does not dominate the others and requires more vehicles.

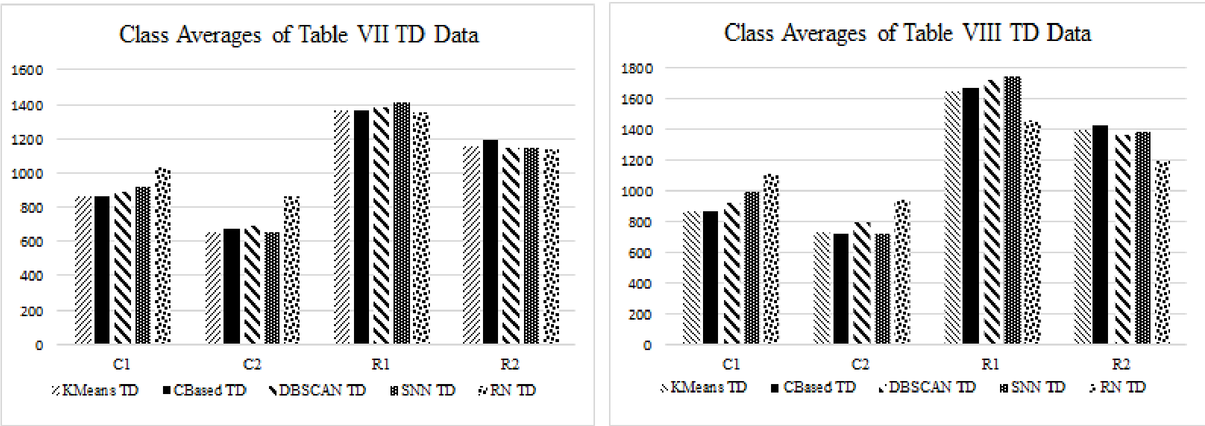

Figure 2: Comparison of the TD data in Table VII and Table VIII.

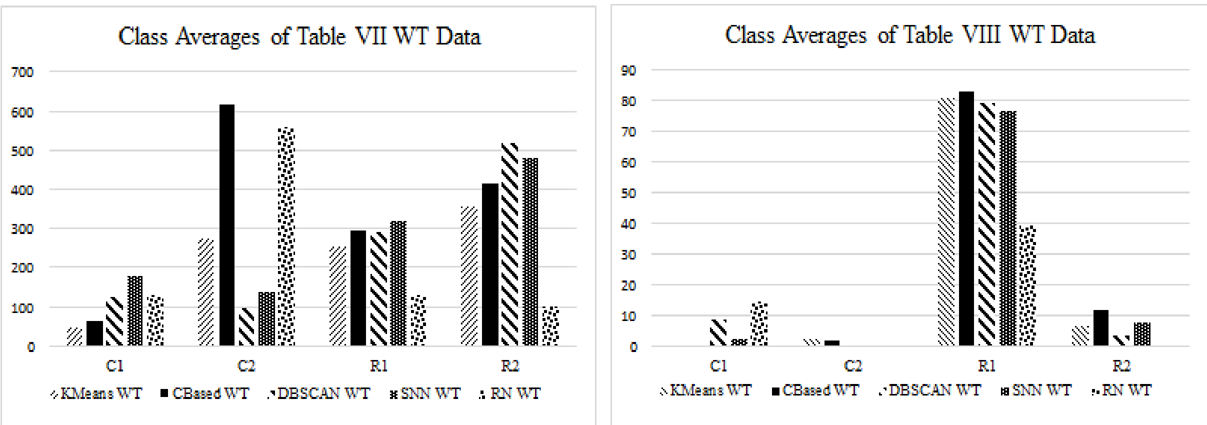

Figure 3: Comparison of the WT data on Table VII and Table VIII.

Fig. 4 indicates the average computational times (in minutes) of the algorithms for each class. K-means and CBased algorithms reach their results in less computational times. DBSCAN, SNN, and RN algorithms require almost twofold or threefold longer computational durations than K-means and CBased algorithms.

All the results of the tests show that K-means algorithm provided fast resolution, better total distance and waiting time values, effective competition with fewer vehicles. Nevertheless, it does not dominate over CBased algorithm.

Table VII: The solution results with the best distance values on the Pareto optimal fronts of the final populations obtained via the algorithms for each instances.

Instance	**K-means**			**CBased**			**DBSCAN**			**SNN**			**RN**		
	TD	***VN***	***WT***	***TD***	***VN***	***WT***	***TD***	***VN***	***WT***	***TD***	***VN***	***WT***	***TD***	***VN***	***WT***
C101	**828,94**	**10**	**0**	**828,94**	**10**	**0**	934,71	11	592	830,36	10	22	968,98	11	380
C102	919,71	11	221	**911,03**	**11**	**449**	970,89	11	390	1009,19	11	589	1170,97	12	128
C103	900,21	11	66	**887,52**	**10**	**0**	934,89	11	69	1005,16	11	453	1014,03	11	189
C104	**883,85**	**10**	**128**	888,23	10	128	928,71	10	68	1046,32	10	235	976,48	10	21
C105	**828,94**	**10**	**0**	**828,94**	**10**	**0**	833,24	10	0	838,07	10	0	1053,50	11	224
C106	**828,94**	**10**	**0**	846,16	10	0	850,46	10	0	846,16	10	0	1004,04	11	29
C107	**828,94**	**10**	**0**	**828,94**	**10**	**0**	880,84	10	0	838,07	10	0	1032,28	11	65
C108	871,58	10	0	871,58	10	0	**853,52**	**10**	**0**	891,43	10	44	996,45	11	71
C109	884,58	10	0	884,58	10	0	**856,31**	**10**	**0**	944,21	11	258	1124,75	11	58
C201	**591,56**	**3**	**0**	**591,56**	**3**	**0**	630,56	3	0	**591,56**	**3**	**0**	842,69	4	1044
C202	704,89	5	16	746,45	4	2559	746,87	3	34	**678,59**	**4**	**9**	839,06	4	839
C203	719,43	4	946	763,11	4	1676	**708,68**	**4**	**179**	741,52	4	467	975,07	4	422
C204	825,28	4	1245	822,34	4	715	782,12	4	459	**765,88**	**4**	**618**	830,00	4	308
C205	597,28	3	0	597,74	3	0	632,40	3	24	**591,56**	**3**	**0**	782,72	4	428
C206	626,60	3	0	626,60	3	0	687,14	3	0	**600,73**	**3**	**0**	814,11	4	141
C207	**599,86**	**3**	**0**	606,37	3	0	670,47	3	62	614,57	3	0	802,56	4	176
C208	599,28	3	0	**598,65**	**3**	**0**	662,01	4	0	626,47	3	0	990,63	4	1134
R101	**1785,17**	**19**	**848**	1816,32	21	1042	1819,82	20	830	1856,13	20	936	1861,53	21	460
R102	1696,91	18	477	1690,09	18	571	1681,08	18	584	1724,28	18	530	**1639,21**	**18**	**178**
R103	1501,75	15	318	1495,03	17	515	1455,93	15	394	1515,98	15	384	**1420,18**	**16**	**207**
R104	1153,63	11	98	1160,10	11	139	1142,56	12	228	1186,48	12	375	**1141,15**	**11**	**0**
R105	1579,18	15	260	1561,72	15	296	**1533,83**	**15**	**261**	1589,33	16	393	1587,80	16	226
R106	1433,61	14	203	1453,81	14	207	1451,11	14	157	1511,68	14	146	**1377,84**	**14**	**46**
R107	1261,82	12	120	**1232,87**	**13**	**216**	1264,28	13	123	1330,84	13	283	1257,75	13	105
R108	**1050,29**	**11**	**102**	1106,98	11	59	1135,08	11	172	1116,15	11	158	1073,72	11	0
R109	**1318,16**	**13**	**200**	1324,35	13	140	1373,14	14	238	1349,14	13	187	1321,02	13	114
R110	**1229,01**	**12**	**105**	1229,78	12	167	1305,84	13	138	1256,72	12	143	1281,80	13	54
R111	1244,90	12	219	**1222,61**	**12**	**84**	1241,10	13	252	1332,42	13	196	1231,28	13	96
R112	**1075,18**	**11**	**113**	1089,49	11	121	1134,05	11	108	1145,58	11	128	1103,03	11	58
R201	1521,51	4	600	1537,10	4	834	1520,62	5	1168	**1501,33**	**5**	**1135**	1507,05	5	447
R202	1380,91	4	485	1404,33	4	400	**1309,66**	**5**	**1156**	1351,14	5	1064	1356,34	5	141
R203	1232,20	4	410	1245,00	4	434	**1172,81**	**4**	**603**	1193,14	4	511	1208,84	4	0
R204	987,02	3	251	1020,69	4	614	1008,67	4	360	**943,33**	**4**	**610**	1014,00	4	80
R205	1248,12	3	331	1278,45	4	502	1212,99	3	110	1298,51	3	382	**1195,72**	**3**	**67**
R206	1184,13	4	351	1259,79	4	290	1134,15	3	325	1153,12	3	147	**1107,45**	**4**	**230**
R207	1073,49	4	499	1111,62	4	230	**1007,53**	**3**	**284**	1011,92	3	162	1034,94	3	25
R208	**829,58**	**3**	**268**	870,01	3	187	873,56	3	105	872,11	3	203	887,41	3	39
R209	1137,03	3	221	1173,20	3	188	1136,24	3	293	1171,62	3	302	**1056,82**	**3**	**3**
R210	**1144,70**	**3**	**177**	1207,22	4	604	1208,68	4	551	1199,28	4	521	1156,34	4	65
R211	958,21	3	323	973,81	3	280	990,49	4	766	**957,97**	**3**	**236**	981,87	3	7

Remark: The bold values represent the best values among the 5 algorithms.
TD – Total distance, *VN* – Vehicle number, *WT* – Waiting time of the vehicles.

Table VIII: The solution results with the best waiting time values on the Pareto optimal fronts of the final populations obtained via the algorithms for each instances.

Instance	K-means			CBased			DBSCAN			SNN			RN		
	TD	***VN***	***WT***	***TD***	***VN***	***WT***	***TD***	***VN***	***WT***	***TD***	***VN***	***WT***	***TD***	***VN***	***WT***
C101	**828,94**	**10**	**0**	**828,94**	**10**	**0**	941,02	11	66	830,36	10	22	1027,93	11	49
C102	**929,1**	**11**	**0**	**928,83**	**11**	**0**	**975,4**	**11**	**0**	**1013,26**	**11**	**0**	1186,18	12	29
C103	**919,65**	**11**	**0**	**887,52**	**10**	**0**	**996,85**	**11**	**0**	**1388,84**	**12**	**0**	1137,16	11	8
C104	**910,33**	**10**	**0**	**930,44**	**11**	**0**	1130,23	11	12	**1314,2**	**11**	**0**	997,94	10	8
C105	**828,94**	**10**	**0**	**828,94**	**10**	**0**	**833,24**	**10**	**0**	**838,07**	**10**	**0**	1266,55	11	8
C106	**828,94**	**10**	**0**	**846,16**	**10**	**0**	**850,46**	**10**	**0**	**846,16**	**10**	**0**	1004,04	11	29
C107	**828,94**	**10**	**0**	**828,94**	**10**	**0**	**880,84**	**10**	**0**	**838,07**	**10**	**0**	**1237,55**	**11**	**0**
C108	**871,58**	**10**	**0**	**871,58**	**10**	**0**	**853,52**	**10**	**0**	**915,97**	**10**	**0**	**1049,01**	**11**	**0**
C109	**884,58**	**10**	**0**	**884,58**	**10**	**0**	**856,31**	**10**	**0**	**995**	**10**	**0**	**1130**	**11**	**0**
C201	**591,56**	**3**	**0**	**591,56**	**3**	**0**	**630,56**	**3**	**0**	**591,56**	**3**	**0**	**1208,68**	**4**	**0**
C202	**806,13**	**5**	**0**	835,5	4	11	**1288,38**	**4**	**0**	**856,1**	**4**	**0**	**914,25**	**4**	**0**
C203	1071,35	4	18	847,47	4	2	**836,41**	**4**	**0**	991,13	4	3	**989,28**	**4**	**0**
C204	**1018,2**	**4**	**0**	**1084,68**	**4**	**0**	**999,16**	**4**	**0**	**933,83**	**4**	**0**	**863,27**	**4**	**0**
C205	**597,28**	**3**	**0**	**597,74**	**3**	**0**	**635,78**	**3**	**0**	**591,56**	**3**	**0**	**900,44**	**4**	**0**
C206	**626,6**	**3**	**0**	**626,6**	**3**	**0**	**687,14**	**3**	**0**	**600,73**	**3**	**0**	**842,21**	**4**	**0**
C207	**599,86**	**3**	**0**	**606,37**	**3**	**0**	**670,96**	**3**	**0**	**614,57**	**3**	**0**	**826,69**	**4**	**0**
C208	**599,28**	**3**	**0**	**598,65**	**3**	**0**	**662,01**	**4**	**0**	**626,47**	**3**	**0**	**997,98**	**4**	**0**
R101	2156,14	20	539	1990,38	19	560	2326,89	21	542	2226,98	20	530	**2037,22**	**20**	**253**
R102	2188,5	19	204	2018,22	18	227	1972,03	18	228	2092,4	18	170	**1953,31**	**21**	**94**
R103	2100,56	18	44	**2232,16**	**18**	**39**	2154,63	18	42	1859,57	15	54	1493,23	15	41
R104	**1361,58**	**11**	**0**	**1510,37**	**15**	**0**	**1737,28**	**14**	**0**	**2015,46**	**16**	**0**	**1141,15**	**11**	**0**
R105	1881,27	16	125	1667,6	15	126	1720,93	15	96	1782,31	15	117	**1726,65**	**16**	**78**
R106	1713,14	15	11	1702,95	15	15	1699,77	15	13	1630,33	14	12	**1495,67**	**15**	**0**
R107	**1557,42**	**14**	**0**	**1465,86**	**14**	**0**	1867,66	17	0	1800,32	15	0	**1285,27**	**13**	**0**
R108	**1097,78**	**11**	**0**	**1125,39**	**10**	**0**	**1210,05**	**11**	**0**	**1233,3**	**11**	**0**	**1073,72**	**11**	**0**
R109	1457,17	13	38	1665,92	15	20	1630,11	14	21	1771,59	15	24	**1498,43**	**14**	**5**
R110	1614,71	14	4	1846,77	14	7	1725,97	14	5	1546,31	13	9	**1345,17**	**13**	**0**
R111	1547,4	14	4	1622,34	14	2	1423,34	13	7	1768,29	15	5	**1318,03**	**13**	**0**
R112	**1204,87**	**11**	**0**	**1173,89**	**11**	**0**	**1214,46**	**11**	**0**	**1205,86**	**11**	**0**	**1119,81**	**11**	**0**
R201	2034,7	4	47	2010,08	4	74	2089,09	4	29	2031,22	4	60	**1928,24**	**4**	**0**
R202	1748,68	4	23	1823,46	4	43	1903,71	4	9	1945,05	4	26	**1408,18**	**5**	**0**
R203	**1631,6**	**4**	**0**	1628,08	4	6	**1379,5**	**3**	**0**	**1618,55**	**4**	**0**	**1208,84**	**4**	**0**
R204	**1238,79**	**3**	**0**	1242,69	3	4	**1184,61**	**3**	**0**	**1053,28**	**3**	**0**	**1027,16**	**3**	**0**
R205	**1467,16**	**3**	**0**	**1540,5**	**3**	**0**	**1415,84**	**3**	**0**	**1520,65**	**3**	**0**	**1250**	**3**	**0**
R206	**1281,59**	**3**	**0**	**1363,65**	**3**	**0**	**1263,4**	**3**	**0**	**1238,65**	**3**	**0**	**1145,06**	**4**	**0**
R207	**1159,62**	**3**	**0**	**1179,94**	**3**	**0**	**1064,87**	**3**	**0**	**1118,94**	**3**	**0**	**1046,36**	**3**	**0**
R208	**960,26**	**3**	**0**	**1009,82**	**3**	**0**	**953,48**	**3**	**0**	**928,7**	**3**	**0**	**888,85**	**3**	**0**
R209	**1369,04**	**3**	**0**	1348,07	3	3	1288,25	3	1	1343,51	3	1	**1082,36**	**3**	**0**
R210	**1336,17**	**3**	**0**	1441,79	3	0	**1357,18**	**3**	**0**	**1375,48**	**3**	**0**	**1232,42**	**4**	**0**
R211	**1120,49**	**3**	**0**	**1125,55**	**3**	**0**	**1133,69**	**3**	**0**	**1119,64**	**3**	**0**	**983,56**	**3**	**0**

Remark: The bold values represent the best values among the 5 algorithms.
TD – Total distance, *VN* – Vehicle number, *WT* – Waiting time of the vehicles.

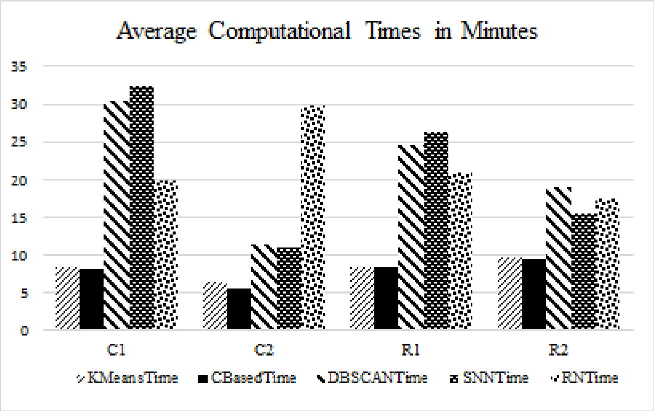

Figure 4: Average computational times of the algorithms for each class.

5. CONCLUSION

In this study, a multi objective GA approach for the VRPTW solution is proposed and the effect of using different clustering algorithms in the initial population generation step of GA is investigated. By using K-means, Centroid-based heuristic, DBSCAN and SNN clustering algorithms, the customers are divided into feasible clusters. Then feasible routes are constructed for each cluster. Lastly, the feasible route solutions are taken as the initial population and GA is utilized for the optimization. For the observation of the effectiveness of the clustering algorithms at the generation of the initial population, for comparison, a random-based algorithm is also used. The formed five algorithms; i.e. K-means, CBased, DBSCAN, SNN and RN; are tested on Solomon's VRPTW benchmark problems.

It can be concluded that the K-means algorithm showed better performance than other algorithms taking into consideration of the travelled distance, waiting time, vehicle number and computational time criteria. Nevertheless, it has not seen a significant difference between K-means and CBased algorithms. They performed better than DBSCAN and SNN algorithms. Therefore, it can be interpreted that partitioning clustering algorithms are more appropriate for the VRPTW solution approaches than the density based clustering algorithms. Lastly, travelled distance results of the RN algorithm indicated that using clustering algorithms in the initial population generation step of GA have a positive effect on the results.

In the future studies, a parameter analysis can be made on the clustering algorithms for examining the influence on the algorithms performance. It may increase the efficiency of the DBSCAN and SNN algorithms. Furthermore, the effects of random selection of initial points for K-means algorithm can be analysed.

REFERENCES

[1] Baker, B. M.; Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem, *Computers & Operations Research*, Vol. 30, No. 5, 787-800, doi:10.1016/S0305-0548(02)00051-5

[2] Karagul, K.; Gungor, I. (2014). A case study of heterogeneous fleet vehicle routing problem: Touristic distribution application in Alanya, *An International Journal of Optimization and Control: Theories & Applications*, Vol. 4, No. 2, 67-76, doi:10.11121/ijocta.01.2014.00185

[3] Mousavi, M.; Yap, H. J.; Musa, S. N.; Dawal, S. Z. M. (2017). A fuzzy hybrid GA-PSO algorithm for multi-objective AGV scheduling in FMS, *International Journal of Simulation Modelling*, Vol. 16, No. 1, 58-71, doi:10.2507/IJSIMM16(1)5.368

[4] Zhao, P. X.; Luo, W. H.; Han, X. (2019). Time-dependent and bi-objective vehicle routing problem with time windows, *Advances in Production Engineering & Management*, Vol. 14, No. 2, 201-212, doi:10.14743/apem2019.2.322

[5] Gocken, T.; Yaktubay, M.; Kilic, F. (2017). Improvement of a genetic algorithm approach for the solution of vehicle routing problem with time windows, *Proceedings of the 2017 International Artificial Intelligence and Data Processing Symposium*, 8 pages, doi:10.1109/IDAP.2017.8090185

[6] Ghoseiri, K.; Ghannadpour, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm, *Applied Soft Computing*, Vol. 10, No. 4, 1096-1107, doi:10.1016/j.asoc.2010.04.001

[7] Wu, D. Q.; Dong, M.; Li, H. Y.; Li, F. (2016). Vehicle routing problem with time windows using multi-objective co-evolutionary approach, *International Journal of Simulation Modelling*, Vol. 15, No. 4, 742-753, doi:10.2507/IJSIMM15(4)CO19

[8] Díaz-Parra, O.; Ruiz-Vanoye, J. A.; Zavala-Díaz, J. C. (2010). Population pre-selection operators used for generating a non-random initial population to solve vehicle routing problem with time windows, *Scientific Research and Essays*, Vol. 5, No. 22, 3529-3537

[9] Comert, S. E.; Yazgan, H. R.; Sertvuran, I.; Sengul, H. (2017). A new approach for solution of vehicle routing problem with hard time window: an application in a supermarket chain, *Sādhanā*, Vol. 42, No. 12, 2067-2080, doi:10.1007/s12046-017-0754-1

[10] Shin, K.; Han, S. (2011). A centroid-based heuristic algorithm for the capacitated vehicle routing problem, *Computing and Informatics*, Vol. 30, No. 4, 721-732

[11] López-Santana, E.; Rodríguez-Vásquez, W. C.; Méndez-Giraldo, G. (2018). A hybrid expert system, clustering and ant colony optimization approach for scheduling and routing problem in courier services, *International Journal of Industrial Engineering Computations*, Vol. 9, No. 3, 369-396, doi:10.5267/j.ijiec.2017.8.001

[12] Min, J. N.; Jin, C.; Lu, L. J. (2019). Maximum-minimum distance clustering method for split-delivery vehicle-routing problem: Case studies and performance comparisons, *Advances in Production Engineering & Management*, Vol. 14, No. 1, 125-135, doi:10.14743/apem2019.1.316

[13] Zhang, J. (2017). An efficient density-based clustering algorithm for the capacitated vehicle routing problem, *Proceedings of the 2017 International Conference on Computer Network, Electronic and Automation*, 465-469, doi:10.1109/ICCNEA.2017.96

[14] Erdogan, S.; Miller-Hooks, E. (2012). A green vehicle routing problem, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 48, No. 1, 100-114, doi:10.1016/j.tre.2011.08.001

[15] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281-297

[16] Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, 226-231

[17] Ertoz, L.; Steinbach, M.; Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, *Proceedings of the 2003 SIAM International Conference on Data Mining*, 47-58, doi:10.1137/1.9781611972733.5

[18] Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester

[19] Fortin, F.-A.; De Rainville, F.-M.; Gardner, M.-A.; Parizeau, M.; Gagné, C. (2012). DEAP: evolutionary algorithms made easy, *Journal of Machine Learning Research*, Vol. 13, 2171-2175

[20] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 182-197, doi:10.1109/4235.996017

[21] Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, Vol. 35, No. 2, 254-265, doi:10.1287/opre.35.2.254

[22] Solomon, M. M. (Northeastern University, Boston). VRPTW Benchmark Problems, from *http://w.cba.neu.edu/~msolomon/problems.htm*, accessed on 07-10-2019