2021 the 7th International Conference on Computer and Communications

A Novel Modified Ant Colony Optimization Algorithm by Resetting and Updating Pheromone for Vehicle Routing Problem with Time Windows

C. Ratanavilisagul, "A novel modified Ant Colony Optimization Algorithm by resetting and updating pheromone for Vehicle Routing Problem with Time Windows," in Proc. 7th Int. Conf. Comput. Commun. (ICCC), Chengdu, China, 2021, pp. 1194–1198, doi: 10.1109/ICCC54389.2021.9674681.

Chiabwoot Ratanavilisagul
Department of Computer and Information Science of Applied Science
King Mongkut's University of Technology North Bangkok (KMUTNB)
Bangkok, Thailand
chaibwoot@hotmail.com, chaibwoot@gmail.com, chaibwoot.r@sci.kmutnb.ac.th

***Abstract*—Ant Colony Optimization (ACO) algorithm is used for solving combinational optimization problem. Vehicle routing problem with time windows (VRPTW) is a well-known combinatorial problem. Many researchers apply ACO to solve VRPTW. However, Applying ACO with VRPTW encounters the problem of trapping in local optimum. The effect of trapping in local optimum originates from the pheromone of ants. This paper proposes improving ACO by resetting pheromone when trapping in local optimum occurs. The resetting pheromone is similar to run again. The results from the previous round searching are useless. Hence, this paper proposes the result from the previous round searching apply with the pheromone to guide the searching next round. The proposed algorithm was tested on fifty-six maps from a series of Solomon and provided more satisfactory results in comparison with other ACO algorithms.**

Keywords—Ant Colony Optimization; Vehicle routing problem with time windows; trapping in local optimum

I. Introduction

The vehicle routing problem with time windows (VRPTW) is NP-hard problem and a famous problem for comparison of algorithms performance [1]-[3]. VRPTW originates from transportation problem, logistics problem, scheduling problem and supply chain management problem.

In the recent years, many researchers have proposed using meta-heuristic to solve VRPTW because the results are highly accurate in effectively solving large and complex problems such as Genetic Algorithms (GA) [4]-[7], TABU search [8], [9] and ACO [10]-[15]. ACO has successfully been applied to solve vehicle routing problem (VRP) [16], [17]. Hence, this paper focuses on ACO apply to solve VRPTW only. ACO is motivated by the foraging behavior of ant colony [18]-[20]. ACO has been widely used for solving optimization problems [21] such as job scheduling problem, traveling salesman problem (TSP), network routing and VRP etc.

The advantages of ACO [21], [22] are rapid discovery of good answers and good efficiency for TSP or similar problems. However, the disadvantage of ACO [21], [23] is that it often encounters problem of trapping in the local optimum. To overcome the disadvantages of ACO, many researchers [14], [15], [24], [25] increase searching diversity by adding the mutation technique and resetting technique into the process of ACO. Moreover, improving answer of ACO, many researchers [13], [14], [15], [25] add the neighborhood search into the process of ACO.

The creation answers of ACO gets influence from pheromone of ants. If the pheromone is unchanged, the high chance that ACO create the answers that are similar to the previous answers [25]. Hence, resetting pheromone leads to create the new answer of ACO. The resetting pheromone can help to search that is not waste after trapping in local optimum occur but it cannot improve searching of ACO. The answers from resetting are rarely better than the answers from normal ACO. In order that the resetting pheromone can improve searching of ACO, this paper proposed that the best answer found in searching from the previous reset round (the previous best or PBEST) and the best answer found in the whole searching (the global best or GBEST) apply with ACO after the resetting pheromone.

A set of the well-known Solomon fifty-six maps [26], [27] is used to compare other ACOs, and the proposed algorithm. All best paths of Solomon fifty-six maps are presented by [28]. The results show that the quality and reliability of answers of the proposed algorithm are better than other compared algorithms.

II. Related Work

A. Vehicle Routing Problem with Time Windows

The goal of VRPTW is to find a set of minimum cost routes to transport goods to all the customers during a given interval without exceed capacity of vehicle [29, 30]. All vehicle routes start and end at a supply depot. Each customer is visited only once. The formulation of VRPTW is described as the following:

$$x_{ijk} = \begin{cases} \dfrac{1 \quad if\_the\_vehicle\_k\_travels\_from\_i\_to\_j}{0 \quad else} \end{cases} \tag{1}$$

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

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

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

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

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

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

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

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

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

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 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 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 is to minimize the total travelling cost (C_{ij}), is given by (2). Maximum number of routes constraint is given by (3). A customer should visit only once, is given by (4) and (5). All vehicles tours start and end 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 serves the customer in proper time interval, is given by (10).

B. 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:

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

where P_{ij}^{k} is the probability of ant k chooses 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 (β > 0). The result from formula (11) that causes selecting path which is shorter and has a greater amount of pheromone. After selection cities of ant has completed, the fitness of ant is calculated by (2). The fitness of each ant is used to update pheromones according to (12). The next step, 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 the sum of new enhanced pheromones at this edge is degree of dissipate for pheromones, 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 - ρ) represents the evaporation of trail between time t and t+n. Q is consistent value, L_k is the fitness of ant k. $\Delta\tau_{ij}^{k}$ is calculated by (14).

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

For VRPTW, each ant starting at a supply depot and it creates a route by selecting customers within condition of the vehicle capacity and the time windows constraints. The selecting customers depend on pheromone [25]. The process is performed repeatedly until all customers are selected.

C. The Neighborhood Search

The neighborhood search or local search is technique for improving answers from searching by changing or swapping one or two customers then gets the new answer that is better than the answer before it is improved [31].

Many researchers proposed the local search improve the answer from ACO. This paper uses three well-known techniques as follows: Customer exchange technique [13] is all possible valid permutation from exchanging customers between two routes in order to get the best answer from the exchanging mechanism. One move operator technique [13] is all possible valid permutation from inserting customers between two routes in order to get the best answer from the insertion mechanism. Two-opt technique [32] is all possible valid permutation from the swapping of two nodes in same route in order to get the best answer from the swapping mechanism.

D. The compared Algorithms

The Many researchers [13, 14, 15] try to solve problem of trapping in local optimum by adding the mutation technique and resetting technique into the process of ACO. These algorithms can summarize the process as follows:

The first algorithm is ant colony optimization and TABU to solve VRPTW (ACOT) that is proposed by [13]. TABU search is used to maintain the diversity of ACO and explore new answers. Two-opt, customer exchange, and one move operator apply with ACOT to improve the answer of ACOT. This algorithm is used the standard of ACO for solving VRPTW.

The second algorithm is Ant colony optimization and pheromone reset to solve VRPTW (ACOR) that is proposed by [14]. The result from resetting pheromone can solve or decrease trapping in local optimum. The pheromone reset is executed depend on the iteration rounds. Moreover, this algorithm proposed the new technique creating answer by choosing new edge not in previous answer in order to obtain the new answer that differ from the previous answer. Two-opt apply with ACOR to improve the performance of ACOR.

The third algorithm is Ant colony optimization and mutation operator to solve VRPTW (ACOM) that is proposed by [15]. This algorithm added three mutation operators into process of ACO and new technique for updating pheromone. These mutation operators are the interchange operator, the shifting operator, and the inverse operator. These mutation operators enhance the local search ability and allow for random global explorations. Moreover, these techniques prevent trapping in local optimum.

Three algorithms (ACOT, ACOR, ACOM) are tested with the Solomon's problems. The results from testing show that answers of these algorithms are better than that of competitive algorithms.

III. Proposed Work

ACOT and ACOM have methods to reduce and avoid trapping in local optimum using the TABU search technique and the mutation technique, respectively. However, ACOT and ACOM have not methods to deal with trapping in local optimum when it occurs. When trapping occurs, ACOT and ACOM rarely get a better answer. This event is similar to a finished search. ACOR has resetting pheromone method to deal with trapping in local optimum when it occurs. The resetting pheromone of ACOR is not appropriate because resetting is executed according to iteration. The resetting may be executed when trapping in local optimum is not happen, it obstructs searching answer of ACO.

The creation answers of ACO gets influence from pheromone of ants. If the pheromone is unchanged, the high chance that ACO create the answers that are similar to the previous answers [25]. This event leads to trap in local optimum. After trapping in local optimum occurs, the searching answer is useless. To searching is not wasted, all pheromones should be reset. The simple techniques to indicate trapping in local optimum state of ACO is to monitor the unchanged number of consecutive best answer [33].

However, the resetting all pheromone of ants is similar to new run again. The resetting that cannot solve problem of trapping in local optimum and it has the high chance that trapping at the previous location occurs. Hence, the resetting rarely increases searching performance of ACO. Moreover, the result from searching before reset is discarded and useless.

In order to improve searching of ACO and solve trapping in local optimum, this research proposed that use PBEST and GBEST update pheromone of ant after resetting pheromone. Both PBEST and GBEST are the best results from previous searching. The reason selects PBEST and GBEST because if pheromone is updated from the good answer, it can generate better answer [25]. Hence, searching next round has high chance to get the better answer.

Before reset, the pheromone table creates from many path of searching of ant. This pheromone table creates answers lead to trap in local optimum. After reset, the pheromone table is updated from the best path only. Effect from updating table pheromone with the best path, the density of pheromone from the best path is more than the density of pheromone from path others. From influence of this pheromone, ACO creates new answers that are close to the previous best answer but it differs from the previous answer much enough. The new answer shows that ACO can jump out local optimum and can solve trapping in local optimum.

For these reasons, this research proposed that checking status of trapping in local optimum from the unchanged number of consecutive PBEST and reset all pheromones when trapping occurs, and then pheromone is updated by PBEST and GBEST. The proposed algorithm is used to solve or decrease problem of trapping in local optimum and increases searching performance of ACO after resetting. This proposed algorithm is called a novel modified ant colony optimization by resetting and updating pheromone or ACORU. Pseudo code of ACORU is shown below:

```
Initialize edges, PBEST, GBEST, and pheromone
reset GR, TP = 0
While termination condition ≠ true do
  For ant x begin 1 to a number of all ants
    For until customers in list of ant x is full
      The chosen customer is not repeat customer in list of ant x
      The chosen customer passes condition of capacity and time windows
      choose the next customer with probability according to formula (11)
      If customers cannot add into ant x
        Add the new vehicle
      End If
    End For
    Evaluate the fitness of ant x
    Apply customer exchange with ant x
    Apply one move operator with ant x
    Apply 2-opt with ant x
    If the fitness of ant x is better than that of PBEST
      PBEST = ant x
    End If
    If the fitness of ant x is better than that of GBEST
      GBEST = ant x
    End If
    Update pheromone according to formula (13) by ant x
  End For
  If PBEST is not improve
    TP++
  End If
  If TP >= RP
    All pheromones are initialized
    Reset PBEST, TP = 0
    For 1 to RPI
      update pheromone according to formula (13) by PBEST and GBEST
    End For
  End If
  Apply the evaporation
End While
```

Where TP is the times of PBEST consecutive unchanged. RP is the round of resetting pheromone. RPI is the round of density of initial pheromone.

IV. Results and Discussions

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 list of ant that is produced by the nearest neighbor heuristic [34] and n is the number of customers. 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 5000. For proposal algorithm, RP = 30 and RPI = 10. The non-ACO parameters for the compared algorithm parameters (ACOT, ACOR, ACOM) are set according to suggested by the original papers. This research is conducted by a personal computer of AMD FX-8320 with 3.5 GHz CPU, 16 GB RAM and Visual

C++ 2005 as the programming language. All datasets are used in experiment are from Solomon's VRPTW instances [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 running (10 runs). ABF indicates the answer searching efficiency of an algorithm. SD is the standard deviation. SD indicates the answer searching reliability of an algorithm.

TABLE I. COMPARATIVE ABF AND SD OF ACOT, ACOR, ACOM, ACORU

Problem	best known	ACOT		ACOR	
		ABF	SD	ABF	SD
C101	828.93	1201.83	98.25	1040.15	97.25
C102	828.93	1231.75	45.99	1185.21	46.36
C103	828.06	1209.11	66.71	1164.93	33.86
C104	824.78	1043.41	29.72	1074.52	52.72
C105	828.94	1172.86	29.56	1062.93	68.17
C106	828.94	1192.22	38.86	1146.43	37.64
C107	828.94	1162.69	38.20	1102.04	45.48
C108	828.94	1057.51	38.46	1134.11	38.72
C109	828.94	1050.19	23.59	1060.37	47.79
C201	591.56	751.879	73.13	691.902	69.99
C202	591.56	973.329	39.98	936.391	70.91
C203	591.17	927.677	34.12	1000.2	46.51
C204	590.6	868.697	27.79	940.573	63.80
C205	588.88	704.152	29.02	666.758	7.15
C206	588.49	685.699	42.79	697.549	25.17
C207	588.29	633.472	1.85	655.293	34.38
C208	588.32	659.61	13.50	652.195	20.71
R101	1650.8	2195.44	13.52	2138.42	78.22
R102	1486.86	1991.68	30.52	1906.93	61.43
R103	1292.67	1657.31	18.93	1623.09	61.91
R104	1007.31	1265.42	30.64	1263.56	88.77
R105	1377.11	1679.21	18.99	1700.55	77.37
R106	1252.03	1594.08	25.93	1611.52	60.39
R107	1104.66	1385.58	28.71	1403.36	41.30
R108	960.88	1148.39	13.35	1204.76	44.08
R109	1194.73	1307.14	14.50	1414.16	77.91
R110	1118.84	1248.64	30.32	1339.02	37.21
R111	1096.73	1304.67	26.97	1323.2	98.51
R112	982.14	1056.42	11.85	1162.57	22.79
R201	1252.37	1546.61	14.82	1540.48	81.58
R202	1191.7	1398.96	25.00	1367.84	32.44
R203	939.5	1240.73	18.78	1291.79	31.34
R204	825.52	975.635	6.76	1041.07	29.02
R205	994.43	1199.44	26.01	1253.39	50.47
R206	906.14	1160.11	18.12	1194.89	57.85
R207	890.61	1130.84	7.33	1132.23	50.70
R208	726.82	884.707	11.45	952.151	50.95
R209	909.16	1173.12	11.91	1210.74	49.42
R210	939.37	1297.06	62.81	1257.73	22.45
R211	885.71	931.842	9.85	1026.6	50.55
RC101	1696.95	2458.17	49.66	2319.51	176.59
RC102	1554.75	2162.1	33.80	2117.54	75.23
RC103	1261.67	1831.77	28.46	1755.17	65.15
RC104	1135.48	1330.64	28.48	1409.73	76.60
RC105	1629.44	2229.67	39.68	2079.18	89.25
RC106	1424.73	1603.67	27.54	1645.97	102.95
RC107	1230.48	1359.82	36.80	1527.65	50.52
RC108	1139.82	1269.26	13.65	1360.43	85.72
RC201	1406.94	1850.15	25.42	1760.05	67.36
RC202	1365.64	1686.04	27.20	1634.45	84.09
RC203	1049.62	1403.27	19.16	1442.23	60.81
RC204	798.46	1060.76	47.11	998.337	21.31
RC205	1297.65	1751.77	111.81	1713.92	107.73
RC206	1146.32	1471.78	72.34	1413.23	21.77
RC207	1061.14	1369.08	20.52	1447.03	42.57
RC208	828.14	1015.15	23.57	1148.33	31.72

TABLE I. (CONTINUED).

Problem	best known	ACOM		ACORU	
		ABF	SD	ABF	SD
C101	828.93	940.01	33.16	828.93	0.00
C102	828.93	943.33	23.34	846.994	10.06
C103	828.06	947.81	20.03	874.162	10.59
C104	824.78	916.51	12.99	868.268	7.71
C105	828.94	944.55	17.35	837.742	13.47
C106	828.94	967.15	18.97	869.265	10.70
C107	828.94	951.65	18.61	891.141	9.96
C108	828.94	961.21	12.09	936.85	7.52
C109	828.94	956.821	10.70	912.80	5.79
C201	591.56	768.34	54.74	591.56	0.00
C202	591.56	856.71	23.13	591.56	0.00
C203	591.17	835.43	35.54	593.272	4.11
C204	590.6	797.87	37.14	604.507	3.40
C205	588.88	673.69	41.04	588.88	0.00
C206	588.49	614.42	14.23	588.49	0.00
C207	588.29	625.87	42.42	588.29	0.00
C208	588.32	614.21	6.79	588.32	0.00
R101	1650.8	1758.50	16.58	1650.8	0.00
R102	1486.86	1587.49	8.39	1486.86	0.00
R103	1292.67	1299.85	9.27	1292.67	0.00
R104	1007.31	1042.69	13.68	1007.31	0.00
R105	1377.11	1558.27	18.06	1377.11	0.00
R106	1252.03	1433.71	17.91	1252.03	0.00
R107	1104.66	1218.13	15.88	1104.66	0.00
R108	960.88	1006.40	11.52	960.88	0.00
R109	1194.73	1279.37	8.66	1194.73	0.00
R110	1118.84	1147.79	11.72	1118.84	0.00
R111	1096.73	1146.69	9.67	1096.73	0.00
R112	982.14	982.14	0.00	982.14	0.00
R201	1252.37	1546.43	14.96	1252.37	0.00
R202	1191.7	1365.05	14.62	1191.7	0.00
R203	939.5	1124.70	38.45	939.5	0.00
R204	825.52	884.48	10.27	825.52	0.00
R205	994.43	1327.59	8.80	1022.92	5.82
R206	906.14	1244.78	14.69	938.885	9.41
R207	890.61	1062.44	19.07	890.61	0.00
R208	726.82	825.44	15.98	736.512	4.88
R209	909.16	1122.93	9.21	909.16	0.00
R210	939.37	1210.43	12.30	944.398	5.28
R211	885.71	885.71	0.00	885.71	0.00
RC101	1696.95	1861.77	23.37	1696.95	0.00
RC102	1554.75	1657.06	13.95	1554.75	0.00
RC103	1261.67	1395.25	20.23	1261.67	0.00
RC104	1135.48	1151.49	7.53	1135.48	0.00
RC105	1629.44	1686.38	20.43	1629.44	0.00
RC106	1424.73	1463.75	25.87	1424.73	0.00
RC107	1230.48	1249.34	10.16	1230.48	0.00
RC108	1139.82	1139.82	0.00	1139.82	0.00
RC201	1406.94	1829.45	23.52	1406.94	0.00
RC202	1365.64	1594.82	20.22	1365.64	0.00
RC203	1049.62	1237.36	13.79	1049.62	0.00
RC204	798.46	928.45	15.86	817.796	6.73
RC205	1297.65	1583.59	24.02	1297.65	0.00
RC206	1146.32	1549.34	17.67	1154.45	6.50
RC207	1061.14	1333.80	17.30	1061.14	0.00
RC208	828.14	828.14	0.00	828.14	0.00

From the experimental results of Table 1 show that the quality answer of ACORU is better than that of ACOT, ACOR, and ACOM because of its lowest ABF all tested maps. The reliability of ACORU is better than that of ACOT, ACOR, and ACOM because of its lowest SD in all tested maps. ACORU can solve trapping in the local optimum and increases searching performance of ACO that are better than ACOT, ACOR, and ACOM. So, ACORU can get better results than ACOT, ACOR, and ACOM.

V. CONCLUSION

Applying ACO with VRPTW encounters the problem of trapping in local optimum. In order to solve trapping in local optimum, the technique of resetting pheromone is apply with ACO. However, resetting all pheromone rarely enhance searching performance of ACO. Moreover, all results from searching before resetting is discarded. In order to solve or decrease problem of trapping in local optimum and increases searching performance of ACO after resetting, this paper proposed that reset all pheromones when trapping occurs, and then pheromone is updated by PBEST and GBEST. The proposed technique is called that ACORU. ACORU can solve trapping in local optimum problem and improve searching of ACO after resetting. Hence, it can get better answer. From experimental results on fifty-six maps from a series of Solomon, all algorithms (ACOT, ACOR, ACOM, ACORU) are tested and the results are compared. The results indicate that the proposed ACORU outperforms ACOT, ACOR, and ACOM with regard to the reliability and quality of answers in all the experiments.

ACKNOWLEDGMENT

This research was funded by Faculty of Applied Science King Mongkut's University of Technology North Bangkok, Thailand Contract no. 651091

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, "the truck dispatching problem," Management Science, pp. 80-91, 1959.

[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] 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.

[4] O. Bräysy and M. Gendreau, "genetic algorithms for the vehicle routing problem with time window," internal report STF42 A01021, 2001.

[5] C. B. Cheng and K. P. Wang, "solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm," expert systems with applications, pp. 7758–7763, 2009.

[6] J. Y. Potvin, T. Kervahut, B. L. Garcia, and J. M. Rousseau, "the vehicle routing problem with time windows," part i: tabu search, informs journal on computing, pp. 158–164, 1996.

[7] H. Nazif and L.S. Lee, "optimized crossover genetic algorithm for vehicle routing problem with time window," american journal of applied sciences, pp. 95-101, 2010.

[8] E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, and J. Y. Potvin, "a tabu search heuristic for the vehicle routing problem with time windows," transportation science, pp. 170–186, 1997.

[9] W. C. Chiang and R. A. Russell, "a reactive tabu search meta-heuristics for the vehicle routing problem with time windows," informs journal on computing, pp. 417–430, 1997.

[10] W. Xudong, "a two stage ant colony optimization algorithm for the vehicle routing problem with time window," IJACT: international journal of advancement in computing technology, vol.4, no.1, pp. 485-491, 2012.

[11] C.H. Chen and C.J. Ting, "a hybrid ant colony system for vehicle routing problem with time window," journal of the eastern asia society for transportation studies, vol. 6, pp. 2822-2836, 2005.

[12] Sandhaya and V. Katiyar, "an enhanced ant colony system for solving vehicle routing problem with time window," international journal of computer application, vol.73, no.12, pp. 27-31, 2013.

[13] B. Yu, Z.Z. Yang, and B.Z. Yao, "a hybrid algorithm for vehicle routing problem with time windows," expert systems with applications, pp.435–441, 2011.

[14] Ashima Gupta and Sanjay Saini, "An enhanced ant colony optimization algorithm for vehicle routing problem with time windows," 2017 ninth international conference on advanced computing (ICOAC), pp. 267-274, 2017.

[15] Huizhen Zhang, Qinwan Zhang, Liang Ma, Ziying Zhang, and Yun Liu, "A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows," information sciences, 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, pp. 196, 2009.

[18] A. Colorni, M. Dorigo and V. Maniezzo, "distributed optimization by ant colonies," proceedings 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 Lm. Gambardella, "a study of some properties of ant-Q," proceedings of the 44th international conference on parallel problem solving from nature, berlin: springer-verlag, pp. 656–665, 1996.

[23] A. Colorni and M. Dorigo, "heuristics from nature for hard combinatorial optimization problems," international trans operational research, pp.1–21, 1996.

[24] C. Ratanavilisagul, "modified ant colony optimization with pheromone mutation for travelling salesman problem," 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 re-initialization 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

[28] http://sun.aei.polsl.pl/~zjc/best-solutions-solomon.html#c201

[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," INF. SCI., 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 res, 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, 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.