



## **A Neighbourhood Search for Artificial Bee Colony in Vehicle Routing Problem with Time Windows**

**Salah Mortada<sup>1\*</sup>      Yuhanis Yusof<sup>1</sup>**

*<sup>1</sup>School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Sintok, Kedah, Malaysia*

*\* Corresponding author's Email: salah\_sms099@yahoo.com*

---

**Abstract:** Artificial bee colony (ABC) is one of the widely used swarm intelligence algorithms in solving combinatorial optimization problems. In this study, the existing Modified ABC (MABC) algorithm is revised to solve the vehicle routing problem with time windows (VRPTW). The reason is that even though MABC is reported to be successful, its exploitation process lacks a selection neighborhood structure during the intensification process. The proposed algorithm, termed as enhanced Modified ABC (E-MABC), offers a neighborhood search that exchanges neighborhood structure between two different routes in the same solution, rather than one route. To evaluate the effectiveness of E-MABC, experiments are performed on 56 instances of VRPTW. The results of E-MABC are compared against the ones produced using MABC and other metaheuristic algorithms. Based on the total travel distance and number of vehicles, the proposed E-MABC is shown to be a solution for VRPTW. The interchange neighborhood search, which is implemented by bees during the exploitation process, improves the solution quality, hence producing an optimal outcome. The proposed E-MABC results are better as compared to MABC in terms of the total travel distance by 71.42 % and the number of vehicles by 35.71%.

**Keywords:** Optimization, Metaheuristic, Swarm intelligence, Local search.

---

### **1. Introduction**

Combinatorial optimization problems (COPs) are the most important category for optimizing problems because of the large diversity of real-life problems that can be overcome in the engineering, science, and industry fields by studying them. COPs aim to determine an ideal solution for a particular problem case among all available solutions on discrete space. Many categories can be defined for the practical problems of COPs, namely, vehicle routing problem (VRP), traveling salesman problem, and scheduling problem. Among these categories, VRP is mostly utilized to verify the performance of any proposed algorithms for COPs and related fields (i.e., operations research, logistics, and transportation management). One of the challenges in transportation management is the VRP [1], which was initially referred to as the "truck dispatching problem." VRP aims to structure a set of directions to serve numerous geographically scattered customers at nominal cost

(i.e. to minimize travel distance or time) while fulfilling different types of constraints [2]. A productive transportation management may help solve VRP and essentially minimize cost, pollution, and heavy traffic jam. To date, various studies deploying artificial intelligence and data mining methods have been reported to contribute in VRP issues [3]. Several extensions of VRP have been presented over the years, and the vehicle routing problem with time windows (VRPTW) is one of the most commonly studied extension. The reason is because of the use of time window factor that is required in most real-world situations [4]. The operational of VRPTW is highly dependent on time window, where it restricts the time at which a vehicle can serve a customer on the basis of the customer's availability. The timeframe for each customer assumes that if a vehicle reaches the customer too early, then the vehicle must wait before the service can be provided.

Several algorithms have been developed to address VRPTW, namely, exact and approximate (heuristic or meta-heuristic) algorithms. These algorithms can ensure the best solutions by searching every possible solution in the search space [2]. Thus, the exact algorithm is a suitable alternative for handling small operations ranging from 25 customers to 50 customers [5, 6]. With respect to large operations, such as 100 customers and above, the incorporation of the entire search space while searching for the best solution is impossible because it requires an unreasonable amount of time [6]. This size incorporates an extraordinary possible solutions number; hence, obtaining the ideal solution by using exact algorithms can be difficult [7]. Moreover, heuristic algorithms are utilized to handle large-sized VRPTW. “A heuristic algorithm (or simply heuristic) is a method which seeks good (i.e., near-optimal) solutions at a reasonable computation cost without being able to guarantee optimality and feasibility; unfortunately, stating how close to optimality a particular solution is may even be impossible” [8]. Subsequently, these algorithms are useful alternatives that have progress in acquiring outcomes at a satisfactory time structure in COPs. Heuristic algorithms are developed to address specific issues; thus, these algorithms cannot tackle all COPs [9, 10].

In addition, meta-heuristic algorithms are employed to handle the above-mentioned issue [11, 12]. Such an algorithm can be characterized by their population-based features. These features replicate nature and lead emerging behavior through the joint action of individual agents in the swarm [13–16]. Previous studies focused on meta-heuristic algorithms in solving problems associated with optimization. These algorithms provide efficient solutions within nominal time periods [17–19]. One of the optimization algorithms that are commonly used in the routing problem is Artificial Bee Colony (ABC) which is inspired from honey bee swarms’ intelligent foraging behavior [20]. ABC has been utilized for several problems with great success [21], and when it is employed to tackle the capacitated vehicle routing problem, the suggested algorithm is also competitive to other population-based algorithms. Furthermore, A. Alvarado-Iniesta, J. L. Garcia-Alcaraz, M. I. Rodriguez-Borbon, and A. Maldonado [22] argued that when ABC is employed to VRP problems, it yields promising results in addition to decreasing computational time. However, research on ABC for VRPTW, particularly focusing on the measures of total distance and number of vehicles, is limited. Therefore, this study investigates on how Modified ABC (MABC) can be adapted to solve VRPTW. The flaw of MABC local search is

addressed by introducing an enhanced MABC algorithm (E-MABC) that deploys different neighborhood operations to enhance the solution. The main advantage of the proposed E-MABC algorithm is that it’s neighborhood search operations (i.e two interchange, cross-exchange, and 2-opt) exchanges neighborhood structure between two different routes in the same solution, rather than in the same route. This helps the algorithm to produce good results that can minimize the total travel distance and the number of vehicles which are the main objectives of VRPTW. The proposed algorithm is then implemented on the benchmark datasets of VRPTW.

The remaining parts of this paper are organized as follows: Section 2 start with related research. Section 3 explains the description and mathematical equations relevant to VRPTW. Section 4 and 5 provide method and the proposed E-MABC of this study. Section 6 present the experimental setup and results, respectively. Section 7 concludes the discussion and provides recommendations for future direction.

## 2. Related research

Numerous population-based algorithms have been utilized to tackle VRPTW. For example, B. Yu, Z. Z. Yang, and B. Z. Yao [23] integrated Ant Colony Optimization (ACO) and Tabu Search (TS) and to enhance the ACO performance, a neighborhood operation is presented on the basis of TS for maintaining the diversity of ACO and exploring new solutions. The shortcoming, though, is that the search could be trapped in a certain region of the search space if there are no changes to the adjacent solutions. Thus, generates a longer travel distance and larger number of vehicles.

Z. Ursani, D. Essam, D. Cornforth, and R. Stocker [24] provided a localized optimization framework that includes two stages: de-optimization and optimization. De-optimization is performed on a problem as a whole, whereas optimization is performed on the parts of the problem. This structure takes VRPTW as domain space and Genetic Algorithm as the technique for optimization. The Localized Genetic Algorithm provides better solutions than other heuristics for small-scale VRPTW problems [24]. The productivity of swap, insertion, inversion, and partially mapped crossover shows that all of them can help the evolution of solutions to a certain extent. Nonetheless, these operators all behave using random searching, which cannot fully utilize the existing information of the current solution. The well-performed sub-routes in the current solution could easily be destroyed by

these operators unintentionally and in this case, affects total travel distance. Y. J. Shi, F. W. Meng, and G. J. Shen [25] proposed ABC-T to deal with VRPTW. ABC-T used the ABC algorithm and enhanced the global search ability with tournament selection. The proposed ABC-T algorithm was evaluated on Solomon's R102 datasets, and the outcomes were compared with well-known algorithms. The outcomes indicated that the suggested method is more effective in solving VRPTW than other algorithms in the literature. Nonetheless, this method of selection does not guarantee the reproduction of best solution. Thus, does not minimize the total travel distance and the number of vehicles.

O. E. Nahum, Y. Hadas, and U. Spiegel [26] solved the multi-objective VRPTW problem by using the ABC algorithm combined with the vector evaluated method called VE-ABC and tested it on Solomon benchmark problems. The VE-ABC showed that in cases with 25 customers or 50 customers, VE-ABC could find results better than well-known algorithms regarding the set of vehicles utilized. VE-ABC revealed a rise in the set of iterations and a high limit rate and this makes the solution near to the best result algorithms for datasets with 100 customers. However, the result is on the basis of previous runs. Therefore, it consumes large computational time. Also, the results of the VE-ABC need to be increased in terms of total travel distance and the number of vehicles to clarify the position of VE-ABC further. S. Yu, C. Tai, Y. Liu, and L. Gao [27] attempted to enhance the efficiency of ABC for VRPTW by employing crossover and mutation operators with a new adaptive strategy. VRPTW benchmark problems were utilized, and the outcomes indicated the performance of the proposed method for resolving VRPTW. Furthermore, the outcomes of the problem of western-style food delivery routing in Dalian City showed that the proposed method is a viable approach for resolving a problem. B. Yao, Q. Yan, M. Zhang, and Y. Yang [28] improved ABC beads on crossover operation copied from the GA and used a scanning approach for solving VRPTW problems. In the standard VRPTW test, the contrast with well-known algorithms confirms the ability of the proposed IABC algorithm. The comparison of the results of ABC-S, ABC-C, and IABC revealed that ABC efficiency for VRPTW can be enhanced by the integration of crossover operation and scanning strategy. With the aim of improving the accuracy of ABC on VRPTW. Nevertheless, the crossover operation may increase the computation time and leads to a slow convergence as it requires more time to search the crossover nodes [27] [28]. Furthermore,

both studies were on small instances and cannot ensure its validity in large instances.

M. Alzaqebah, S. Jawarneh, H. M. Sarim, and S. Abdullah [32] investigated the use of bees algorithm (BA) for VRPTW. BA was applied to resolve Solomon's VRPTW benchmark problems. The outcomes showed that BA produced quality solutions and accomplished comparable results when compared with best-known results from the literature. The main drawback of the BA is that it is parameter dependent, so each dataset may need various values of parameters. In addition, using random neighborhood searching method, which cannot completely utilize existing experience of the current solution, influences the total travel distance and the number of vehicles. K. C. Tan, L. H. Lee, and K. Ou [33] explore the hybridizations of Genetic Algorithm and local search improvement termed as (HGA) for better performance in VRPTW. HGA is used to resolve Solomon's benchmark problems. Results achieved are encouraging and demonstrate the performance of the HGA. However, as can be shown, because of its large population size and its incremental evolutionary style, HGA takes the longest time to operate as anticipated and HGA cannot show optimality in terms of both total travel distance and number of vehicles.

M. Alzaqebah, S. Abdullah, and S. Jawarneh [30] introduced MABC to address the VRPTW. They enhanced the traditional ABC algorithm to improve the solution quality. The high exploration capability of ABC affects and slows-down full convergence velocity, which may be attributed to the scout bees replacing exhausted solutions with generated ones, randomly. In the MABC algorithm, an archive solutions list is utilized by the scout bees (SB) to memorize the exhausted sources from the exploitation process. Then, the SB selects a source from the archive list on the basis of the roulette wheel method and replaces it with a new source by using random routes selected from the best solution found. MABC was tested and compared with the standard ABC and best-known results on a traditional set of 56 VRPTW benchmark dataset. The simulated outcomes demonstrated that MABC provides better results than the standard ABC. However, the deployed neighborhood operations cannot exploit the search space intensely, due to neighborhood search that exchanges neighborhood structure between customers on the same route in the solution. In this case, it affects the final vehicle routing results by generating longer travel distance with larger number of vehicles for operation.

The proposed E-MABC algorithm is implemented to solve the above-stated shortcomings

hence leading to an optimized outcome of minimal travel distance and less number of vehicle to be utilized.

### 3. Description and mathematical formulation of VRPTW

VRPTW could be drawn as an undirected graph;  $G = (V, E)$ . Assume  $V$  represents the number of vertices  $V = \{0, 1, 2, \dots, n\}$ , where vertex 0 denotes the depot, and vertices 1, 2, ... n denotes the customers. Assume  $E$  is the number of links, where  $E = \{(i, j) : i \neq j \text{ and } i, j \in V\}$ . Each link is related to a travel distance, which can be denoted as  $c_{ij} = c_{ji}$  and  $c_{ij} > 0$ . Except for the depot, each customer has a particular service time and demand that must be known earlier before starting. A static set of required vehicles ( $v$ ) exists in VRPTW. Given that the capacity of each vehicle is identified in advanced, if the vehicle reaches the customer too early, then the vehicle must wait for the service to be served. Otherwise, the vehicle fails in serving the customer when it reaches after the customer's time window is finished. Consider the given variables:

- $n$  number of customers;
- $v$  number of vehicles;
- $q_i$  demand of customer  $c_i$ ;
- $Q_k$  capacity of  $k$  vehicle;
- $s_i$  service duration of customer  $c_i$ ;
- $t_i$  arrival time at customer  $c_i$ ;
- $t_{ij}$  travel time between customer  $c_i$  and  $c_j$ ;
- $W_i$  waiting time of customer  $c_i$ ;
- $e_i$  start of the time of customer  $c_i$ ;
- $l_i$  end of the time of customer  $c_i$ .

The formulation of VRPTW can be presented as shown below:

$$X_{ij}^k = \begin{cases} 1 & \text{if vehicle } k \text{ travels from } c_i \text{ to } c_j \\ 0 & \text{otherwise} \end{cases}$$

$$y_i^k = \begin{cases} v & \text{if customer } c_i \text{ is served by vehicle } k \\ n & \text{otherwise} \end{cases}$$

The primary problem feature is shown in Eq. (1),

$$f(S) = \min \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^v t_{ij} \times X_{ij}^k$$

subject to

$$\sum_{i=0}^n X_{ij}^k = y_j^k, \forall k = 1, \dots, v, \forall j = 1, \dots, n \quad (2)$$

$$\sum_{j=0}^n X_{ij}^k = y_i^k, \forall k = 1, \dots, v, \forall i = 1, \dots, n \quad (3)$$

$$\sum_{i=0}^n y_i^k \times q_i \leq Q_k, \forall k = 1, \dots, v \quad (4)$$

$$\sum_{k=1}^v y_i^k = 1, \forall i = 1, \dots, n \quad (5)$$

$$\sum_{k=1}^v y_0^k = v \quad (6)$$

$$t_i + W_i + s_i + t_{ij} = t_j, \forall i, j = 0, 1, 2, \dots, n, i \neq j \quad (7)$$

$$e_i \leq t_i \leq l_i, \forall i = 0, 1, 2, \dots, n \quad (8)$$

$$W_i = \max \{e_i - t_i, 0\}, \forall i = 0, 1, 2, \dots, n \quad (9)$$

Eq. (2) and (3) confirm that each vehicle can start and end from any customer once it completes serving the specific customer. Equation 4 checks that the capacity of vehicle is unviolated. Eq. (5) ensures that the service for each customer is fulfilled only once. Eq. (6) confirms that the beginning point of each vehicle is from the depot. Eq. (7), (8), and (9) verify the time limitations to validate that no time window is exceeded.

### 4. Method

In this section, the modification is on the intensification process of the MABC algorithm presented by M. Alzaqebah, S. Abdullah, and S. Jawarneh [30]. To improve the algorithm performance, this study increases the intensification in a promising region for fully exploring the search space with the hope to find the best solutions by using three neighborhood operations. These operations iteratively change the current solution by performing neighborhood operations for improved neighboring solutions. Basically, a neighborhood comprises a set of solutions that can be achieved from the present one by swapping a subset of routes between solutions. The neighborhood operations are described below:

**Two interchange:** customers are interchanged between routes by following the local search procedure. The structure for customer search can be identified by combining routes either randomly or systematically. That is, a maximum of two customer vertices can be interchanged between routes. For example, eight operators are defined: (0,1), (1,0), (1,1), (0,2), (2,0), (2,1), (1,2), and (2,2); operator (1,2) indicates that on a route combination (R1, R2), one customer moves from R1 to R2, and two customers move from R2 to R1. Therefore, a new

solution is produced by changing some links connecting customers in the current solution. Subsequently, the new solution is assessed on the basis of the objective function (distance). If the new solution is better, then it replaces the previous solution.

**Cross-exchange:** local search procedure eliminates one link from two different routes and reconnects the included vertices by cross-linking the heads and tails. If two arcs (R1, R2) and (R1\*, R2\*) from two distinguished routes are chosen, the cross-exchange neighborhood reconnects the vertices as (R1\*, R2) and (R1, R2\*) where two distinguished routes are formed. The improved solutions are be selected.

**2-opt:** the local search procedure chooses two random customers and cuts the route into two segments. Afterward, the two are reconnected to form a new route. Only improved solutions are accepted during the 2-opt.

## 5. Proposed E-MABC algorithm for VRPTW

E-MABC starts with initial population solutions and iteratively improvises solutions for a given problem instance. In each iteration, E-MABC improvises the solutions by using employed bees, onlooker bees, and scout bees. The E-MABC algorithm has five phases as follows:

**Initialization phase:** the initial population solutions of the algorithm are randomly created by verifying the constraints of VRPTW. Subsequently, the fitness value of each solution is computed, and the best solution is identified.

**Employed bees (EB) phase:** each EB is randomly assigned with a solution from the population. Then, every EB seeks to improve its solution by applying random neighborhood operation for the selected solution in its memory. The current solution is modified on the basis of the amount of nectar (i.e., fitness value distance) of the new solution. The neighborhood operation is performed on each solution to explore the local neighborhood and to improve the current solution.

**Onlooker bees (OB) phase:** each OB is responsible for choosing promising food sources on the basis of the information provided by the EB. The food sources are selected according to a probability, which is computed as shown in Eq. (10), below:

$$p = \frac{f_i}{\sum_{i=1}^N f_j} \quad (10)$$

Where  $f_i$  is the fitness value of the  $i$  solution and  $N$  is the number of food sources in the colony. The better solution is the one with a higher probability ( $p$ ) of the  $i$  food source. Each OB chooses to adjust the food source by applying the random neighborhood operation of the current solution in its memory on the basis of the amount of nectar (i.e., fitness value distance). The neighborhood operation is performed on the selected solution to explore the local neighborhood and to enhance the current solution. If a current solution is unimproved in a predefined number of iterations called limit, then the solution is supposed to be abandoned and OB is converted into an SB.

**Scout bees (SB) phase:** the SB phase is implemented to replace abandoned solutions. If the number of trials for a food source is greater than a predefined value limit, an SB generates a new solution by choosing a random route from the best solution and replacing it with a random route in the newly discovered solution. In this manner, the SB learns from the best solution as the best solution because such a solution contains some beneficial information.

Termination process: termination criterion of E-MABC. If the termination criterion is satisfied, then E-MABC stops and utilizes the best solution found so far. Otherwise, EB, OB, and SB phases are repeated. Flowchart of the E-MABC algorithm is shown in Fig. 1.

## 6. Experimental setup

The proposed E-MABC algorithm has been tested on VRPTW datasets [31]. This benchmark has 56 problem instances that are distributed into six groups: R1, R2, C1, C2, RC1, and RC2, as demonstrated in Table 1. The instances in the C1 and C2 categories have clustered customers. The instances in the RC1 and RC2 categories have partially clustered customers and are partially have random. The instances in the R1 and R2 category randomly distributed customers.

Furthermore, instances specified with C1, R1, and RC1 have a tight time window width (short scheduling horizon), whereas instances specified with C2, R2, and RC2 have a large time window width (a long scheduling horizon allowing more than one customer to be fulfilled by the same vehicle). Each test conducts 31 runs independently. The exact parameter settings, as utilized by MABC [30], are deployed in this work and are demonstrated in Table 2.

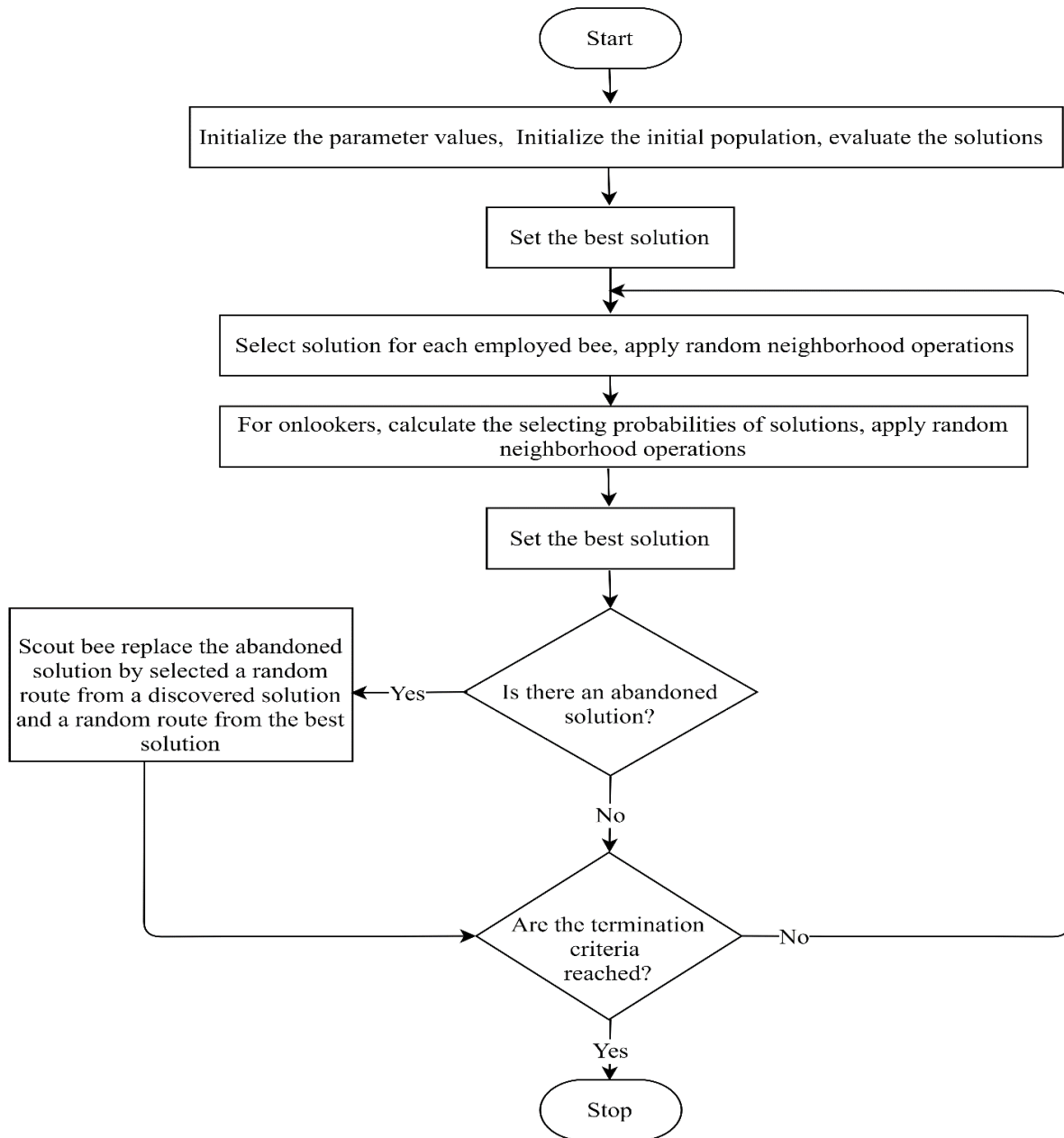


Figure 1. Flowchart of the E-MABC algorithm for VRPTW

Table 1. Features of Solomon’s VRPTW benchmark problems

Problem instance	No. of customers	Vehicle capacity	Customers distribution	Width of TW
R1	100	200	Random	Small
R2	100	1000	Random	Large
C1	100	200	Cluster	Small
C2	100	700	Cluster	Large
RC1	100	200	Random-Cluster	Small
RC2	100	1000	Random-Cluster	Large

Table 2. Parameter settings for E-MABC algorithm

Parameter	Value
Number of iterations	1000
Population size = EB=OB	50
Limit	100

## 6.1 Experimental results

This sub-section discusses the experimental results obtained by the proposed E-MABC algorithm. Two sets of experiments are constructed to investigate the performance of the proposed E-MABC algorithm for VRPTW. The first experiment is to evaluate the effect of neighborhood operations on ABC where a comparison between MABC and E-MAABC is made. The second experiment involves four well-known optimization algorithms in the literature, namely, ACO-T, VE-ABC, BA, and HGA. The results of the nonparametric Friedman test with Holm's post-hoc test is used for statistical analysis. This statistical test aims to determine if the proposed algorithm is significantly better than other optimization algorithms for the average distance and number of vehicles.

### 6.1.1. Compaction between E-MABC and MABC

Table 3 provides the best traveled distance, average, and gap from the best-known solutions for E-MABC over 31 runs. Table 3 presents that for the traveled distance, E-MABC is better for 40 instances (out of 56) than MABC. E-MABC is also at the same par with MABC on 14 instances. In terms of the number of vehicles (N.V), E-MABC obtains better results in 20 instances (out of 56) and utilizes the similar number of vehicles as MABC for 36 instances. The results of the total traveled distance and number of required vehicles are listed in Table 3. The E-MABC algorithm produces solutions with the lowest distance for eight instances (out of 56). In comparison to best-known solutions, the E-MABC algorithm produces competitive results for 22 instances with the same set of vehicles. These results have proven that neighborhood operations can be used to exploit the local search in the searching process for attaining significant results.

### 6.1.2. Comparison between E-MABC and other optimization algorithms

In this experiment, the best results of E-MABC are compared with those of well-known optimization algorithms, as shown in Table 4. Several algorithms [23], VE-ABC [26], BA [32], and HGA [33].

Table 4 depicts data showing that E-MABC achieves the best results in 23 instances and is equally good in six instances when compared with ACO-T. E-MABC outperforms VE-ABC as it is better in 45 instances and obtains equal results in three instances. E-MABC also achieves better outcomes in 41

instances and equal results in 14 instances than BA. The comparison with HGA reveals that E-MABC is better in 41 instances. The proposed E-MABC algorithm can accomplish the solutions close to the best-known for groups C1 and C2. In addition, the E-MABC algorithm can obtain comparable results for groups R1, R2, RC1, and RC2 upon comparison with other algorithms.

Table 5 demonstrates the results averaged over all problem instances in VRPTW. Data in this table include the average of the total traveled distance and average number of required vehicles. The proposed E-MABC exhibits the best distance among other optimization algorithms. ACO-T obtains the best result for a number of required vehicles. E-MABC achieves the second-best in the average number of required vehicles in the overall instances of VRPTW.

The average distance ranking is assigned against the average number of vehicles ranked over the 56 instances. A low average rank indicates a better algorithm performance. Fig. 2 proves that the results obtained by E-MABC dominate other optimization algorithms when seeing the distance and number of required vehicles. E-MABC only performs slightly worse than ACO-T in terms of the number of required vehicles, but it is significantly better than ACO-T and other optimization algorithms in terms of distance.

## 7. Conclusion

In this study, a new variant of MABC termed as the E-MABC algorithm is introduced for VRPTW. The experimental results proved that the process of interchanging neighborhood structure (two interchange, cross-exchange, and 2-opt) between two different routes in the same solution improves the local search operations of the swarm bees. This leads to an optimized outcome of minimal total travel distance and number of vehicles. For the first measure, the E-MABC algorithm outperformed MABC by 71.42%. The algorithm also achieves the best results among other optimization algorithms by reducing the total distance travelled by 41.07% compared to ACO-T, by 80.35% compared to VE-ABC, by 72.21% compared to BA and HGA. Unlike the popular development of a problem-particular algorithm, the suggested algorithm also has the benefit of being competent to resolve a wide variety of problems. To further investigate the effectiveness of the proposed E-MABC, it will be applied on VRPTW applications such as postal delivery and school bus routing.

Table 3. Comparison of E-MABC with the MABC algorithm

Problem instance	Best Known		MABC			E-MABC			
	N.V	Distance	N.V	Distance	Average	N.V	Distance	Average	Gap
R101	18	1607.70	20	1643.18	1647.91	20	1642.41	1644.66	2.16
R102	17	1434	18	1480.73	1490.66	18	1476.11	1483.30	2.94
R103	13	1175.67	14	1240.87	1258.50	14	1228.66	1240.77	4.51
R104	10	982.01	12	1047.06	1070.27	11	1035.41	1075.47	5.44
R105	15	1346.12	16	1369.52	1382.35	15	1367.3	1373.94	1.57
R106	13	1234.6	13	1271.13	1285.81	13	1264.5	1269.28	2.42
R107	11	1051.84	12	1129.99	1142.23	11	1122.15	1244.41	6.68
R108	9	960.88	11	1004.11	1026.11	11	987.88	1005.15	2.81
R109	12	1013.2	13	1170.50	1211.12	13	1168.91	1173.43	15.37
R110	12	1068	12	1123.36	1145.30	12	1117.01	1128.80	4.59
R111	12	1048.70	12	1101.59	1129.55	12	1099.26	1106.60	4.82
R112	10	935.63	11	1019.84	1026.25	11	1009.68	1024.74	7.91
R201	4	1252.37	8	1185.57	1192.87	7	1179.83	1184.35	-5.79
R202	3	1158.98	7	1103.15	1114.87	6	1092.22	1105.04	-5.76
R203	3	939.50	6	958.94	984.34	6	947.81	961.55	0.88
R204	2	825.52	4	818.44	836.49	4	814.68	821.86	-1.31
R205	3	994.42	6	1020.53	1023.79	5	999.54	1037.20	0.51
R206	3	833	5	960.29	976.45	4	959.98	974.14	15.24
R207	3	814.78	5	905.70	930.46	4	903.59	910.77	10.9
R208	2	726.75	4	764.90	789.02	3	747.15	759.98	2.81
R209	3	855	6	943.16	952.73	5	935.57	947	9.42
R210	3	939.34	6	1003.91	1015.11	5	994.29	999.35	5.85
R211	2	877.55	5	837.66	855.79	4	837.72	847.05	-4.54
C101	10	827.30	10	828.94	828.94	10	828.93	828.93	0.2
C102	10	827.30	10	828.94	828.94	10	828.93	828.93	0.2
C103	10	826.90	10	828.94	840.66	10	828.93	828.93	0.25
C104	10	822.90	10	858.90	889.10	10	843.22	849.86	2.47
C105	10	827.30	10	828.94	828.94	10	828.93	828.93	0.2
C106	10	827.30	10	828.94	828.94	10	828.93	828.93	0.2
C107	10	827.30	10	828.94	828.94	10	828.93	828.93	0.2
C108	10	827.30	10	828.94	830.85	10	828.93	828.93	0.2
C109	10	827.30	10	828.94	836.47	10	828.93	828.93	0.2
C201	3	589.10	3	591.56	591.56	3	591.56	603.43	0.42
C202	3	589.10	3	591.56	601.78	3	591.56	603.96	0.42
C203	3	591.17	3	600.54	616.39	3	600.54	608.26	1.58
C204	3	590.60	3	610.01	648.57	3	606.9	612.07	2.76
C205	3	588.88	3	588.88	596.10	3	588.88	591	0
C206	3	588.49	3	588.88	601.49	3	588.88	596.08	0.07
C207	3	588.29	3	589.58	601.60	3	589.58	597.16	0.22
C208	3	588.32	3	591.65	613.47	3	593.15	603.13	0.82
RC101	15	1619.8	16	1634.52	1668.07	16	1633.28	1654.82	0.83
RC102	13	1530.86	15	1492.89	1505.94	14	1486.85	1497.90	-2.87
RC103	11	1261.67	13	1334.57	1360.15	12	1307.59	1316.33	3.64
RC104	10	1135.48	11	1215.62	1245.35	11	1195.48	1211.94	5.28
RC105	13	1589.91	15	1546.43	1575.46	15	1539.48	1543.41	-3.17
RC106	12	1371.69	14	1423.10	1443.77	13	1413.07	1426.80	3.02
RC107	11	1222.16	12	1300.00	1324.00	12	1281.14	1303.99	4.83
RC108	10	1133.90	12	1193.68	1213.67	11	1180.05	1198.51	4.07
RC201	4	1134.91	8	1308.76	1320.24	8	1302.93	1315.82	14.8
RC202	4	1130.53	8	1167.00	1180.48	6	1165.97	1180.31	3.13
RC203	3	1026.61	6	1014.79	1032.77	5	1010.56	1061.52	-1.56
RC204	3	798.41	4	881.88	894.76	4	864.6	875.582	9.29
RC205	4	1297.19	7	1210.68	1232.84	7	1011.96	1091.90	-21.9
RC206	3	1112.20	6	1112.38	1133.99	6	1099.21	1128.11	-1.17



RC207	3	1040.67	7	1059.62	1076.47	5	1042.65	1066.97	0.19
RC208	3	828.14	5	882.06	898.45	4	878.87	885.12	6.13

Table 4. Comparison of E-MABC with optimization algorithms

Problem instance	ACO-T		VE-ABC		BA		HGA		E-MABC	
	N.V	Distance	N.V	Distance	N.V	Distance	N.V	Distance	N.V	Distance
R101	19	1655.03	22	1674.2	20	1642.67	19	1660.33	20	1642.41
R102	18	1491.18	19	1492.1	18	1480.73	18	1486.82	18	1476.11
R103	14	1243.22	15	1264.5	16	1240.87	13	1272.14	14	1228.66
R104	10	982.01	12	1042.6	11	1047.06	11	1073.63	11	1035.41
R105	16	1380.44	17	1433.6	16	1369.52	16	1380.44	15	1367.3
R106	13	1265.36	15	1291	14	1271.13	13	1283.06	13	1264.5
R107	11	1100.25	13	1130	12	1129.99	12	1117.81	11	1122.15
R108	9	958.66	11	998.7	11	1004.11	10	995.36	11	987.88
R109	12	1101.99	14	1222.4	14	1170.5	14	1101.37	13	1168.91
R110	12	1119.53	14	1142.3	13	1123.36	11	1162.19	12	1117.01
R111	12	1091.11	12	1117.4	13	1101.59	12	1093.76	12	1099.26
R112	10	974.73	11	1015.1	11	1019.84	10	1012.85	11	1009.68
R201	7	1214.22	11	1227.2	8	1185.57	6	1243.18	7	1179.83
R202	5	1105.2	8	1100.3	8	1103.15	6	1188.91	6	1092.22
R203	4	960.14	7	939.9	7	958.94	5	1050.03	6	947.81
R204	4	771.47	6	803.5	6	818.44	4	800.36	4	814.68
R205	4	1050.26	7	1034	7	1020.53	5	1056.54	5	999.54
R206	4	954.85	7	963.9	6	960.29	5	984.64	4	959.98
R207	3	870.33	5	881.1	4	905.7	3	920.26	4	903.59
R208	3	777.72	4	775.2	5	764.9	3	770.69	3	747.15
R209	3	934.21	6	932.9	6	943.16	3	902.67	5	935.57
R210	5	949.02	8	966.7	7	1003.91	5	1037.58	5	994.29
R211	4	877.55	5	825.4	5	837.99	4	887.47	4	837.72
C101	10	828.93	10	828.9	10	828.94	10	828.93	10	828.93
C102	10	828.93	10	829.9	11	828.94	10	832.67	10	828.93
C103	10	826.06	10	841.2	10	828.94	10	859.77	10	828.93
C104	10	828.2	10	869.6	10	858.9	10	893.22	10	843.22
C105	10	828.9	10	828.9	11	828.94	10	828.93	10	828.93
C106	10	828.93	10	828.9	10	828.94	10	836.65	10	828.93
C107	10	828.93	11	859.1	11	828.94	10	828.93	10	828.93
C108	10	830.93	10	846.2	10	828.94	10	861.99	10	828.93
C109	10	829.22	11	861.3	10	828.94	10	890.92	10	828.93
C201	3	591.58	4	618.6	3	591.56	3	591.55	3	591.56
C202	3	591.56	4	642.6	3	591.56	3	613.47	3	591.56
C203	3	593.25	4	618.2	3	600.54	3	646.77	3	600.54
C204	3	595.55	4	657.9	4	610.01	4	605.34	3	606.9
C205	3	588.88	3	601.2	3	588.88	4	631.21	3	588.88
C206	3	588.49	4	614.7	3	588.88	3	648.91	3	588.88
C207	3	588.88	3	607.9	3	589.58	3	609.15	3	589.58
C208	3	588.03	4	611.2	3	591.65	3	614.95	3	593.15
RC101	14	1650.14	18	1698	16	1634.52	15	1658.96	16	1633.28
RC102	13	1514.85	15	1517.1	15	1492.89	15	1514.85	14	1486.85
RC103	11	1277.11	13	1371.1	12	1334.57	13	1149.86	12	1307.59
RC104	10	1159.37	12	1211.7	11	1215.62	10	1173.47	11	1195.48
RC105	15	1617.88	15	1604.9	17	1546.43	16	1585.34	15	1539.48
RC106	13	1387.63	15	1467.6	13	1423.1	15	1403.1	13	1413.07
RC107	11	1280.01	13	1315.9	12	1300	13	1290.76	12	1281.14
RC108	11	1157.44	12	1190.9	12	1193.68	10	1157.20	11	1180.05
RC201	5	1279.65	10	1331.1	9	1308.76	5	1354.96	8	1302.93
RC202	5	1157.02	9	1168.6	8	1167	5	1257.48	6	1165.97

RC203	6	1046.33	5	1035.8	6	1014.79	5	1063.77	5	1010.56
RC204	4	847.33	4	837.6	4	881.88	4	899.34	4	864.6
RC205	5	1334.55	9	1221.6	8	1210.68	8	1236.18	7	1011.96
RC206	5	1112.2	7	1137.4	6	1112.38	5	1133.86	6	1099.21
RC207	5	1087.52	7	1043.9	6	1059.62	5	1068.73	5	1042.65
RC208	3	911.15	5	841.1	7	882.06	5	854.74	4	878.87

Table 5. Overall results based on the average performance

	Algorithm					
	MABC	ACO-T	VE-ABC	BA	HGA	E-MABC
Distance	3.68	2.71	4.35	3.70	4.44	2.13
Number of vehicles	3.68	2.25	4.98	4.31	2.91	2.87

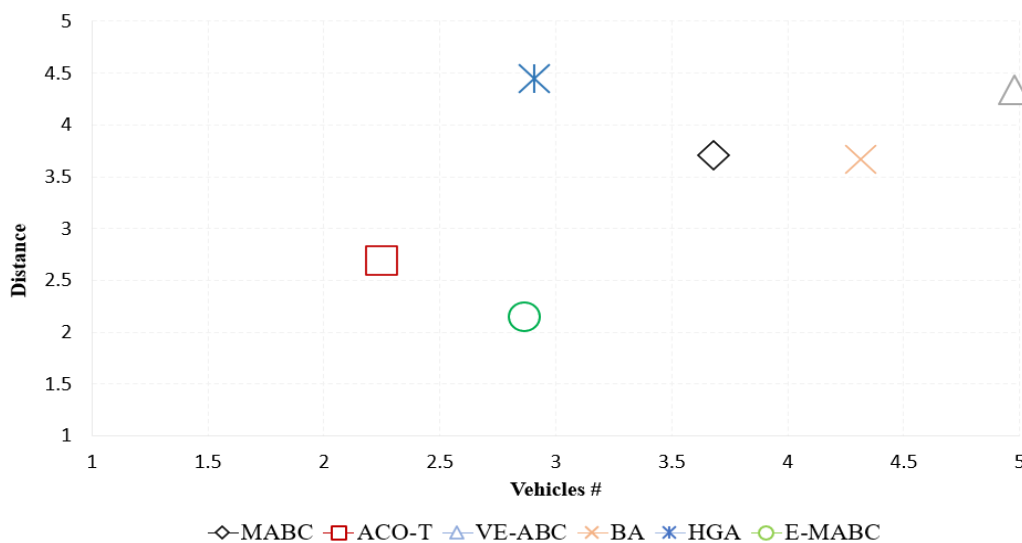


Figure 2. Average rank test of the E-MABC compared with other optimization algorithms

**Conflicts of Interest**

There are no ethical issues related to this research.

**Author Contributions**

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing-original draft preparation, writing-review and editing, visualization, have been done by 1<sup>st</sup> author. The supervision and project administration have been done by 2<sup>nd</sup> author.

**References**

[1] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem”, *Manage. Sci.*, Vol., 6, No. 1, pp. 80–91, 1959.  
 [2] A. Dixit, A. Mishra, and A. Shukla, “Vehicle Routing Problem with Time Windows Using Meta-Heuristic Algorithms: A Survey”, *Harmony Search and Nature Inspired*

*Optimization Algorithms*, Springer, Singapore, pp. 539–546, 2019.  
 [3] H. Awad and R. Elshaer, “A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants”, *Comput. Ind. Eng.*, Vol., 140, p. 106242, 2020, doi: 10.1016/j.cie.2019.106242.  
 [4] A. O. Adewumi and O. J. Adeleke, “A survey of recent advances in vehicle routing problems”, *Int. J. Syst. Assur. Eng. Manag.*, Vol., 9, No. 1, pp. 155–172, 2018, doi: 10.1007/s13198-016-0493-4.  
 [5] Y. J. Gong, J. Zhang, O. Liu, R. Z. Huang, H. S. H. Chung, and Y. H. Shi, “Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach”, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, Vol., 42, No. 2, pp. 254–267, 2012.  
 [6] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuysse, “The vehicle routing problem: State of the art classification and review”, *Comput. Ind. Eng.*, Vol., 99, pp. 300–313, 2016, doi: 10.1016/j.cie.2015.12.007.

- [7] M. Blocho, "Exact algorithms for solving rich vehicle routing problems", *Smart Delivery Systems*, Elsevier, pp. 93–99, 2020.
- [8] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, Part I: Route construction and local search algorithms", *Transp. Sci.*, Vol., 39, No. 1, pp. 104–118, 2005, doi: 10.1287/trsc.1030.0056.
- [9] B. Arbaoui, J. Wahid, and S. Abdul-Rahman, "Enhancing the Sorting Layers in the Initial Stage of High School Timetabling", In: *Proc. of ISCAIE 2020 - IEEE 10th Symp. Comput. Appl. Ind. Electron.*, pp. 59–63, 2020, doi: 10.1109/ISCAIE47305.2020.9108804.
- [10] I. Stojanović, I. Brajević, P. S. Stanimirović, L. A. Kazakovtsev, and Z. Zdravev, "Application of Heuristic and Metaheuristic Algorithms in Solving Constrained Weber Problem with Feasible Region Bounded by Arcs", *Math. Probl. Eng.*, 2017.
- [11] O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, Part II: Metaheuristics", *Transp. Sci.*, Vol., 39, No. 1, pp. 119–139, 2005, doi: 10.1287/trsc.1030.0057.
- [12] A. S. Hameed, B. M. Aboobaidar, M. L. Mutar, and N. H. Choon, "A new hybrid approach based on discrete differential evolution algorithm to enhancement solutions of quadratic assignment problem", *Int. J. Ind. Eng. Comput.*, Vol., 11, No. 1, pp. 51–72, 2020, doi: 10.5267/j.ijiec.2019.6.005.
- [13] S. A. Yasear and K. R. Ku-Mahamud, "Taxonomy of Memory Usage in Swarm Intelligence-Based Metaheuristics", *Baghdad Sci. J.*, Vol., 16, No. 2(SI), pp. 445–452, 2019, doi: 10.21123/bsj.2019.16.2(si).0445.
- [14] J. Wahid and H. F. A. Al-Mazini, "Classification of Cervical Cancer Using Ant-Miner for Medical Expertise Knowledge Management", In: *Proc. of Knowl. Manag. Int. Conf.*, No. November, pp. 25–27, 2018.
- [15] S. Banerjee and N. Agarwal, "Analyzing collective behavior from blogs using swarm intelligence", *Knowl. Inf. Syst.*, Vol., 33, No. 3, pp. 523–547, 2012, doi: 10.1007/s10115-012-0512-y.
- [16] E. Cuevas, F. Fausto, and A. González, "A Swarm Algorithm Inspired by the Collective Animal Behavior", *New Advancements in Swarm Algorithms: Operators and Applications*, Springer, Cham, pp. 161–188, 2020.
- [17] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, "HYBRID ANT COLONY OPTIMIZATION AND ITERATED LOCAL SEARCH FOR RULES-BASED CLASSIFICATION", *J. Theor. Appl. Inf. Technol.*, Vol., 98, No. 04, pp. 657–671, 2020.
- [18] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Modified ACS centroid memory for data clustering", *J. Comput. Sci.*, Vol., 15, No. 10, pp. 1439–1449, 2019, doi: 10.3844/jcssp.2019.1439.1449.
- [19] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey", *Artif. Intell. Rev.*, Vol., 52, No. 4, pp. 2191–2233, 2019, doi: 10.1007/s10462-017-9605-z.
- [20] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", *Tech. report-tr06*, Erciyes Univ. Eng. Fac. Comput. Eng. Dep., Vol., 200, pp. 1–10, 2005, doi: 10.1007/978-1-4614-7990-1.
- [21] W. Y. Szeto, Y. Wu, and S. C. Ho, "An artificial bee colony algorithm for the capacitated vehicle routing problem", *Eur. J. Oper. Res.*, Vol., 215, No. 1, pp. 126–135, 2011.
- [22] A. Alvarado-Iniesta, J. L. Garcia-Alcaraz, M. I. Rodriguez-Borbon, and A. Maldonado, "Optimization of the material flow in a manufacturing plant by use of artificial bee colony algorithm", *Expert Syst. Appl.*, Vol., 40, No. 12, pp. 4785–4790, 2013.
- [23] B. Yu, Z. Z. Yang, and B. Z. Yao, "A hybrid algorithm for vehicle routing problem with time windows", *Expert Syst. Appl.*, Vol., 38, No. 1, pp. 435–441, 2011, doi: 10.1016/j.eswa.2010.06.082.
- [24] Z. Ursani, D. Essam, D. Cornforth, and R. Stocker, "Localized genetic algorithm for vehicle routing problem with time windows", *Appl. Soft Comput. J.*, Vol., 11, No. 8, pp. 5375–5390, 2011, doi: 10.1016/j.asoc.2011.05.021.
- [25] Y. J. Shi, F. W. Meng, and G. Jiang Shen, "A modified artificial bee colony algorithm for vehicle routing problems with time windows", *Inf. Technol. J.*, Vol., 11, No. 10, pp. 1490–1495, 2012.
- [26] O. E. Nahum, Y. Hadas, and U. Spiegel, "Multi-Objective Vehicle Routing Problems with Time Windows: A Vector Evaluated Artificial Bee Colony Approach", *Int. J. Comput. Inf. Technol.*, Vol., 3, No. 1, pp. 41–47, 2014.
- [27] S. Yu, C. Tai, Y. Liu, and L. Gao, "An improved artificial bee colony algorithm for vehicle routing problem with time windows: A real case in Dalian", *Adv. Mech. Eng.*, Vol., 8, No. 8, pp. 1–9, 2016, doi: 10.1177/1687814016665298.
- [28] B. Yao, Q. Yan, M. Zhang, and Y. Yang, "Improved artificial bee colony algorithm for vehicle routing problem with time windows",

- PLoS One*, Vol., 12, No. 9, pp. 1–18, 2017, doi: 10.1371/journal.pone.0181275.
- [29] C. Chen and K. Zhou, “Application of Artificial Bee Colony Algorithm in Vehicle Routing Problem with Time Windows”, in *In 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control*, pp. 781–785, 2018, doi: 10.1109/SDPC.2018.8664999.
- [30] M. Alzaqebah, S. Abdullah, and S. Jawarneh, “Modified artificial bee colony for the vehicle routing problems with time windows”, *Springerplus*, Vol., 5, No. 1, 2016, doi: 10.1186/s40064-016-2940-8.
- [31] M. M. Solomon, “Algorithms for the Vehicle Routing and Scheduling Problems With Time Window Constraints”, *Oper. Res.*, Vol., 35, No. 2, pp. 254–265, 1987, doi: 10.1287/opre.35.2.254.
- [32] M. Alzaqebah, S. Jawarneh, H. M. Sarim, and S. Abdullah, “Bees algorithm for vehicle routing problems with time windows”, *Int. J. Mach. Learn. Comput.*, Vol., 8, No. 3, pp. 236–240, 2018, doi: 10.18178/ijmlc.2018.8.3.693.
- [33] K. C. Tan, L. H. Lee, and K. Ou, “Artificial intelligence heuristics in solving vehicle routing problems with time window constraints”, *Eng. Appl. Artif. Intell.*, Vol., 14, No. 6, pp. 825–837, 2001, doi: 10.1016/S0952-1976(02)00011-8.