Proceedings of the Eastern Asia Society for Transportation Studies, Vol.12,2019

A Novel Recombination Approach to Solve Prize Collecting Vehicle Routing Problem

Anurag TIWARI[a], Ching-Jung TING[b*], Gouthaman ELANGOVAN[c]

[a] *Institute of Business Management, Symbiosis International University, Nagpur, India*
[b] *Department of Industrial Engineering & Management, Yuan Ze University, Taoyuan, 32003, Taiwan*
[c] *Department of Information Management, Yuan Ze University, Taoyuan, 32003, Taiwan*
[a] *E-mail:* t*wr.anurag007@gmail.com*
[b] *E-mail: ietingcj@saturn.yzu.edu.tw*
[c] *E-mail: gouthaman_foru@yahoo.co.in*

Abstract: Prize collecting vehicle routing problem (PCVRP) is a variant of vehicle routing problem which is derived from the hot rolling production of the iron and steel industry. In PCVRP, it is not necessary to visit all the customers but every visited customer has a prize to be collected. Besides the vehicle capacity constraint, the total demand of served customers must be larger than the predetermined amount. The objective is a linear combination of minimization of the total travel distance, minimization of vehicle used, and maximization of collected prizes. In this paper we present a hybrid of two different crossover operators, edge recombination and edge assembly approach. We have used edge recombination operator (ERX), edge assembly operator (EAX) combining with block recombination approach to find the better solution. Computational result shows that the proposed methodology has the promising result.

Keywords: Prize Collection, Vehicle Routing Problem, Edge Assembly Operator, Edge Recombination Operator, Block Recombination

1. INTRODUCTION

The vehicle routing problem introduced by Dantzig and Ramser (1959) is to determine the optimum route sequence with minimum traveling distance for a fleet of homogeneous vehicle with capacity limitation to serve a set of customers. The prize collection vehicle routing problem **(**PCVRP) has some characteristics in common with VRP, like all the vehicles start from the single depot with same capacity and demand of all customers cannot exceeds the total capacity of the vehicle. However, the variant of VRP (PCVRP) is a more realistic problem in logistics. In this problem the number of vehicles is limited, prize will be collected by the vehicle which minimizes the total cost, all customers with different priorities or profits are not necessary to be visited compulsory, and penalty will be paid for unvisited customers. The objective of this problem is finding a schedule of vehicle to minimize the total distance and total cost and

[*] Corresponding author.

maximize the total prize collected by vehicle. The prize collecting traveling salesman problem (PCTSP) was proposed by Balas and Martin (1985) as a model for scheduling the daily operation of a steel rolling mill. The PCVRP has a strong background in practical real life industries. Tang and Wang (2006) proposed an iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem with non-given number of vehicles (PCVRP-NP). The hot-rolling batch scheduling method based on the prize collecting vehicle routing problem with predetermined number of vehicles (PCVRP-P) was developed by Zhang et al. (2009). The problem was solved by an enhanced particle swarm optimization algorithm (PSO). Stenger et al. (2013) proposed the prize collecting vehicle routing problem with single and multiple depots and non-linear cost. They developed an adaptive variable neighborhood search algorithm to solve the PCVRP.

Jia et al. (2013) developed a Pareto max-min ant system (P-MMAS) to solve a PCVRP with predetermined number of vehicles. The problem was also derived from the hot rolling batch scheduling process. Li and Tian (2016) proposed a two-level self-adaptive variable neighborhood search (VNS) for PCVRP-NP. Long et al. (2019) studied both PCVRP-P and PCVRP-NP. They first developed a Pareto-based evolutionary algorithm featuring a genetic algorithm combined with a local search strategy (HGLS) to solve the multi-objective PCVRP-P. The PCVRP-NP was decomposed into multiple PCVRP-P problems and solved it with HGLS. In this research, we studied the PCVRP-NP which is more difficult to handle than the PCVRP-P.

VRP and its variants are considered as NP-hard problems, so many researchers have done different studies to design efficient algorithms for solving the VRP and its variants. Azi et al. (2010) have proposed an exact method based on column generation to solve the vehicle routing problem. Dong et al. (2011) proposed a set of partitioning based exact algorithm for vehicle routing and scheduling problem of free pickup and delivery service in flight ticket sales companies based on set-partitioning model. Although the exact method can provide the optimal result but they are applicable for only small size instances and exact algorithm computational time is quite more than evolutionary algorithm. So many researchers have worked in the area of heuristic and metaheuristic algorithms that can solve the large size problem efficiently with less computation time. Chang and Wang (2009) developed a decomposition technique and a genetic algorithm (GA) to solve the vehicle routing problem with time windows. A multi-objective dynamic vehicle routing problem with fuzzy time windows have developed by Ghannnadpour et al. (2014).

In this paper we used a hybrid of edge recombination approach and edge assembly crossover to solve the PCVRP. Our main objective is to collect maximum prize with minimum cost. We have selected the unvisited customer by comparing the prize collect from that customer with the cost to visit the same customer. We have used an angle, demand and capacity based clustering method to divide the number of customers in different cluster. Number of cluster is based on number of vehicle used to serve the customer. We have used three different kinds of heuristic techniques at different stage of proposed methodology, like edge recombination approach (ERX) and edge assembly approach (EAX) to form the better cluster, block recombination approach to form the artificial chromosome and edge recombination operator to find the best sequence in final chromosome.

The remainder of the paper is organizes as follows. Problem description and formulation of the problem are presented in section 2. Section 3 describes the proposed solution methodology in details. Section 4 presents the experimental results drawn from a set of benchmark instances. Finally, we summarize our findings in section 5.

2. PROBLEM DESCRIPTION

The PCVRP is the variant of VRP. In this problem we consider a set of customers with different demands and prizes. All customers will be served by a single depot. We have assumed that all the vehicles have same capacity. Selection of customers is based on collected prize and total travelling cost. Each customer needs not to be visited compulsorily because the available vehicles are insufficient to visit all customers. However, the total demands of served customers must exceed a pre-specified value. In this paper we have two objectives, (1) minimization of total distance travelled by vehicles, and (2) minimization of total cost which includes prize collecting, vehicle cost and penalty for not visiting customers. These two objectives are summed up into a single objective as our objective function in eq. (1). The solution of the PCVRP determines a set of delivery routes which satisfy the customers' requirement and obtain the maximum prize collected by visited customers and minimum total cost of transportation traveled from the depot to the visited customers. All vehicles have same capacity and they deliver the goods from a single depot to all customers and return back to the depot. The number of vehicles is fixed. The mathematical model of the PCVRP can be expressed as fallows.

2.1 Notations

In this problem we have considered different customers with different demands and prizes. Customer has been served by a single depot. The following symbols are used to define the problem parameters.

Parameters:

C: Vehicle capacity

d_i: Demand of customer i

d_{ij}: Distance between customers i and j

f_c: Fixed cost of each vehicle if it is used

k_l: The lower bound for available number of vehicles

k_u: The upper bound for available number of vehicles

N: Set of customers, $N = \{0, 1, \ldots, n\}$, the depot is denoted as $\{0\}$

N_v: Set of customers that are served by vehicle v

P: Penalty for unserved customers

p_i: The prize obtained by serving customer i

r: The minimum ratio of the demands that must be served

Decision variables:

k: The number of vehicles used, $k_l \le k \le k_u$

$$x_{ij} = \begin{cases} 1 & \text{if customer } j \text{ is served immediately after } i \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ii} = \begin{cases} 1 & \text{if customer } i \text{ is not served in the optimal solution} \\ 0 & \text{otherwise} \end{cases}$$

2.2 Mathematical Model Formulation

The mathematical model of the PCVRP is as follow.

$$Min \quad \sum_{i\in N}\sum_{j\in N} d_{ij}x_{ij} - \sum_{i=1}^{n} p_i(1-x_{ii}) + (f_c k + P) \tag{1}$$

$$S.T. \quad \sum_{i=1}^{n} x_{0i} = k \tag{2}$$

$$\sum_{i=1}^{n} x_{i0} = k \tag{3}$$

$$\sum_{i\in N} x_{ij} \le 1 \qquad j = 1,\ldots,n \tag{4}$$

$$\sum_{j\in N} x_{ij} \le 1 \qquad i = 1,\ldots,n \tag{5}$$

$$\sum_{i\in N_v} d_i(1-x_{ii}) \le C \qquad v = 1,\ldots,k \tag{6}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S|-1 \qquad \forall S \subseteq N, 2 \le |S| \le N-1 \tag{7}$$

$$\sum_{i=1}^{n} d_i(1-x_{ii}) \ge r\sum_{i=1}^{n} d_i \tag{8}$$

$$k_l \le k \le k_u \tag{9}$$

$$x_{ij} \in \{0,1\} \qquad \forall i, j \in N \tag{10}$$

The objective function (1) has three components, the total distance travelled by the vehicles, prize collected from visited customers, and vehicle cost and penalty. Constraints (2) and (3) ensure that k vehicles must depart from and return to the depot. Constraints (4) and (5) state that each customer can be visited at most once by only one vehicle. Constraint (6) represents that the total demand of customers served cannot exceed the maximum capacity of a vehicle. Constraint (7) is employed to avoid the existence of sub tours in the vehicle routes. Constraint (8) ensures that the total served demand should not be less than the predetermined required amount. Equation (8) uses to eliminate the sub-tours for each vehicle route. Constraint (9) is the bound for the total number of vehicles. Constraint (10) is the integrity condition of the decision variables.

3. Proposed Methodology

In this paper we propose a hybrid of edge assembly and edge recombination techniques to calculate the minimum distance, minimum cost and maximal prize. To get the better chromosome we used the angle and capacity based allocation method to generate the initial chromosome. To find the best artificial chromosomes, we apply edge assembly approach on half of initial chromosome and apply edge recombination on other half artificial chromosomes. Once we get the better chromosome using edge recombination operator (ERX) we calculate the cost in two different ways. First we calculate the cost, while vehicle visited all the customers and second we calculate the cost after removing one customer. To select the un-visited customers we calculate the cost and prize collected from the customers. If cost is more than collected prize from those customers then we escape those customers. The overall procedure flowchart is shown in figure 1. Detailed steps are described in the following sections.

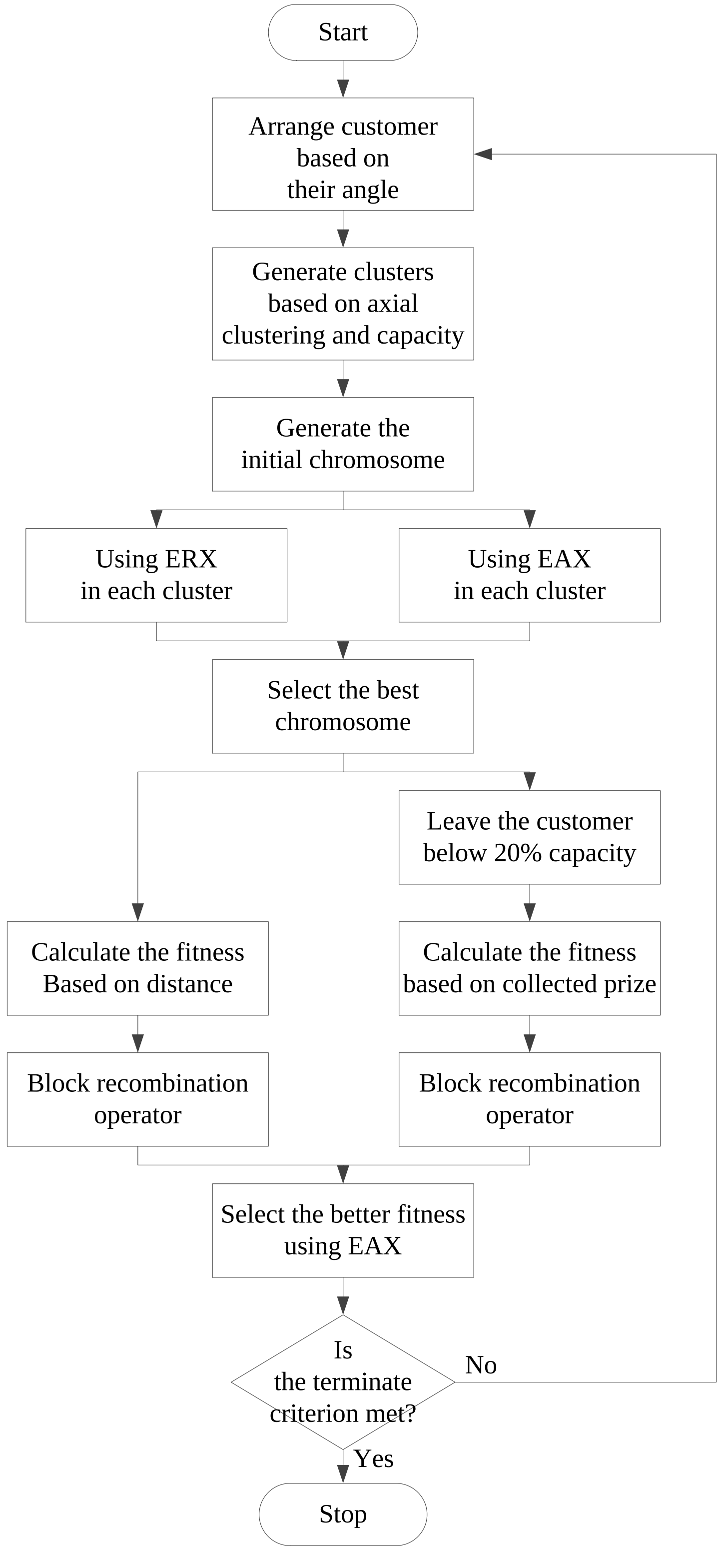

Figure 1: Proposed Methodology

3.1 Generate Initial Solution

Wang and Lu (2009) proposed a hybrid genetic algorithm that optimizes capacitated vehicle problem. We used an angle based approach to find the different cluster. To arrange the city according to angle, first we calculate the coordinate (*X*, *Y*) points for all the distribution point. The *X* and *Y* coordinate for all the customers relative to the depot are calculated as follows:

$$\begin{cases} X_i = x_i - x_0 \\ Y_i = y_i - y_0 \end{cases} \tag{11}$$

where (X_i, Y_i) is coordinates of the *i*th customer relative to the depot (x_0, y_0) and (x_i, y_i) is the original coordinates of *x* and *y* for the *i*th customer.

3.2 Generate the Cluster based on Axial Clustering and Capacity Allocation

We calculate the polar angle of different customers. The *X* and *Y* coordinate of all distribution points are converted to polar angle using the formula proposed by Wang and Lu (2009, 2010).

$$\phi_i = \begin{cases} \tan^{-1} \dfrac{y_i}{x_i} & X_i > 0, Y_i > 0 \\ \pi + \tan^{-1} \dfrac{y_i}{x_i} & X_i < 0 \\ 2\pi + \tan^{-1} \dfrac{y_i}{x_i} & X_i > 0, Y_i < 0 \end{cases} \tag{12}$$

where ϕ_i represents the polar angle of the *i*th customer. We sort the customers in both ascending and descending order to get better customers in each cluster. We sort the customers based on ϕ_i calculated from equation (12). The polar angle is sorted as follows:

$$\phi_{j-1} < \phi_j < \phi_{j+1} \tag{13}$$

where ϕ_j represents the *j*th customer after sorting (in either ascending or descending order). We consider load as an important factor, and the initial chromosome is generated based on the capacity allocation. Each vehicle has a maximum capacity limit. After getting the customers from polar angle we generate the new route based on capacity allocation, we used nearest neighborhood method to get the minimum distance between the two customers. The capacity allocation of the customers is as fallows from the equation proposed by Wang and Lu (2009).

$$\sum_{j=r_{k-1}+1}^{r_k} d_j \le C \le \sum_{j=r_{k-1}+1}^{r_k+1} d_j \qquad k = 1, \ldots, m \tag{14}$$

where C represents the vehicle capacity constraint, d_j is the demand of the jth customers, k is the number of vehicles, r_k is the last customer after sorting according to polar angle, and $r_0 = 0$.

3.3 Solution Representation

The chromosome represents the visiting sequence of the vehicles. We divide the chromosome into different clusters based on the previous section. The initial chromosome is the combination of all clusters which start from the first cluster to the last cluster. Each vehicle is used for each cluster. Figure 2 presents a solution example of 25 customers with 5 clusters. The first vehicle visits customers 6, 5, 3, 12, and 15 in sequence. The other four vehicles will visit the customers by the same way.

6 5 3 12 15	9 7 21 8 17	3 14 10 17 2 21	16 19 11 1 24 20	23 4 13
C1	C2	C3	C4	C5

Figure 2. A solution representation of 25 customers

3.3.1 Generation of better artificial chromosome using edge recombination operator

Edge recombination crossover approach is commonly used to solve the best sequence for travelling salesman problem (TSP) using genetic algorithm. It preserves the edge information between parent tours. Starkweather et al. (1991) proposed an enhancement to the edge recombination operator by preserving common edges of the parent tours. Steps of edge recombination operator are as follows.

- Step 1: Identify two parents as parent 1 and parent 2 and copy a segment of parent 1 to the offspring.
- Step 2: Construct an edge map using the edge information in parent 2, segment of parent 1 not copied to the offspring.
- Step 3: Set the last customer in a segment as a first customer.
- Step 4: Remove all occurrences of the current customer from all the edge lists.
- Step 5: If the current customer has elements in its list, return to step 6, otherwise go step 8.
- Step 6: Select the share edges to be the current customer, if more than one edge is available, then return step 4. If there is no share edge go to step 7.
- Step 7: Find the nearest customer in the edge list, and it will become the current customer. Go to step 4.
- Step 8: If no customer left, then end. Otherwise choose an unvisited customer which is nearest to the current customer, and go to step 4.

Figure 3 presents an example of the edge recombination operator with 12 customers. A fixed segment (3, 4, 5) of parent 1 is inherited to the offspring. Customer 5 is the last one in the segment. Three customers, 2, 6, and 9, are the candidates for the next customer based on the edge table as shown in table 1. Customer 2 is chosen because the edge is the shortest. The remaining customers are selected by the same process until the total number of routes is reached. In this problem we leave a customer to get the minimum cost. We leave customer based on the travelling distance and demand. If prize collected from the customer is less than the total travelling cost and penalty, we leave the customer to find the best sequence.

1 2 3 4 5 6 7 8 9 10 11 12	6 7 8 3 12 11 10 2 5 9 14
(a) Parent 1	(b) Parent 2
3 4 5	3 4 5 2 10 11 12 1 9 8 7 6
(c) A fixed segment of parent 1	(d) Offspring 1

Figure 3. An example of edge recombination operator

Table 1. The edge table

Node	Edge list	Node	Edge list	Node	Edge list
1	12, 2, 9	5	2, 6, 9	9	8, 10, 1
2	1, 10	6	-7	10	9, -11, 2
3	-	7	-6, -8	11	-10, -12
4	-	8	-7, 9	12	-11, 1

3.4 Generation of Better Chromosome Using Edge Assembly Operator

Nagata and Kobayashi (1997) introduced a genetic algorithm, which employs a form of elitist tournament selection. Two parents are randomly selected, without replacement, from the population and recombined using crossover. The two parents and the resulting child are then compared, and the individual with the best fitness is passed to the next generation. This procedure is repeated to produce all *N* members of the next generation. The edge assembly operator (EAX) defines recombination as an iterative procedure as follows (Nagata, 2007).

Step 1: Identify two parents as parent 1 and parent 2 and merge them into a graph.

Step 2: Partition all edges of the graph into AB-cycles, where and AB-cycle is a cycle such that edge of tour A and edges of tour B are alternatively linked.

Step 3: Construct an E-set which is the union of AB-cycles.

Step 4: Generate an intermediate solution from parent 1 by removing the edges of parent 1 and adding the edges of parent 2 in the E-set. One or more sub-tours may be found in the intermediate solution.

Step 5: Generate an offspring by connecting all sub-tours into a tour.

Step 6: If other offspring solution can be generated, then go to step 3. Otherwise, stop.

3.5 Block Recombination

We rank the entire cluster and calculated the fitness value of each cluster. We selected the entire cluster with higher fitness and combined it according to their rank. Block recombination technique provide a best chromosome with minimum fitness value. Figure 4 describes the block recombination approach in an example with 25 customers and 5 vehicles.

In figure 4, five chromosomes are compared with the same clusters. Each cluster with different permutations provides different costs of a cluster. By selecting the lowest cost among different clusters and recombine them to obtain a better solution. In this example, the best solution is obtained from C_{51}, C_{22}, C_{53}, C_{44}, and C_{25}, respectively. The new solution after the block recombination is shown in figure 5.

4 5 6 10 11	9 7 12 13 1	2 3 14 15 21	8 16 19 20 25	24 21 23 17 18
C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
4 10 11 6 5	7 9 13 12 1	2 3 15 21 14	16 25 8 25 20	21 23 17 18 24
C_{21}	C_{22}	C_{23}	C_{24}	C_{25}
4 5 10 6 11	9 7 12 1 13	2 3 15 14 21	8 16 20 19 25	23 21 24 17 18
C_{31}	C_{32}	C_{33}	C_{34}	C_{35}
11 10 5 4 6	13 9 12 7 1	21 3 15 14 2	20 19 16 8 25	18 21 23 17 24
C_{41}	C_{42}	C_{43}	C_{44}	C_{45}
11 6 5 4 10	12 9 7 13 1	21 3 14 15 2	8 19 8 16 25	18 23 24 21 17
C_{51}	C_{52}	C_{53}	C_{54}	C_{55}

$C_{11} = 20$	$C_{12} = 30$	$C_{13} = 23$	$C_{14} = 38$	$C_{15} = 38$
$C_{21} = 28$	$C_{22} = 24$	$C_{23} = 30$	$C_{24} = 29$	$C_{25} = 27$
$C_{31} = 32$	$C_{32} = 25$	$C_{33} = 42$	$C_{34} = 22$	$C_{35} = 28$
$C_{41} = 42$	$C_{42} = 38$	$C_{43} = 31$	$C_{44} = 19$	$C_{45} = 30$
$C_{51} = 18$	$C_{52} = 41$	$C_{53} = 22$	$C_{54} = 23$	$C_{55} = 29$

Figure 4. An example of block recombination

11 6 5 4 10	7 9 13 12 1	21 3 14 15 2	20 19 16 8 25	21 23 17 18 24
C_{51}	C_{22}	C_{53}	C_{44}	C_{25}

Figure 5. The solution after the block recombination

4. COMPUTATIONAL EXPERIMENT

To test the performance of the proposed approach, we have evaluated the performance on different benchmark problems for the capacitated VRP that can be obtained from the website: http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm.old. We have selected different numbers of customers and numbers of vehicles ranging from 21 to 100 and 2 to 10, respectively. For prize collection we have generated the data randomly. We have set the fixed cost 1000, the prize p_i is randomly generated in [1, 100]. The demand of all customers that must be satisfied is generated in [0.2, 0.6, 0.8]. For each benchmark problem, we generate 10 instances. We have tested the result on E instances, A instances, and B instances, for small, and large size problems. The number of iterations is set at 1000 and population size is 100 after a preliminary test on parameter setting test.

The proposed algorithm was coded in Microsoft Visual Studio 2010 C# and implemented on a personal computer with Intel Core i5-2400 3.10 GHz CPU and 8GB RAM under Windows operation system. All the instances were conducted for 10 independent runs. The results and comparison with block recombination approach are presented in table 2.

In table 2, the first column is the instance name. The meaning of the instance name such as A-n45-k7 is that the number of customers is 45, and the number of vehicles available is 7.

Columns 2 and 3 are the results of visiting all customers of PCVRP in travelled distance TD and average total cost ATC of 10 instances of each problem from the literature. Columns 4-9 are the results by allowing a customer being unvisited. Columns 4 and 5 are the results for only using the block based approach, while columns 6 to 9 are the results for edge assembly only and our hybrid approach, respectively. From table 2, we can find that the hybrid approach with ERX and EAX can provide better results than those by the EAX or by block based approach.

Table 2. Summary of benchmark problem results

Instance	PCVRP		block based		EAX		EAX/ERX	
	TD	ATC	TD	ATC	TD	ATC	TD	ATC
A-n45-k7	1146	757.2	1168	898.67	1132	862.65	1132	862.67
A-n65-k9	1174	603.2	1146	589.60	1071	514.60	1071	514.60
A-n69-k9	1159	555.8	1151	657.67	1143	649.67	1143	649.67
A-n80-k10	1736	988.8	1732	1104.27	1720	1092.27	1711	1083.27
B-n35-k5	955	696.6	892	795.74	866	769.74	849	752.74
B-n41-k6	849	486.6	846	574.27	829	557.27	811	539.27
B-n45-k6	678	303.8	670	404.07	662	396.07	643	377.07
B-n50-k7	714	291.0	706	365.67	692	351.67	692	351.67
B-n78-k10	1247	625.6	1124	525.00	1078	479.00	1061	462.00
E-n76-k10	830	209.8	796	184.87	728	116.87	728	116.87
E-n76-k14	1028	406.4	989	429.27	860	300.27	856	296.27
E-n101-k8	817	161.2	803	170.14	749	116.14	749	116.14
Ave.	1027.8	507.17	1001.9	558.27	960.8	517.19	953.8	510.19

5. CONCLUSION

In this paper we proposed a novel hybrid approach that used edge recombination operator, edge assembly operator, and block recombination to solve the prize collecting vehicle routing problem. PCVRP is the variant of the VRP that not necessary to serve all customers. We select the customer based on the prize collected and travelling cost. The hybrid approach first used the angle based approach to form different clusters and then combined the clusters with block recombination. The proposed approach was tested with benchmark instances from the literature. The number of clusters is the same as the number of vehicles used in benchmark datasets. The results were compared with other well-known heuristic techniques. It is found that our approach performs better than the previous proposed approaches. We believe that the proposed hybrid approach can be applied to other vehicle routing problems. In the future we can implement some new local search techniques to solve the PCVRP problem.

REFERENCES

Azi, N., Gendreau, M. and Potvin, J.Y. (2010) An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3), 756-763.

Balas, E. and Martin, C.H. (1985) ROLL-A-ROUND: Software package for scheduling the

rounds of a rolling mill. Balas and Martin Associates, Pittsburgh, USA.
Balas, E. (1989) The prize collecting traveling salesman problem. *Networks*, 19(6), 621-636
Cheng, C.B. and Wang, K.P. (2009) Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. *Expert Systems with Applications*, 36(2), 7758-7763.
Dantzig, G.B. and Ramser, J.H. (1959) The truck dispatching problem. *Management Science*, 6(1), 80-91.
Dong, D., Tang, J.F., Lai, K.K. and Kong, Y. (2011) An exact algorithm for vehicle routing and scheduling problem of free pickup and delivery service in flight ticket sales companies based on set-partitioning model. *Journal of Intelligent Manufacturing*, 22(3), 759-774.
Ghannadpour, S.F., Noori, S., Tavakkoli-Moghaddam, R. and Ghoseiri, K. (2014) A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. *Applied Soft Computing*, 14C, 504-527.
Jia, S.J., Yi, J., Yang, G.K., Du, B. and Zhu, J. (2013) A multi-objective optimisation algorithm for the hot rolling batch scheduling problem. *International Journal Production Research*, 51(3), 667-681.
Li, K. and Tian, H. (2016) A two-level self-adaptive variable neighborhood search algorithm for the prize-collecting vehicle routing problem. *Applied Soft Computing*, 43, 469-479.
Long, J.Y., Sun, Z.Z., Pardalos, P.M., Hong, Y., Zhang, S.H. and Li, C. (2019) A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem. Information Sciences, 478, 40-61.
Nagata, Y. and Kobayashi, S. (1997) Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. In *Proceedings of the 7th International Conference on Genetic Algorithms*, pp. 450-457.
Nagata, Y. (2007) Edge assembly crossover for the capacitated vehicle routing problem. *Lecture Notes in Computer Science*, 4446, 142-153.
Starkweather, T., McDaniel, S., Mathias, K., Whitley, D. and Whitley, C. (1991) A comparison of genetic sequence operators. In *Proceedings of the 4th International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 69-76.
Stenger, A., Schneider, M. and Goeke, D. (2013) The prize-collecting vehicle routing problem with single and multiple depots and non-linear cost. *EURO Journal on Transportation and Logistics*, 2(1-2), 57-87.
Tang, L.X. and Wang, X.P. (2006) Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem, *International Journal of Advanced Manufacturing Technology*, 29(11-12), pp 1246-1258.
Wang, C.H. and Lu, Z.J. (2010) An effective evolutionary algorithm for the practical capacitated vehicle routing problems, *Journal of Intelligent Manufacturing*, 21(4), 363-375.
Wang, C.H. and Lu, Z.J. (2009) A hybrid genetic algorithm that optimizes capacitated vehicle routing problems. *Expert Systems with Applications*, 36(2), 2921-2936.
Zhang, T., Chaovaitwongse, W.A. and Zhang, Y.J. (2009) The hot-rolling batch scheduling method based on the prize collecting vehicle routing problem. *Journal of Industrial and Management Optimization*, 5(4), 749-765.