An Enhanced Ant Colony Optimization Algorithm for Vehicle Routing Problem with Time Windows

Ashima Gupta
Department of Physics & Computer Science
Dayalbagh Educational Institute
Agra – 282 005, India
ashimagupta17@gmail.com

Sanjay Saini
Department of Physics & Computer Science
Dayalbagh Educational Institute
Agra – 282 005, India
sanjay.s.saini@gmail.com

***Abstract*—Path optimization is a critical issue in vehicle routing problems (VRPs). This study tends to solve a VRP with time windows (VRPTW), which is an important version of VRP. Its principal objective is to find out the minimum cost routes to serve the customers within service time intervals, by a fleet of vehicles having confined capacity. An improved ant colony optimization (ACO) is applied to the problem, which uses a new pheromone reset and update function to enhance the route searching and a 2-opt method to improve the optimized path. The proposed ACO is tested on standard benchmarks of VRPTW. The results verifies the effectiveness of the algorithm and confirms it as better path optimizing method for VRPTW, in comparison to the traditional ACO and other heuristics.**

Keywords—* *VRP; vehicle routing problem; VRPTW; time window; meta-heuristic; ACO.

I. Introduction

The VRP with time window has received great attention of researchers all around the world, due to its significance in routing, transportation scheduling, and other practical problems. This widely studied variant of VRP (fig. 1) is an important field for lifelike problems and generalizations of the fundamental routing model. VRPTWs is described as planning the minimum cost set of routes to service clients in specific time periods, by a group of homogeneous vehicles having restricted capacity. All vehicle routes starts and ends at a central depot by visited only one client.

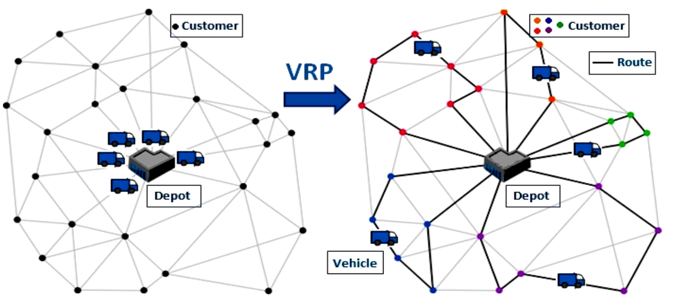

Fig. 1. Classical Vehicle Routing Problem

Many researchers have contributed in this field of study as in Schrage [1] introduced routing and scheduling with time constrains. Wang and Lang [2] proposed multi-period VRP with recurring time windows. People applied VRPTW in different application fields like dial-a-ride systems, collection of waste, school bus routing, real-time traffic handling, working hour regulations, and also in achieving multiple objectives. Li, Tian and Leung [3] solved ambiguous and uncertain VRP using fuzzy systems. Kim and Sahoo [4] used it for waste collection business.

In recent years, various versions of VRPTW have been derived such as, soft time window VRPs (VRPSTW), in which time limitations are violated by paying some extra cost. Similarly, in hard time window VRPs (VRPHTW), violations are not permitted, than simultaneous pickup and delivery (VRPTW-SPD), VRPTW with stochastic service times (VRPTWST) where time varies randomly, VRPTW with potential demands (VRP-PDTW) and the case of split deliveries (VRPTW-SD). Also, multiple issues are combined together like multi-trip VRP with heterogeneous fleet and driver working hour constraint [5]. Comprehensive reviews on VRPTW were given by many people; such as: Yang and Liu [6] Ahumada and Villalobos [7], Cordeau et.al. [8], Sherbeny and Naseer [9], Calvete et.al. [10], Gupta and Saini [11].

There exists numerous of techniques aimed to solve VRPTW problems, which are classified as: precise (exact) methods, heuristics and meta heuristics. In the past years, meta-heuristics has been frequently applied on VRPs, due to their high-precision quality to solve large and complex problems effectively. These techniques include: TS (tabu search) [12], SA (simulated annealing) [13], GA (genetic algorithm) [14, 15], PSO (particle swarm optimization) [16], ACO (ant colony optimization) [17-20] etc. The article under study mainly focuses on the VRPTW solutions formed using ACO algorithm only.

A. Literature on ACO Solutions to VRPTW

Gambardella, Taillard and Agazzi [21] build MACS-VRPTW (multi ant colony system) to solve VRPHTW with multiple objectives. In their algorithm one colony optimized the routes and the second colony minimizes the distance. An enhanced ACS (ant colony system) algorithm was proposed by Sandhaya and Katiyar [20] for VRPTW. The performance of the algorithm was evaluated on standard benchmarks and results are compared with the best available results in literature. The above two algorithms are considered as the best ACO algorithms for solving the VRPTWs, as per the literature.

Reimann, Doerner and Hartl [22], presented an AS (ant system) for VRPBTW (backhauls), in which they use an insertion method for solution construction. Results are compared with the solutions of other heuristics exists. Rizzoli et.al. [23] implemented ACO for VRPTW, VRPPD and for two industrial problems. Their result outcomes were able to solve these problems up to near optimal.

Time dependent model combined with LS (enhanced local search) was presented by Donati et.al. [24]. This new version of the algorithm performs exceptionally good for both optimality of solutions and speed. A hybrid DSACA-VRPTW, presented by Zhang Zhen [25], was tested on Solomon's datasets. Computational results proved it as an efficient approach to find VRPTW solutions.

Ting and Chen [26] combined MACS with SA and tested it on several benchmarks. They found it an effective technique to solve MDVRPTW (multi depot), when compared to other heuristics.

Yang and Yu [27] solved the periodic VRPTW problem using an improved version of ACO algorithm and tested the performance on standard benchmarks. They found their solutions much better than the reported results, which proved IACO as a competitive tool for solving periodic VRPTW. Balseiro [28] implemented hybrid ACO by combining it with insertion approach for TDVRPTW.

Zha, Luo and Yin [29] presented a modified MMAS (max-min ant system) method and applied it to solve VRPTW variations. Simulation results verified on well-known benchmarks, proved this method as highly competitive in achieving quality of solutions. Veen et al.[30] modified the MACS algorithm for dynamic problems in order to solve DVRPTW (dynamic). They extended their study by designing new benchmarks for dynamic problems, by considering a case study of a Dutch mobile surveillance company.

In the remaining sections, section II gives the problem description and mathematical model for VRPTW. Section III discusses the enhanced ACO algorithm, followed by the relevant performance of the algorithm on standard benchmarks and result comparison with other heuristics in section IV. Finally, section V conclude the study with future research goals.

II. VRPTW Formulation

A. Problem Description

In VRPTW a time constraint is added to the problem. In VRPTW the fleets of vehicles have restricted capacity (which can't be exceeded), traversed a set of routes to fulfil customers' requests, within a specific time frame. It is defined by a weighted graph $G = (V,A)$, as $V = \{0, 1, 2, 3,..., N\}$ are vertices and A is the edge set. The depot from where every route starts and ends is denoted by vertex *0* and the *N* customers or cities are represented by *1 to N* vertices.

Each customer has a non-negative request q_i, which is *0* for the depot. For each edge $_{i,j}$ there is a distance or travelling cost C_{ij}, which is computed using Euclidian distance between the customers. The fleets of vehicles that serve the customers have total capacity Q. An interval $[et_i, lt_i]$ (earliest and latest arrival time resp.) is associated to each client, during which customers have to be served and vehicle has to wait till service completion. Moreover, vehicle cannot arrive at the customer after lt_i, and also vehicle has to wait, if it reaches before et_i i.e. the beginning of service time.

Therefore, the total time can be computed as: (a) time at which vehicle leave the station (b) total travel time t_i (c) the service time s_i. The goal is to find a set of optimized tours to serve all the clients by strictly following the listed constraints:

- A client should be serviced only once.
- Each vehicle route open and close at the depot.
- Sum of demands for any vehicle route must not exceed its capacity.
- Clients are serviced only in the given time window.

B. Mathematical Model of VRPTW

Mathematically VRPTW can be defines as:

$$x_{ijk} = \begin{cases} 1 & \text{if the vehicle } k \text{ travels from i to j} \\ 0 & \text{else} \end{cases} \quad (2.1)$$

where, x is the decision parameter. The goal of VRPTW is,

$$\min = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} C_{ij}\, x_{ijk} \quad (2.2)$$

subject to,

$$T_{Ei} \leq t_i \leq T_{Li} \quad (2.3)$$

$$\sum_{j=0}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \quad (\mathrm{i} \in \mathrm{N}) \quad (2.4)$$

$$\sum_{i=0}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \quad (\mathrm{i} \in \mathrm{N}) \quad (2.5)$$

$$\sum_{j=1}^{N} x_{ijk} = \sum_{i=1}^{N} x_{jik} = 1 \quad (i = 0, k \in K) \quad (2.6)$$

$$\sum_{i=0}^{N} \sum_{j=0}^{N} x_{ijk} * q_i \leq Q \quad (2.7)$$

In the above model of VRPTW the primary objective is to minimize the total travelling cost C_{ij}, is given by equation 2.2. The time window constraint, which ensures that the vehicle serves the customer in proper time interval, is given by 2.3. Here, $[T_{Ei}, T_{Li}]$ is time window for customer *i*, and t_i is the travel time between customers *i* and *j*. Equation 2.4, 2.5 shows that a client should be visited by only one vehicle. Condition 2.6 means all vehicles tours open and close at the depot. Constraint 2.7 gives the capacity restriction i.e. the total quantity of goods that a vehicle carries should be less than its capacity *Q*.

III. Enhanced Ant Colony Optimization Algorithm

A. Basic ACO

For this article we purposely chose Ant Colony Optimization (ACO), due to an ongoing research on solving CVRP and VRPTW using ACO metaheuristic. ACO was proposed by Dorigo in 1992. It is a probabilistic technique to solve large complexities optimally [31]. It is achieved by seeking the foraging behavior of ants to find a shortest path from food to the nest, with the help of a trace called pheromone that permit them to communicate with each other [32]. The standard ACO algorithm is given below:

```
procedure ACO_Algorithm
    while(not_termination)
            Construct_Solution()
            Local_Search()
            Pheromone_Update()
    end while
end procedure
```

Researchers developed many successful ACO algorithms, for the large and complex problems in different domains. Table 1 list out these famous ACO algorithms in chronological order.

TABLE I. VARIANTS OF ACO ALGORITHM

Year	Author	Algorithm
1991	Dorigo.et.al	Ant System (AS)
1995	Gambardella & Dorigo	Ant-Q
1996	Dorigo, Maniezzo, Colorni	Elitist AS (EAS)
1997	Dorigo & Gambardella	Ant Colony System (ACS)
1997	Stutzle & Hoos	Max-Min AS (MMAS)
1999	Bullnheimer, Hartl, Strauss	Rank Based AS (ASrank)
1999	Gambardella et al.	Multi Ant Colony System (MACS)
2001	Blum, Roli & Dorigo	Hypercube AS (HAS)
2013	Sandhaya & Katiyar	Enhanced ACS

B. Proposed ACO algorithm for VRPTW

The proposed ACO method is different from the basic ACO by these factors:

1. Two customers from different tours are swapped, if it can enhance the solution.
2. After some iterations, pheromone is reinforced for all the edges (Bullnheimer et al. ACO) [33] and rose for some edges that found best solution, to achieve the exploitation.
3. Each ant has an in-build memory (to store current solution) and a counter (iterations for which solution is not improved).

Hence, solution is improved instead of constructing a new solution, in each run. Each ant starts its tour at some random vertex and then chooses an edge from the neighborhood. The tour is constructed as:

$$P_{i,j} = \begin{cases} [T_{ij}]^{\alpha}[\eta_{ij}]^{\beta} / \sum_{h \epsilon \Omega} [T_{ih}]^{\alpha}[\eta_{ih}]^{\beta} & if\ v_j \epsilon \Omega \\ 0 & otherwise \end{cases} \tag{3.1}$$

Where, $p_{i,j}$ is the probability for selecting an *edge(i,j)*, $T_{i,j}$ is the pheromone present on *edge(i,j)*, visibility of the edge is given by $\eta_{i,j}$, *f* and *g* are the scaling factors. The pheromones are updated as:

$$\mathrm{T}_{ij}^{new} = \rho\, \mathrm{T}_{ij}^{old} + a \sum_{\mu=0}^{\sigma-1} \Delta \mathrm{T}_{ij}^{\mu} + b\ \Delta \mathrm{T}_{ij}^{*} \tag{3.2}$$

$$\Delta \mathrm{T}_{ij}^{\mu} = (\sigma - \mu) / L_{\mu} \tag{3.3}$$

Where, ρ denotes the trail persistence with $(0 \leq \rho \leq 1)$, Δ_{ij}^{μ} is the increase in trail scaled by factor *a*. $\Delta T_{i,j}^{*} = 1 / L^{*}$ denotes the amount of pheromone deposited scaled by parameter *b*. L^{*} is the best-so-far solution. Fig. 2 gives the flowchart of enhanced ACO algorithm.

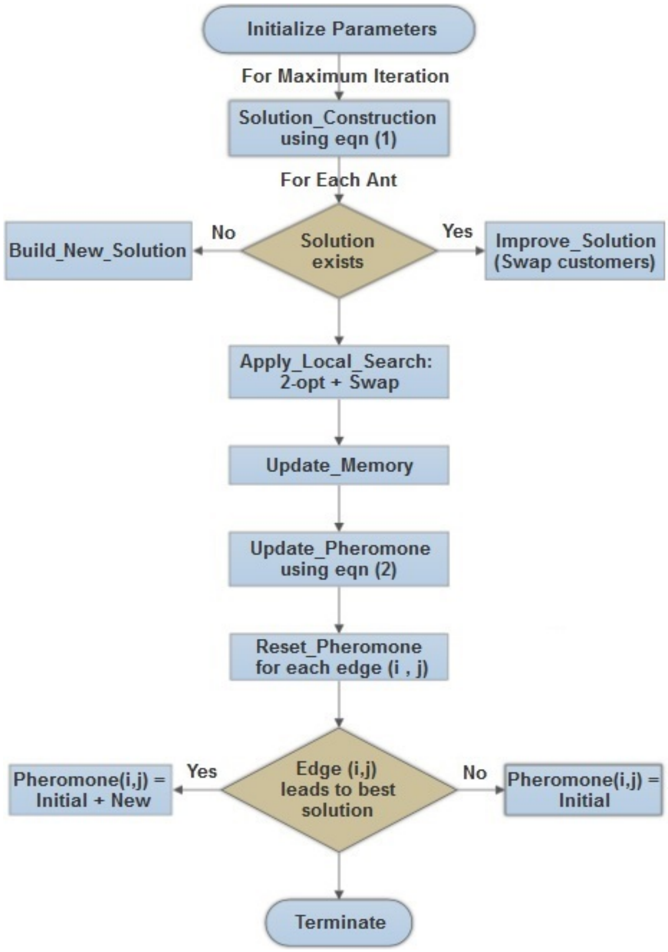

Fig. 2. Flowchart of the enhanced ACO algorithm

The pseudo code for enhanced ACO is given as:

1) *Initialize Parameters*
   For maximum Iterations:
2) *Solution Construction:*
   For each ant:
      if Previous_Solution = Null
         Build New Solution starting from depot
      else Improve Previous_Solution as follows:
         a) Choose new edge not in Previous_Solution
            that lead to maximum saving
         b) New_Tour = Old_Tour + new edge
                  + Build remaining solution
3) *Apply Local Search: 2-opt + Swap*
4) *Update Memory*
   For each ant:
      Previous_Solution = New_Solution
      if New_Solution_Cost < Previous_Solution_Cost
         Count = 0   % no. of times solution not improved
      else Count = Count + 1
         if Count > Max_Count
            Previous_Solution = Null
            Count = 0;   % reset count
5) *Update Pheromone*
6) *Reset Pheromone*
   if Iteration % Max_R ==0
      Reinforce Pheromone for each edge (i, j) as:
         if (i, j) belongs to best Solution
            Pheromone (i, j) = Initial + New Pheromone
         else
            Pheromone (i, j) = Initial Pheromone

C. Steps in enhanced ACO algorithm

- Step I: *Initialize:* pheromone matrix, Visibility matrix, Distance matrix, No. of ants, Neighbourhood size, Number of elitist ants etc.
- Step II: *Solution construction:* each ant start at some random vertex *V* and then selects one edge from its neighbourhood using probability given in equation 3.2.
- Step III: *Local search:* customer are swapped from different routes, if it can improve the solution quality. and routes are reordered if they crosses over itself.
- Step IV: *Memory update:* memory and count, that are associated to each ant are updated. Whenever, count exceeds max_count, pheromones are reinforced (go to to step VI).
- Step V: *Pheromone updates:* pheromones are updated using equations 3.2 and 3.3.
- Step VI: *Reset Pheromone:* pheromones are reinforced for all the edges, and increased for the edges that gives best solutions.

D. Parameters Used

The proposed algorithm has been coded in MATLAB 2015 and simulations were performed on 2.93 GHz i7 Octa Core machine. The enhanced ACO was evaluated on Solomon's' VRPTW benchmarks for 100 customers. These benchmarks contain total 56 instances (C1, C2, R1, R2, RC1, RC2 type), where in C type clients are clustered, while in R type they are remotely distributed. In RC group customers' distribution is the mixture of R and C instances.

Set C1 consists of 9 instances ranging from C101 to C109 and set C2 has C201 to C208 instances. R1 has 12 test problems from R102 to R112 and R2 has R211 as last instance. Set RC1 and RC2 contain 8 instances each. In each test problem customers have non-negative demands and Euclidian type distances.

Also, three time components are associated to each customer i.e. ready_time, due_date, and service_time. The difference between due date and ready time will be considered as the time window for each customer. Other than that vehicle count and capacity of vehicles is specified in each instance. Table 2 below listed the parameters used for simulating enhanced ACO algorithm for VRPTW.

TABLE II. PARAMETERS USED BY THE ALGORITHM

Parameter	Value
Population size	*N-1, (Customers no. = N)*
Nearest neighborhood	*NN = N/4*
Elitist ants	$\sigma = 10$
Trail_persistence	$\rho = 0.9$
Initial pheromones	$T_0 = 1.0$ & $T_1 = 1.2$
Max_Count	*K = 20*
Max_R	*R = 20 (to reset pheromone)*
Scaling factors	$\alpha = \beta = 5$ & $f = g = 2$,
Maximum iterations	*300*

Initial pheromone is set to $T_0 = 1.0$ (tuned from *0.92*), as it is a good practice to set the initial pheromone slightly higher than the trail [9] persistence ($\rho = 0.9$, tuned from *0.80)*. Moreover, for exploiting more edges the pheromone is increased as $T_1 = 1.2$ (tuned from *2.0*).

To reduce the number of parameters tuning we fixed $\alpha = \beta$ *and* $f = g$. For majority of the instances, the value setting $\alpha = \beta = 5$ *and* $f = g = 2$, gives the best compromise between the solution optimality and computational speed. The maximum iterations were simulated using $\sigma = 10$ elitist ants (ranked ants), only that can contribute to update the pheromones.

IV. RESULTS AND COMPARISON

This section discusses the computational results obtained by evaluating enhanced ACO algorithm on six different sets of Solomon's instances, by showing the plots. The comparison of results between proposed algorithm and existing heuristics will also be presented.

Fig. 3 shows the vehicle route plot for C101 instance in which a set of 100 customers is served by 10 vehicles, where each vehicle has maximum capacity equal to 200.

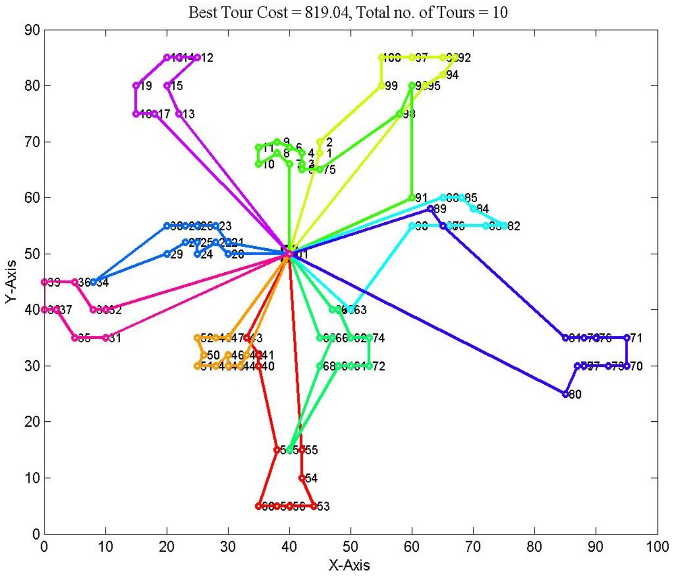

Fig. 3. Plot for VRPTW, Instance: C101, Customers: 100, Vehicles: 10, Capacity: 200, Best_known: 827.30, **eACO Solution: 819.04**

Fig. 4 shows the routes chosen by the vehicles to satisfy the customers' demands. In each route a set of customers served by a vehicle, which start and ends at the depot. The total demand of each route is less than or equal to the vehicle capacity for example we can clearly see the demand for route 1 is 190 which is less than the vehicle capacity. For each route total number of customers served is also displayed. Hence, the enhanced ACO algorithm is able to optimize cost, while fulfilling the VRPTW constraints.

```
Capacity of each vehicle is : 200
Routes Traversed by Ants are :
Route 1 :  Demand : 190   No. of Customers served : 10
 101   43   41   40   59   60   58   56   53   54   55   101
Route 2 :  Demand : 130   No. of Customers served : 10
 101   42   44   45   46   48   51   50   52   49   47   101
Route 3 :  Demand : 200   No. of Customers served : 9
 101    1    2   99   100   97   93   92   94   95   101
Route 4 :  Demand : 180   No. of Customers served : 13
 101    7    8   10   11    9    6    4    3    5   75   98   96   91   101
Route 5 :  Demand : 190   No. of Customers served : 11
 101   67   65   62   74   72   61   64   57   68   66   69   101
Route 6 :  Demand : 180   No. of Customers served : 8
 101   88   85   84   82   83   86   90   63   101
Route 7 :  Demand : 190   No. of Customers served : 12
 101   21   23   26   28   30   34   29   27   25   24   22   20   101
Route 8 :  Demand : 180   No. of Customers served : 11
 101   80   79   77   73   70   71   76   78   81   87   89   101
Route 9 :  Demand : 190   No. of Customers served : 8
 101   13   15   12   14   16   19   18   17   101
Route 10 :  Demand : 180   No. of Customers served : 8
 101   32   33   36   39   38   37   35   31   101
Total Number of Customers Served by all vehicles : 100
```

Fig. 4. Plot Routes chosen by all the vehicles to visit their customers with the total demand for each route

Fig. 5 shows the tour plot for instance C201, in which 100 customers are served by 3 vehicles, here each vehicle has capacity limit equal to 700. In this instance a large number of customers are covered by each vehicle as there are only 3 vehicles at service.

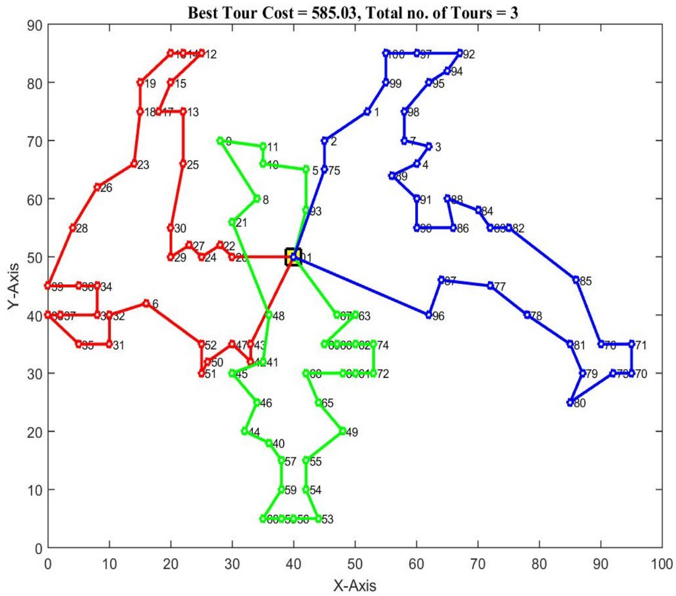

Fig. 5. Plot for VRPTW, Instance: C201, Customers: 100, Vehicles: 3, Capacity: 700, Best_known: 589.15, **eACO Solution: 585.03**

Fig. 6 below shows the routes traversed by each vehicle to fulfill the demands in C201 instance. Here, total of 3 vehicles service a set of 100 customers. In each route number of customers' served along with the sum of demands for that route is displayed. Like in route 1 total 34 customers are served and the sum of their demands is 630.

```
Capacity of each vehicle is : 700
Routes Traversed by Ants are :
Route 1 :  Demand : 630   No. of Customers served : 34
101   20   22   24   27   29   30   25   13   17   15   12   14
16   19   18   23   26   28   39   36   34   33   37   38
35   31   32    6   52   51   50   47   42   43   101
Route 2 :  Demand : 610   No. of Customers served : 33
101   67   63   69   66   62   74   72   61   64   68
65   49   55   54   53   56   58   60   59   57   40   44
46   45   41   48   21    8    9   11   10    5   93   101
Route 3 :  Demand : 570   No. of Customers served : 33
101   96   87   77   78   81   79   80   73   70   71
76   85   82   83   84   88   86   90   91   89    4    3
 7   98   95   94   92   97   100  99    1    2   75   101
Total Number of Customers Served by all vehicles : 100
```

Fig. 6. Routes travesed by the vehicle in order to fufill customer' demands alongwith the sum of demands for each route

Table 3 shows the results obtained by enhanced ACO algorithm for Solomon's VRPTW instances. First column represents the instance name and second column dedicated to the best known solutions for each instance. The best known values for all the six sets are taken from [34]. Column 8 shows the solutions of our enhanced ACO algorithm which are compared with the results of different heuristics (column 3-7).

Bold values in the 8th column denote the solutions which are better than the heuristics and best known solutions.

TABLE III. COMPUTATIONAL RESULTS OF ENHANCED ACO FOR VRPTW INSTANCES, COMPARED WITH THE BEST KNOWN AND OTHER HEURISTICS RESULTS

Set	Best_ known [A]	S-PSO [B]	CRO [C]	OCGA [D]	ACO-TS [E]	BCO_ SIH [F]	eACO [G]
C1							
C101	827.30	828.93	828.94	828.94	828.93	828.94	**819.04**
C102	827.30	829.71	828.94	828.94	828.94	828.94	**824.49**
C103	826.30	851.37	828.06	828.06	828.06	835.71	**817.86**
C104	822.90	868.52	824.78	824.78	828.21	885.06	**822.34**
C105	827.30	828.93	828.94	828.94	828.90	828.94	**816.74**
C106	827.30	828.93	828.94	828.94	828.94	828.94	828.94
C107	827.30	828.93	828.94	828.94	828.94	828.94	**809.60**
C108	827.30	828.93	828.94	828.94	830.94	831.73	**817.05**
C2							
C201	589.1	591.55	591.56	591.56	591.58	591.56	**585.03**
C202	589.1	591.55	591.56	591.56	591.56	591.56	**582.31**
C203	591.17	591.17	591.17	591.17	593.25	593.21	**584.68**
C204	590.6	615.43	590.6	590.60	595.55	606.90	**584.30**
C205	588.88	588.57	588.88	588.88	588.88	588.88	**586.13**
C206	588.49	588.87	588.49	588.49	588.49	588.88	**586.53**
C207	588.29	591.35	588.29	588.29	588.88	590.59	**582.62**
C208	588.32	588.49	588.32	588.32	588.03	593.15	**585.52**
R1							
R101	1607.7	1652	1648.3	1667.4	1655	1643.2	**1596.9**
R102	1434	1500.8	1476.6	1480.7	1491.1	1476.1	1448.5
R103	1175.7	1242.6	1213.6	1254.6	1243.2	1245.8	**1154.6**
R104	982.01	1042.2	976.6	1005.3	982	1026.9	**978.52**
R105	1346.1	1385.1	1366.7	1372.6	1380.4	1361.4	**1343.5**
R106	1234.6	1294.8	1254.1	1263.5	1265.3	1264.5	**1192.8**
R107	1051.8	1123.9	1079.8	1095.1	1100.2	1108.1	1067.7
R108	960.88	1011.6	951.12	972.13	958.66	994.7	**944.56**
R109	1013.2	1277.6	1166.3	1201.5	1102	1168.9	1026.2
R110	1068	1190.3	1087.9	1119.1	1119.5	1108.2	**1050.1**
R111	1048.7	1102.9	1084.5	1088.5	1091.1	1080.8	1056.7
R112	953.63	1029.1	1007.4	994.22	974.73	992.22	**948.62**
R2							
R201	1252.4	1274.9	1158.8	1248.8	1214.2	1197.1	**1229.1**
R202	1158.9	1247	1066.3	1079.3	1105.2	1092.2	1167.2
R203	939.50	1052.7	930.48	965.7	960.14	983.06	**908.28**
R204	825.52	844.16	766.95	813.90	771.47	845.30	**822.60**
R205	994.42	1061.4	958.46	994.88	1050.2	999.54	**973.46**
R206	833	1016.3	902.81	928.96	954.85	955.94	**828.23**
R207	814.78	946.77	818.56	847.54	870.33	903.59	836.52
R208	726.75	834.72	720.30	725.42	777.72	769.96	**706.45**
R209	855	1003.2	866.40	890.27	934.21	935.57	**831.7**
R210	939.34	1040.5	941.63	946.55	949.02	988.34	944.53
R211	820	861.32	820.29	888.73	877.55	867.95	**812.13**
RC1							
RC101	1619.9	1641.2	1644.5	1652.1	1650.1	1637.4	**1554.8**
RC102	1530.8	1510.9	1466.4	1496.2	1514.8	1486.8	**1492.3**
RC103	1261.7	1294.7	1282.6	1306.1	1277.1	1299.4	**1249.6**
RC104	1135.4	1190.5	1173.4	1177.6	1159.3	1200.6	1147.5
RC105	1554.7	1603.7	1533.3	1618.5	1617.8	1535.8	**1528.4**
RC106	1262.1	1410.9	1394.9	1382.2	1387.6	1403.1	1277.9
RC107	1222.2	1249.7	1211.1	1243.2	1280.1	1230.3	1226.2
RC108	1133.9	1181.8	1150.6	1128.4	1157.4	1165.2	**1114.7**
RC2							
RC201	1134.9	1423.5	1254.9	1387.5	1279.6	1315.5	1142.6
RC202	1130.5	1193.6	1095.6	1176.9	1157	1169.7	**1084.3**
RC203	1026.6	1123.4	952.8	1082.9	1046.3	1010.7	1031.8
RC204	798.41	795.48	899.34	836.13	847.33	890.28	**794.7**
RC205	1297.2	1321.4	1236.1	1270.6	1334.5	1221.3	**1231.9**
RC206	1112.2	1307.9	1118.1	1149.7	1112.2	1097.6	**1094.2**
RC207	1040.7	1130.3	1028.7	1053.5	1078.5	1024.2	**1028.4**
RC208	828.14	958.23	817.9	816.10	911.15	864.5	**808.5**

[A] Best known for VRPTW [34]
[B] Discrete PSO for VRPTW [35]
[C] Chemical reaction optimization[36]
[D] Crossover genetic algorithm[14]
[E] ACO and Tabu search [37]
[F] Modified ABC for VRPTW [34]
[G] Proposed eACO algorithm

Highlighted values in the table denote the new best solution found by our eACO algorithm, which are better than the best known solutions, reported in the literature. However, due to different simulation setup of the heuristics, comparison between execution times is not meaningful.

Fig. 7 shows the cost v/s iteration plot for C101 instance, which clearly shows that the solution is improving continuously and finally settles down to an optimal value of 819.04 after some 220 iterations.

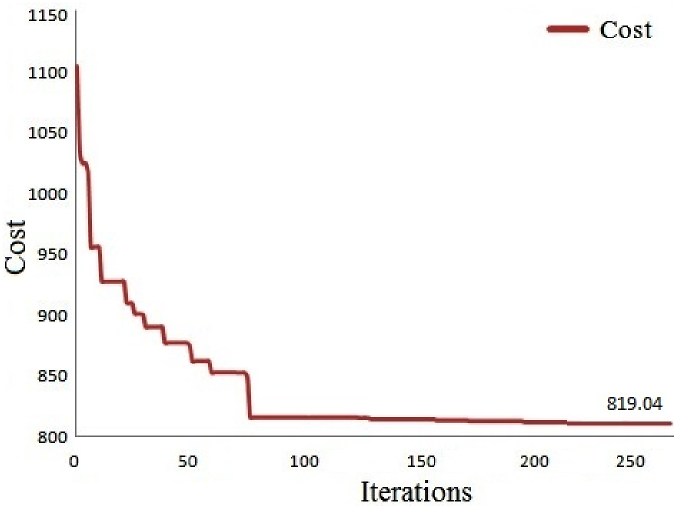

Fig. 7. Plot Number of Iterations v/s Cost plot for C101 instance

V. CONCLUSION

This paper discusses the enhanced ACO algorithm, which was evaluated on the Solomon's VRPTW instances of 100 customers. From the results, we can clearly see the improvements in solutions that are obtained using enhanced ACO algorithm. The obtained solutions (table 3) are found better than the best known results for 42 instances out of 56 (bold values). The proposed algorithm was earlier tested on capacitated VRP instances (reported [38]), which proves it as a competitive approach to solve VRPs. Hence, to conclude, the eACO is capable of solving VRPTWs optimally and can compute new solutions for most of the problems. Thus, the proposed eACO can possibly optimize real time problems which may reduce the effective cost of goods, transportation and makes an effective impact on our economy.

In future, the aim is to evaluate eACO on large VRPTW instances, and also on different variants of VRP. Further, it would be more interesting to improve the evaluation speed of eACO algorithm through parallel implementation and hybridization with other intelligent techniques.

REFERENCES

[1] L. Schrage, "Formulaton and structure of more complex / realistic routing and schedulng problems," Networks, vol. 11, no. 2, pp. 229–232, 1981.

[2] Y. Wang and M. Lang, "Study on the model and tabu search algorithm for delivry and pickup vehicle routing problem with time window," in proc. IEEE International Conference on Service Operations and Logistics and Informatics, pp. 1464–1469, 2008.

[3] X. Li, P. Tian, and S. C. H. Leung, "Vehicle routing problem with time window and stochastic travel service time: model and algorithm," *International Journal of Production Economics*, vol. 125, no. 1, pp. 137–145, 2010.

[4] B. I. Kim, S. Kim, and S. Sahoo, "Waste collection vehicle routing problem with time window," *Computers and Operations Research*, vol. 33, no. 12, pp. 3624–3642, 2006.

[5] M. P. Seixas and A. B. Mendes, "Column generaton for a multi-trip vehicle routing problem with time window, driver work hour, heterogeneous fleet," Mathematical Problems in Engineering, vol. 2013, 13 pages, 2013.

[6] C. Yang, Z. Guo., L. Liu, "Comparison study on algorithm for vehicle routing problem with time window," In: E. Qi, J. Shen, R. Dou (eds) Proc. 21st International Conference on Industrial Engineering and Engineering Management, Atlantis Press, Paris, 2015.

[7] O. Ahumada and J. R. Villalobo, "Application of planning model in the agri food supply chain: a review," European Journal of Operational Research, vol. 196, no. 1, pp. 1–20, 2009.

[8] J. F. Cordeau, M. Gendreau, G. Laporte, J. Y. Potvin, and F. Semet, "A guide to vehicle routing heuristic," Journal of the Operational Research Society, vol. 53, no. 5, pp. 512–522, 2002.

[9] E. l. Sherbeny and A. Nasser, "Vehicle routing with time window: an overview of exact, heuristic and metaheuristic method," Journal of King Saud University Science, vol. 22, no. 3, pp. 123–131, 2010.

[10] H. I. Calvete, M.J. Oliveros, and B. S. Valverde, "A goal programmng approach to vehicle routing problem with sof time window," European Journal of Operational Research, vol. 177, no. 3, pp. 1720–1733, 2007.

[11] A. Gupta, S. Saini, "On Solution to Vehicle Routing Problem Using Swarm Optimization Technique: A Review," In: Bhatia S., Mishra K., Tiwari S., Singh V. (eds) Advances in Computer and Computational Sciences. Advance in Intelligent Systems and Computing, Springer, Singapore, vol 553, pp. 345-354, 2017.

[12] QI. Mingyao, Z. Ying, Z. Jinjin, and M. Lixin, "A new two phase hybrid metaheuristic for vehicle routing problem with time window," Journal of the Eastern Asia Society for Transportation Studies, vol. 9, pp. 880-896, 2013.

[13] S.R. Thangiah, H. O. Ibrahim, and S. Tong, "Hybrid genetic algorithm, simulated annealing and tabu search method for vehicle routing problem with time window," Computer Science Department, Slipery Rock University, Technical Report, 1994.

[14] H. Nazif and L.S. Lee, "Optimized crossover genetic algorithm for vehicle routing problem with time window," American journal of applied sciences, vol. 7, no. 1, pp. 95-101, 2010.

[15] B. Minocha and S. Tripathi, "Two phase algorithm for solving VRPTW problem," International Journal of Artificial Intelligence and Expert Systems (IJAE), vol. 4, no. 1, pp. 1-16, 2013.

[16] S. Masrom and A. M. Nasir, "Evaluation of vehicle routing problem with time window by using metaheuristic algorithm," Recent Researches in Mathematical Methods in Electrical Engineering and Computer Science, 2012.

[17] 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.

[18] 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.

[19] S. Bansal, R. Goel, and C. Mohan, "Use of ant colony system in solving vehicle routing problem with time window constraint," Proceedings of the Second International Conference on Soft Computing for Problem Solving, Springer India, 2014.

[20] 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.

[21] L.M. Gambardella, E.Taillard, and G.Agazzi, "MACS-VRPTW: a multiple ant colony system for vehicle routing problem with time window," In D. Corne and M. Dorigo (eds.), New Idea in Optimization, McGraw-Hill, UK, 1999.

[22] M. Reimann, K. Doerner, and R.F. Hartl, "Insertion based ant for vehicle routing problem with backhaul and time window," Ant Algorithm: Third International Working, pp.135–148, 2002.

[23] A. E. Rizzoli, F. Oliverio, R. Montemanni, and L.M. Gambardella, “Ant colony optimization for vehicle routing problem: from theory to application,” Istituto Dalle Molle di Studi sull’ Intelligenza, 2004.

[24] V.Donati, R.Montemanni, E.Rizzoli, and L.M. Gambardella, “Time dependent VRP with multi ant colony system,” European Journal of Operational Research, vol. 185, no. 3, pp. 1174–1191, 2008.

[25] T. Zhen and Q. Zhang, “Hybrid ant colony algorithm for the vehicle routing with time windows,” Computing Communication, Control and Management, ISECS International Colloquium, pp.8–12, 2008.

[26] C.J. Ting and H. Chen, “Combination of multi ant colony and simulated annealing for the multidepot vehicle routing problem with time window,” Journal of Transportation Research Board, pp. 85–92, 2009.

[27] B.Yu. and Y.Z. Zhen, “An ant colony optimization: the periodic vehicle routing problem with time window,” Transportation Research: Logistics and Transportation Review, vol. 47, no. 2, pp. 166–181, 2011.

[28] S.R.Balseiro, I.Loiseau, and J. Ramone, “An ant colony algorithm hybridized with insertion heuristic for the time dependent vehicle routing problem with time window,” Computers and Operation Research, vol. 38, pp. 954–966, 2011.

[29] G. Zhao, W. Luo, R. Sun and C. Yin, "A modified max-min ant system for vehicle routing problems," 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, pp. 1-4, 2008.

[30] B. V. Ven, M. Emmerich, T. Back, and J. Kok, "Ant colony algorithm for the dynamic vehicle routing problem with time window," Natural & artificial computation in engineering and medical application, Springer, Berlin, pp. 1–10, 2013.

[31] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization. Computational Intelligence Magazine, IEEE, vol. 1, pp. 28-39, 2006.

[32] H. Parunak, Dyke, and S. Brueckner, “Engineering swarming system,” Methodologies and Software Engineering for Agent System, pp. 341-376, 2004.

[33] B.Bullnheimer, R.F. Hartl, and C.Strauss, “An improved ant system algorithm for the vehicle routing problem,” Annals of Operations Research, vol. 89, pp. 319–328, 1999.

[34] M. Alzaqebah, S. Abdullah, and S. Jawarneh, “Modifid artifiial bee colony for the vehicle routing problems with time windows,” Springer Plus, Springer International Publishing, vol. 5, 14 pages, 2016.

[35] Y.J. Gong, J. Zhang, R.Z. Huang, and Y. . Shi, “ Optimizing the vehicle routing problem with time window: a discrete particle swarm optimization approach,” IEEE Transactions On System, Man, and Cybernetic Part C: Applications and Review, vol. 42, no. 2, 2012.

[36] A.Y.S. Lam, V.O.K. Li, “Chemical-reaction-inspired metaheuristic for optimization,” IEEE Trans. Evolutionary Computation, vol. 14, no. 3, pp. 381–399, 2010.

[37] B. Yu, Z. Yang, and B.Z. Yao, “A hybrid algorithm for vehicle routing problem with time window,” Expert Syst. Appl., vol. 38, no. 1, pp. 435-441, 2010.

[38] A. Gupta, S. Saini, “On Solutions to Capacitated Vehicle Routing Problem Using an Enhanced Ant Colony Optimization Technique,” In: Perez G., Mishra K., Tiwari S., Trivedi M. (eds) Networking Communication and Data Knowledge Engineering. Lecture Notes on Data Engineering and Communications Technologies, Springer, Singapore, vol 3, pp. 257-266, 2017.