ICIC Express Letters
Part B: Applications
Volume **13**, Number **6**, June **2022**

ICIC International ©2022 ISSN 2185-2766
pp. 571–579

MODIFIED ANT COLONY OPTIMIZATION FOR VEHICLE ROUTING PROBLEM WITH TIME WINDOWS USING LIMITED SEARCH SPACE AND NOVEL UPDATING PHEROMONE AND RE-INITIALIZATION PHEROMONE TECHNIQUES

Chiabwoot Ratanavilisagul

Department of Computer and Information Science of Applied Science
King Mongkut's University of Technology North Bangkok (KMUTNB)
1518 Pracharat 1 Road, Wongsawang, Bangsue, Bangkok 10800, Thailand
chaibwoot.r@sci.kmutnb.ac.th; chaibwoot@hotmail.com; chaibwoot@gmail.com

Received September 2021; accepted December 2021

Abstract. *The goal of vehicle routing problem with time windows (VRPTW) is to find the minimum cost routes to deliver the customers within due time, considering that each vehicle has a limited capacity. This paper proposed ant colony optimization (ACO) to solve VRPTW problems. However, ACO often experiences trapping in local optimum. This article proposed an improved ACO by using the limited search space technique, the novel updating pheromone technique, and the re-initialization pheromone technique. Using these techniques simultaneously has proven to solve VRPTW problems and increased searching performance of the ACO. The proposed technique was tested on fifty-six maps from a series of Solomon and provided more satisfactory results in comparison with other ACO techniques.*
Keywords: Vehicle routing problem with time windows, Ant colony optimization, Optimization, Re-initialization, Local search, Swarm intelligence

1. **Introduction.** The vehicle routing problem with time windows (VRPTW) is a combinatorial optimization problem and NP-hard problem [1,2], as well as a popular problem for comparing the performance of algorithms. VRPTW is caused by logistical problems, scheduling issues, and supply chain management issues. In the recent years, many researchers have proposed using heuristic to solve VRPTW or problems are similar to it because the results are highly accurate in effectively solving large and complex problems such as simulated annealing (SA) [3], combined scheduling model [4], neighborhood search [5], genetic algorithms (GA) [6,7], TABU search [8,9], particle swarm optimization (PSO) [10], and ACO [11-15]. ACO has successfully solved vehicle routing problems (VRP) [16,17]. Hence, this paper focuses only on the ACO that solves VRPTW problems. ACO is motivated by the foraging behavior of an ant colony [18-20]. This approach has been widely used for solving optimization problems [21] such as job scheduling problems, traveling salesman problems (TSP), network routing and VRP. The advantages of ACO [21-23] are rapid discovery of good solutions and good efficiency for TSP or similar problems. However, the disadvantage of ACO [21-23] is that it has a propensity for trapping in the local optimum. To overcome the disadvantages of ACO, many researchers [14,15,24,25] have increased its searching diversity by adding the mutation technique and reset technique to the ACO process. In addition, many researchers [13-15,25] have added the neighborhood search technique or local search technique to the ACO process to improve the resulting solutions. To manage trapping in the local optimum, this paper proposes using the limited search space technique, the novel updating pheromone technique, and the re-initialization pheromone technique applied to ACO. A set of the Solomon fifty-six

DOI: 10.24507/icicelb.13.06.571

maps [26,27] was used to compare with other ACOs, and the proposed algorithm. The best paths of the Solomon fifty-six maps were presented by [28]. The results show that the quality and reliability of the solutions using the proposed technique were better than other comparative algorithms. VRPTW, ACO, the neighborhood search, and the previous modified ACO with VRPTW are provided in Section 2. Section 3 explains the proposed algorithm. Section 4 explains the experiment results, and Section 5 concludes the paper with a brief summary.

2. **Background.**

2.1. **VRPTW.** The goal of VRPTW is to find a set of minimum cost routes to transport goods to all customers for a specified period of time without exceeding the capacity of the vehicle [29,30]. All vehicle routes start and end at the supplies depot. Each customer is visited only once. The formulation of VRPTW is described as follows:

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

$$Min \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} C_{ij} x_{ijk} \tag{2}$$

$$\sum_{k=1}^{K} \sum_{j=1}^{N} x_{ijk} \leq K \text{ for } i = 0 \tag{3}$$

$$\sum_{i=1}^{N} x_{ijk} = \sum_{j=1}^{N} x_{ijk} \leq 1 \text{ for } i = 0,\ k \in \{1, \ldots, K\} \tag{4}$$

$$\sum_{j=0, j \neq i}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \text{ for } i \in \{1, \ldots, N\} \tag{5}$$

$$\sum_{i=0, j \neq i}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \text{ for } j \in \{1, \ldots, N\} \tag{6}$$

$$\sum_{i=1}^{N} \sum_{j=0, j \neq i}^{N} x_{ijk} \times q_i \leq Q \text{ for } k \in \{1, \ldots, K\} \tag{7}$$

$$t_0 = w_0 = 0 \tag{8}$$

$$\sum_{i=1}^{N} \sum_{j=0, j \neq i}^{N} x_{ijk} = x_{ijk}(t_i + t_{ij} + w_i) \leq t_j \text{ for } k \in \{1, \ldots, K\} \tag{9}$$

$$T_{Ei} \leq t_i \leq T_{Li} \tag{10}$$

where q_i is the quantity of goods shipped to customer i for the depot is 0. C_{ij} is the distance from i to j customers, which is computed using the Euclidian distance between i and j customers. Q is the capacity of a vehicle. t_i is the arrival time for i customer. t_{ij} is the travel time between i and j customers. w_i is the wait time at i customer. K is the total number of vehicles. N is the depot and set of customers. T_{Ei} is the earliest arrival time at i customer. If the vehicle visits before T_{Ei}, t_i is set T_{Ei}. T_{Li} is the latest arrival time at i customer. The vehicle cannot arrive at i customer after T_{Li}. The goal of VRPTW to minimize the total travelling cost (C_{ij}) is given by (2). Maximum number of routes constraint is given by (3). A customer being visited only once is given by (4) and (5). All vehicle tours starting and ending at the depot are given by (6). The total quantity of goods that a vehicle carries cannot exceed the Q capacity is given by (7).

The time constraint is given by (8) and (9). The vehicle serving the customer within the proper time interval is given by (10).

2.2. **The neighborhood search.** The neighborhood search is conducted by changing one or two customer nodes in routes to improve solutions [31]. This paper uses three well-known techniques as follows: customer exchange [13], one move operator [13], and two-opt [32].

2.3. **Ant colony optimization.** ACO can be described as follows. Initially, each edge has an initial pheromone $\tau_{ij}(0)$ between two cities. The next step is to select a customer of ant. The first customer of each ant is randomly selected, and then each ant selects the next customer according to the probability function as follows:

$$P_{ij}^k = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [n_{ij}]^\beta}{\sum\limits_{t \in \text{allowed}} [\tau_{ij}(t)]^\alpha [n_{ij}]^\beta}, & j \in \text{allowed} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where P_{ij}^k is the probability of ant k choosing to move from customer i to customer j, τ_{ij} is the pheromone, $n_{ij} = \frac{1}{d_{ij}}$ is the inverse of the distance, β is a parameter which determines the relative importance of pheromone versus distance ($\beta > 0$). The result from Formula (11) is selection of a path that is shorter and has a greater amount of pheromone. After the ant city selection process is completed, the fitness of the ant is calculated by (2). The fitness of each ant is used to update pheromones according to (12). The next step is to repeat this until the stop condition is reached.

$$\Delta\tau_{ij}(t+n) = \rho\tau_{ij} + \Delta\tau_{ij} \tag{12}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \tau_{ij}^k \tag{13}$$

where $\Delta\tau_{ij}$ is total amount of pheromones that have been left by ants using the route between customers i and j, τ_{ij}^k is the amount of pheromones of the k ant at its edge between t and $t+n$, ρ is a coefficient and $(1-\rho)$ represents the evaporation of the trail between time t and $t+n$. Q is a consistent value, and L_k is the fitness of ant k. $\Delta\tau_{ij}^k$ is calculated by (14).

$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{L_k}, & \text{if the } k\text{th ant uses edge } (i,j) \text{ in its tour} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

In the VRPTW, each ant starts at the depot and creates a route by selecting a customer under the capacity constraints and the time window constraints until the capacity of the vehicle cannot be further loaded. It is then returned to the depot. Customer selection is based on pheromone [25]. The process is repeated until all customers are selected.

2.4. **The previous modified ACO with VRPTW.** To address the trapping local optimum problem, many researchers proposed the following techniques to improve ACO: the first technique is ant colony optimization and TABU to solve VRPTW (ACOT) [13]. TABU search is used to maintain ACO diversity and explore new solutions. The second technique is ant colony optimization and pheromone reset to solve VRPTW (ACOR) [14]. The result from pheromone reset can decrease the trapping in local optimum problem. The third technique is to optimize ant colony and mutation operator to solve VRPTW problems (ACOM) [15]. This technique added three mutation operators to the ACO process and a new technique for updating pheromone. Three algorithms have been tested with Solomon's problems. The results showed that the solutions of these algorithms were better than those of the competitive algorithms.

3. **Proposed Work.**

3.1. **ACO with limited search space.** The three previously discussed techniques (ACOT, ACOR, and ACOM) still encounter trapping problems in local optimum. The experiment results show that these techniques cannot find many of the best paths. The cause of the trapping problem in local optimum may be that the search space is larger than what the algorithm can solve. Therefore, decreasing search space size may solve the trapping problem in local optimum. Reducing the search space can be applied to the ACO process for creating solutions. As part of the solution creating solution process of ACO, a customer is selected from all customers. For example, if there are one hundred customers, the ACO must select one of the hundred making the possibility of selecting the most suitable customers to be minimal.

The criteria of selecting the suitable customers are determined by the greedy algorithm as follows: the nearest distance from the current customer to the selected customer, the minimum time of the earliest arrival time (TE) from the current customer to the selected customer, the least time of the latest arrival time (TL) from the current customer to the selected customer, the least goods quantity of the selected customer. Moreover, the suitable customer selection criteria change with the ACO cycle so that pheromone is updated from multiple criteria because each map obtains the good solution from different criteria.

Under the normal ACO solution creation process, a customer is added to a vehicle until no further customers can be added. A new vehicle is created if unselected customers remain. The technique is called sequential addition. If a customer is added by the greedy algorithm, the first vehicles will get a good solution but the latter vehicles get a very bad solution. A good selection should be a good solution for all vehicles.

Hence, this paper proposes a parallel addition. All customers are to be added to all vehicles alternately. For example, a scenario is considered in which the number of vehicles is two and number of all customers is 8 (0 to 7). First addition, the first vehicle is added and supposes its path is 0, 1. Next, the second vehicle is added and supposes its path is 0, 3. Next, the first vehicle is added and supposes its path is 0, 1, 6. Next, the second vehicle is added and supposes its path is 0, 3, 2. The addition is swapped as mentioned earlier until all customers are selected. However, this technique requires evaluating the number of vehicles (NV) that are used. NV is calculated by (15). If vehicles are not enough, a new vehicle will be created and uses the sequential addition.

$$NV = \frac{\sum_{i=1}^{N} Q_i}{Q} \tag{15}$$

As mentioned earlier, this paper proposes to use reducing the search space from a variety criteria and adding parallel to the ACO to decrease or solve trapping in the local optimum. The proposed technique is called ACO with limited search space.

3.2. **ACO with novel updating pheromone.** As previously mentioned, neighborhood search can improve the solution from ACO but it cannot optimize ACO searches. If pheromone is updated from the good solution, it can create better solutions [25]. Hence, this paper then proposes to select the best ant that gets the best solution in that cycle. The best ant which is selected is called the leader. Then the neighborhood search process is applied to the leader to improve its solution. Moreover, the best solution found in the whole search (GBEST) should be used to update the pheromone combined with its leader. The proposed technique is called ACO with updating pheromone by leader and GBEST or ACO with novel updating pheromone.

3.3. **ACO with re-initialization pheromone.** As previously mentioned, limited search space technique can decrease trapping in local optimum problems. However, trapping in the local optimum is possible for the length of the search time. Under the normal process of ACO, the pheromone is a factor in creating a solution. When pheromone is stuck trapping in local optimum, the pheromone of ACO creates the same solution as the previous solution or close to the previous solution [25]. The simple technique to indicate trapping in local optimum state of ACO is to monitor the unchanged number of consecutive best solution [33]. For these reasons, this paper therefore proposed that all pheromones be reset when the unchanged number of the best consecutive solution is more than the threshold value. This proposed technique is called ACO with re-initialization pheromone.

3.4. **Modified ACO using limited search space, novel updating pheromone, and re-initialization pheromone.** As previously mentioned, the three proposed techniques are applied together with the ACO to increasing searching performance of ACO and decreasing or solving trapping in local optimum problems. This proposed technique is called modified ACO using limited search space and novel updating pheromone, and re-initialization pheromone (ACOLPR). Pseudo code of ACOLPR is shown below:

```
Initialize edges (i, j) and pheromone
Calculate NV by Equation (15)
Iteration = 0, reset GR, TG = 0
While termination condition ≠ true do
   Reset the fitness of leader
   MV = NV
   For ant x begin 1 to a number of all ants
      For until customers in list of ant x is full
         For car k begin 1 to MV
            If car k can add the next customer
               The chosen customer is not repeat customer in list of ant x
               The chosen customer passes condition of capacity and time windows
               If Iteration % 4 = 0
                  The chosen customer is according to the goods quantity criterion (sort
                  min to max)
               Else If Iteration % 4 = 1
                  The chosen customer is according to the earliest arrival time criterion
                  (sort min to max)
               Else If Iteration % 4 = 2
                  The chosen customer is according to the latest arrival time criterion
                  (sort min to max)
               Else If Iteration % 4 = 3
                  The chosen customer is according to the distance criterion (sort min
                  to max)
               End If
               A number of chosen customers in the top lists are not over SN
               Only customers in list are considered to choose the next customer with
               probability according to Formula (11)
               Insert chosen customer into list of ant x
            End If
            If no one vehicle cannot add the next customer and some customers are
            not stay in list ant x
               Add the new vehicle and MV = MV + 1
            End If
         End For
      End For
   End For
```

```
    Evaluate the fitness of each ant
    If the fitness of an ant is better than that of leader
      leader = this ant
    End If
    Apply customer exchange algorithm with leader
    Apply one move operator algorithm with leader
    Apply 2-opt algorithm with leader
    If the fitness of leader is better than that of GBEST
      GBEST = leader
    End If
    If the fitness of leader is better than that of GR
      GR = leader
    End If
    Update pheromone by GBEST and leader according to Formula (13)
    If GR is not improve
      TG++
    End If
    If TG >= RP
      All pheromones are initialized
      Reset GR, TG = 0
    End If
    Apply the evaporation
    Iteration++
  End While
```

Here GR is the best solution to each reset cycle. TG is the times of GR consecutive unchanged. SN is the most suitable number of customers. RP is the re-initialize period.

4. **Results and Discussions.** The parameters are as follows for all experiments: $\beta = 4$, $Q = 0.6$, $\alpha = 1$, $\rho = 0.7$, $\tau_0 = (n \times L_{nn})^{-1}$ where L_{nn} is the length of the list of ants that is produced by the nearest neighbor heuristic [34]. The number of ants used is 50. The number of experiments of each map is 10 runs. The maximum number of iterations is set as 100000. For proposal algorithm, RP = 100 and SN = 30. Non-ACO parameters for compared algorithm parameters (ACOT, ACOR, and ACOM) are set according to the recommended source document. This research is conducted by AMD FX-8320 personal computer with 16 GB RAM and Visual C++ 2005 as the programming language. All data sets used in the experiment are from Solomon's VRPTW maps [26]. The measures of algorithm performance in the experiments are as follows: the average best fitness value (ABF) is the average of best fitness in the final iteration from all runs. ABF indicates the solution searching efficiency of an algorithm. SD is the standard deviation. SD indicates the solution searching reliability of an algorithm.

From the experimental results in Table 1, ACOLPR can locate the optimum points of all tested maps. The quality of ACOLPR's solution is better than that of ACOT, ACOR, and ACOM because of its lowest ABF value of all tested maps. The reliability of ACOLPR is better than that of ACOT, ACOR, and ACOM because of its lowest SD in all tested maps. ACOLPR can solve trapping in the local optimum, which is better than ACOT, ACOR, and ACOM. So, ACOLPR can get better results than ACOT, ACOR, and ACOM.

5. **Conclusion.** This research paper has proposed ACOLPR derived from the concurrent application of the limited search space, the novel updating pheromone, and the re-initialization pheromone techniques to solve optimization problems and enhance searching

Table 1. Comparative ABF and SD of ACOT, ACOR, ACOM and ACOLPR

Problem	**Best known**	**ACOT**		**ACOR**		**ACOM**		**ACOLPR**	
		ABF	**SD**	**ABF**	**SD**	**ABF**	**SD**	**ABF**	**SD**
C101	828.93	883.63	21.23	851.41	30.45	834.99	69.60	828.93	0
C102	828.93	902.93	19.96	860.81	20.85	846.18	43.48	828.93	0
C103	828.06	845.71	15.41	846.94	31.88	830.09	10.01	828.06	0
C104	824.78	854.14	30.69	840.23	9.83	836.77	12.55	824.78	0
C105	828.94	892.08	21.49	869.34	15.09	860.63	16.28	828.94	0
C106	828.94	908.41	16.39	852.50	24.73	830.69	16.81	828.94	0
C107	828.94	896.28	26.97	878.20	11.90	849.57	26.47	828.94	0
C108	828.94	864.68	26.86	862.06	24.15	842.85	12.40	828.94	0
C109	828.94	874.63	30.01	863.99	9.90	833.37	14.36	828.94	0
C201	591.56	657.65	30.70	654.39	48.46	626.24	53.85	591.56	0
C202	591.56	647.60	22.84	628.11	25.11	627.96	29.22	591.56	0
C203	591.17	618.52	46.69	610.41	13.82	601.62	30.69	591.17	0
C204	590.60	671.40	31.54	646.56	24.01	595.15	22.21	590.60	0
C205	588.88	637.11	22.90	618.51	22.83	593.90	4.87	588.88	0
C206	588.49	615.11	10.73	610.00	7.51	602.38	9.73	588.49	0
C207	588.29	616.46	12.30	613.40	30.15	595.78	10.50	588.29	0
C208	588.32	623.64	5.95	612.61	5.32	596.49	5.85	588.32	0
R101	1650.80	1736.72	24.35	1694.49	13.17	1671.36	59.30	1650.80	0
R102	1486.86	1536.67	61.11	1535.31	12.45	1509.85	11.34	1486.86	0
R103	1292.67	1335.83	18.66	1296.03	45.47	1292.67	0.00	1292.67	0
R104	1007.31	1041.70	15.46	1025.41	10.98	1016.75	35.83	1007.31	0
R105	1377.11	1460.61	15.62	1442.80	10.70	1434.52	67.92	1377.11	0
R106	1252.03	1321.22	15.51	1315.86	58.30	1309.90	8.81	1252.03	0
R107	1104.66	1156.59	16.68	1140.21	13.47	1112.06	72.80	1104.66	0
R108	960.88	1006.37	26.21	999.29	9.91	978.37	12.96	960.88	0
R109	1194.73	1215.04	14.48	1211.01	12.57	1205.88	57.45	1194.73	0
R110	1118.84	1160.63	7.19	1129.90	8.57	1125.32	14.01	1118.84	0
R111	1096.73	1146.05	47.26	1130.02	11.94	1116.59	14.27	1096.73	0
R112	982.14	982.14	0.00	985.65	7.22	982.14	0.00	982.14	0
R201	1252.37	1412.74	14.62	1339.50	86.18	1326.67	15.70	1252.37	0
R202	1191.70	1291.06	13.44	1258.93	47.42	1221.74	18.61	1191.70	0
R203	939.50	1001.41	38.31	964.04	68.04	959.44	71.10	939.50	0
R204	825.52	872.82	10.05	869.47	6.83	851.28	28.94	825.52	0
R205	994.43	1218.43	18.07	1110.48	14.68	1081.92	68.11	994.43	0
R206	906.14	1032.73	10.32	1017.40	20.43	914.29	59.71	906.14	0
R207	890.61	979.61	15.53	931.16	45.86	928.65	17.94	890.61	0
R208	726.82	753.93	14.40	749.83	36.46	746.10	14.10	726.82	0
R209	909.16	990.49	10.39	980.13	10.22	943.02	61.97	909.16	0
R210	939.37	1058.68	11.97	999.85	100.13	971.10	19.19	939.37	0
R211	885.71	885.71	0.00	885.71	0.00	885.71	0.00	885.71	0
RC101	1696.95	1793.54	30.28	1742.37	104.89	1715.77	26.03	1696.95	0
RC102	1554.75	1584.28	8.22	1573.42	64.44	1572.62	17.21	1554.75	0
RC103	1261.67	1303.17	81.94	1300.90	10.01	1264.73	13.20	1261.67	0
RC104	1135.48	1158.41	16.71	1144.58	9.45	1136.40	48.89	1135.48	0
RC105	1629.44	1671.30	54.32	1655.75	13.22	1637.61	25.95	1629.44	0
RC106	1424.73	1474.09	24.95	1435.61	13.20	1426.34	54.09	1424.73	0
RC107	1230.48	1251.12	12.87	1237.55	52.08	1235.43	5.93	1230.48	0
RC108	1139.82	1140.88	3.35	1139.82	0.00	1139.82	0.00	1139.82	0
RC201	1406.94	1834.86	33.65	1664.69	56.95	1592.13	17.41	1406.94	0
RC202	1365.64	1443.15	24.55	1387.86	100.28	1392.71	17.84	1365.64	0
RC203	1049.62	1125.85	83.07	1113.84	13.81	1057.92	15.24	1049.62	0
RC204	798.46	828.63	15.91	811.25	6.16	804.23	24.72	798.46	0
RC205	1297.65	1528.47	26.66	1358.75	113.70	1334.22	22.47	1297.65	0
RC206	1146.32	1541.73	38.77	1309.39	10.23	1233.97	94.27	1146.32	0
RC207	1061.14	1239.78	17.04	1083.63	24.31	1065.38	93.55	1061.14	0
RC208	828.14	828.14	0.00	828.14	0.00	828.14	0.00	828.14	0

performance. For the Solomon's VRPTW maps, the ACOT, ACOR, ACOM, and ACOLPR were tested and results were compared. The results indicated that the proposed ACOLPR outperforms ACOT, ACOR, and ACOM with regard to the reliability and quality of the solutions in all experiments. However, the proposed algorithm requires several iterations to solve trapping in the local optimum. In the future, it would be of interest to improve the proposed algorithm by reducing the number of iterations required that can still solve trapping in the local optimum.

REFERENCES

[1] S. R. Balseiro, I. Loiseau and J. Ramonet, An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows, *Comput. Oper. Res.*, pp.954-966, 2011.
[2] K. Braekers, K. Ramaekers and I. V. Nieuwenhuyse, The vehicle routing problem: State of the art classification and review, *Comput. Ind. Eng.*, pp.300-313, 2016.
[3] M. H. Hassan, L. Makdyssiian and W. H. Bila, An integration between the improved ant algorithm and the simulated annealing algorithm to contribute to solve the vehicle routing problem with time windows, *International Journal of Novel Research in Physics Chemistry & Mathematics*, vol.3, pp.64-74, 2016.
[4] Q. Xu, Z. Li, L. Shen, Z. Bian, L. Xing and Z. Jin, Optimization on combined scheduling of tractor and trailer routing problem considering synchronized operations, *International Journal of Innovative Computing, Information and Control*, vol.15, no.3, pp.983-995, 2019.
[5] S. Belhaiza, R. M'hallah and G. B. Brahim, A new hybrid genetic variable neighborhood search heuristic for the vehicle routing problem with multiple time windows, *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp.1319-1326, 2017.
[6] Z. Guo, Y. Li, X. Jiang and S. Gao, The electric vehicle routing problem with time windows using genetic algorithm, *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp.635-639, 2017.
[7] K. A. Putri, N. L. Rachmawati, M. Lusiani and A. A. Ngurah Perwira Redi, Genetic algorithm with cluster-first route-second to solve the capacitated vehicle routing problem with time windows: A case study, *Jurnal Teknik Industri*, vol.23, no.1, pp.75-82, 2021.
[8] G. Li and J. Li, An improved TABU search algorithm for the stochastic vehicle routing problem with soft time windows, *IEEE Access*, vol.8, pp.158115-158124, 2020.
[9] L. Li, Y. Chen and J. Meng, Improved TABU search algorithm for solving the vehicle routing problem with soft time windows in B2C environment, *Chinese Control and Decision Conference*, pp.3629-3633, 2020.
[10] J. Zhang, F. Yang and X. Weng, An evolutionary scatter search particle swarm optimization algorithm for the vehicle routing problem with time windows, *IEEE Access*, vol.6, pp.63468-63485, 2018.
[11] H. Wu, Y. Gao, W. Wang and Z. Zhang, A hybrid ant colony algorithm based on multiple strategies for the vehicle routing problem with time windows, *Complex & Intelligent Systems*, 2021.
[12] A. Palma-Blanco, E. R. González Ponzón and C. D. Paternina-Arboleda, A two-pheromone trail ant colony system approach for the heterogeneous vehicle routing problem with time windows, multiple products and product incompatibility, *International Conference on Computational Logistics ICCL 2019 Computational Logistics*, pp.248-264, 2019.
[13] B. Yu, Z. Z. Yang and B. Z. Yao, A hybrid algorithm for vehicle routing problem with time windows, *Expert Systems with Applications*, vol.38, no.1, pp.435-441, 2011.
[14] A. Gupta and S. Saini, An enhanced ant colony optimization algorithm for vehicle routing problem with time windows, *2017 9th International Conference on Advanced Computing (ICOAC)*, pp.267-274, 2017.
[15] H. Zhang, Q. Zhang, L. Ma, Z. Zhang and Y. Liu, A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows, *Information Sciences*, vol.490, pp.166-190, 2019.
[16] B. Bullnheimer, R. F. Hartl and C. Strauss, An improved ant system algorithm for the vehicle routing problem, *Annals of Operations Research*, pp.319-328, 1999.
[17] B. Yu, Z. Z. Yang and B. Z. Yao, An improved ant colony optimization for vehicle routing problem, *European Journal of Operational Research*, vol.196, no.1, pp.171-176, 2009.
[18] A. Colorni, M. Dorigo and V. Maniezzo, Distributed optimization by ant colonies, *Proc. of European Conference on Artificial Life*, Paris, France, pp.134-142, 1991.

[19] M. Dorigo, V. Maniezzo and A. Colorni, Ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.26, pp.1-13, 1996.

[20] M. Dorigo and L. M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, vol.1, pp.53-66, 1997.

[21] V. Selvi and R. Umarani, Comparative analysis of ant colony and particle swarm optimization techniques, *International Journal of Computer Applications*, vol.5, pp.1-6, 2010.

[22] M. Dorigo and L. M. Gambardella, A study of some properties of ant-Q, *Proc. of the 44th International Conference on Parallel Problem Solving from Nature*, pp.656-665, 1996.

[23] A. Colorni, M. Dorigo et al., Heuristics from nature for hard combinatorial optimization problems, *International Transactions in Operational Research*, vol.3, no.1, pp.1-21, 1996.

[24] C. Ratanavilisagul, Modified ant colony optimization with pheromone mutation for travelling salesman problem, *The 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2017.

[25] C. Ratanavilisagul, Modified ant colony optimization with updating pheromone by leader and reinitialization pheromone for travelling salesman problem, *International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, 2018.

[26] M. M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, pp.254-265, 1987.

[27] https://people.idsia.ch/~luca/macs-vrptw/problems/welcome.htm, Accessed on April 18, 2022.

[28] http://sun.aei.polsl.pl/~zjc/best-solutions-solomon.html#c201, Accessed on April 18, 2022.

[29] E. Teymourian, V. Kayvanfar, G. M. Komaki and M. Zandieh, Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem, *Information Sciences*, vols.334-335, pp.354-378, 2016.

[30] T. Vidal, Technical note: Split algorithm in O(n) for the capacitated vehicle routing problem, *Comput. Oper. Res.*, pp.40-47, 2016.

[31] B. Bullnheimer, R. F. Hartl and C. Strauss, An improved ant system algorithm for the vehicle routing problem, *Annals of Operations Research*, pp.319-328, 1999.

[32] G. A. Croes, A method for solving traveling salesman problems, *Operations Research*, vol.6, no.6, pp.791-812, 1958.

[33] C. Ratanavilisagul and B. Kruatrachue, A modified particle swarm optimization with mutation and reposition, *International Journal of Innovative Computing, Information and Control*, vol.10, no.6, pp.2127-2142, 2014.

[34] D. J. Rosenkrantz, R. E. Stearns and P. M. Lewis, An analysis of several heuristics for the traveling salesman problem, *Siam J. Comput.*, vol.6, pp.563-581, 1977.