An Enhanced Ant Colony System for Solving Vehicle Routing Problem with Time Window

Sandhya
CSE Department
M. M. Engineering College,
M. M. University,
Mullana, Ambala, Haryana, India-133207

Vijay Katiyar (Member IEEE)
CSE Department
M. M. Engineering College,
M. M. University,
Mullana, Ambala, Haryana, India-133207

ABSTRACT
Vehicle Routing Problem with Time Window (VRPTW), an extension of VRP, is a complex combinatorial problem having many real life applications. It can be described as the problem of minimizing the total route cost while satisfying the capacity and time window constraint. Ant Colony System (ACS) is a meta heuristic that is often applied to solve VRPTW. In this paper an attempt has been made to enhance the already existing ant colony system to solve the problem efficiently. Experimentation with the Solomon data sheet is performed and compared with best known results available in literature.

Keywords
Ant Colony Optimization, Meta heuristics, Time Windows, Vehicle Routing Problem.

1. INTRODUCTION
Vehicle Routing Problem [1] is a complex combinatorial, multi constraint problem that holds a central place in logistics, transportation. Many different types of VRP can be generated to include aspects like customers demands, the vehicles size, network characteristics, which makes problem more difficult to solve. VRPTW is a generalization of VRP where vehicles are required to serve customers within a limited time period. The objective is to design routes that minimize traveling cost in terms of total vehicles and total time. Moreover it has been included in the category of NP-Hard problems, it cannot be solved to get optimal solution in reasonable time, especially when problem size is too large. Some exact methods [10] exist in literature to solve problem, but they produces poor results in large size instances. Heuristics [10] and metaheuristics [7] are suitable to solve VRPTW. They generate good results with reduced run time. Genetic Algorithm [9], Tabu Search [13], Simulated Annealing [13], Ant Colony Optimization [12] are some of the examples of Meta heuristics. Among these Meta heuristics, ACO is a positive feedback, distributed optimization technique, proposed by Dorigo [11], mimics the food seeking behavior of real ant colonies in nature. It has been successfully applied to TSP, quadratic assignment, job shop scheduling etc. Among the earliest studies was that of Bullnheimer [13] who used a hybrid ACS with 2 opt heuristic and saving algorithm to construct routes. However, the result is not competitive with other metaheuristics. Furthermore, Bullnheimer et al [14] makes use of candidate list to improve the results. The search speed of this algorithm was faster, but solution quality reduces with increase in problem size. Gambardella [8] proposed the multiple ant colony system, MACS-VRPTW, to improve best known results. Hybridization of ACS and insertion heuristic was presented in Balserio et al. [15].X. Hu et al [16] proposed an improved ACS coupled with saving algorithm and disaster operator to overcome the problem of getting in local optimal solution. In order to improve the performance of existing ACS Qiulei Ding et al [17] has used hybrid ant colony optimization with candidate list. However their state transition formula is not suitable for VRPTW as other factors except distance are not considered. Chengming Qi et al [5] presents a RACS-VRPTW, which hybridized ACS with randomized algorithm, only partial customers are randomly chosen to compute the transition probability. In [6] ACS is improved to get fast convergence speed, and used pheromones adjust strategies to prevent falling into local optimization in order to improve search results. This paper is an extension of [18].Here an enhanced ant colony system (EN_ACS) having new pheromone up-dating rule based on link quality is proposed. Moreover EN_ACS restricts the range of pheromone to avoid premature convergence of ACS. Also values for pseudo proportional rule are dynamically implemented to get better results Rest of the paper is organized as follows. Section 2 presents mathematical model for VRPTW. The problem solving methodology is described in Section 3 and computation results are discussed in Section 4. Finally Section 5 provides conclusion and future work.

2. MATHEMATICAL MODEL FOR VRPTW
The mathematical definition, objective function, notations, terms used in VRPTW can modeled as a Graph theoretic model. Let G= (V, A) be a graph where V = $\{v_0, v_1, \ldots, v_n\}$ denotes a node set having depot and customers , and A= $\{ (vi , v_j) : v_i , v_j \in V \}$ denotes links between them. Following are parameters and decision variable used during formulation of mathematical model.
Parameters:
N is the number of customers including depot (i.e., nodes)
a_i is the earliest time to allow the service used by customer i.
b_i is the latest time to allow the service used by the

customer i.
C_{ij} is the travel cost from node i to j (i.e the distance or time required for traveling from node i to node j.)
q_i is the demand at customer i.
Q is the total capacity of vehicle.
K is the total number of vehicles.
t_i is the travel time.
w_i is the waiting time if vehicle k arrives early at node i.
Decision Variable:

$$x_{ij}{}^{k} = \begin{cases} 1 & \text{if vehicle } k \text{ visits } j \text{ after } i,\ j \neq i \\ 0 & \text{otherwise} \end{cases}$$

The VRPTW can be stated as:
Minimize:

$$F=\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}{}^{k} C^{k}{}_{ij}$$

The objective of VRPTW is to minimize the cost function when subjected to some constraints. We categorize the constraints in three parts: route feasibility constraints, time constraints, and demand flow constraint. The following (1, 2, 3) specifies route feasibility constraint.

$$\sum_{k=1}^{K}\sum_{i=1}^{N} x_{0,i,k} \leq V \qquad (1)$$

$$\sum_{i\in N} x_{o,i,k} = \sum_{i\in N} x_{i,0,k} = 1 \qquad \forall k \in K \qquad (2)$$

$$\sum_{i\in N} x_{i,j,k} = \sum_{j\in N} x_{j,i,k} = 1 \qquad \forall k \in K \qquad (3)$$

Equation (4,5) guarantees time window constraint

$$t_i \geq a_i \qquad i=1,2,\ldots,n \qquad (4)$$

$$t_i + w_i \leq b_i \qquad i=1,2,\ldots,n \qquad (5)$$

Demand flow constraint is enforced by (6)

$$\sum_{i=1}^{N} q_i \leq Q \qquad (6)$$

In formulating the above mathematical model, the following assumptions have been made:

1. Identical vehicles with known capacities Q are used.
2. Demand of each customer is qi is known.
3. Every vehicle leaves the depot and returns to the depot.
4. Each customer is serviced by only one vehicle.
5. The total demand of any customer is not more than the capacity of the vehicle.

3. ENHANCED ACS FOR VRPTW

In order to make existing ACS suitable for VRPTW other factors like urgency to serve, bias factor, wait time in addition to distance is added and the state transition formula is changed accordingly as presented in [18]. To remove the premature convergence of ACS and to increase the probability of obtaining higher quality solutions, upper and lower limits are fixed for pheromone updating. Moreover pheromone trail is updated by taking into account only the best solution produced by the search to date. To increase the ability of algorithm for searching variety of routes dynamic values are assigned to search parameters. Pheromone updation is performed according to contribution of each link so that more delicate searches can be performed in the next cycle.

3.1.1 Solution steps

Step 1: Initialize every controlling parameter, define the repeated counter as nc=0, put m ants that simulates individual vehicle on the depot having empty customer set.

Step 2: Ant constructs its route by selecting next customer using probabilistic rule [18] taking into account both the visibility and the pheromone information. To avoid bias towards customers connected by short distance and having large amount of pheromone pseudo proportional rule [18] is followed. Moreover to improve the algorithm search ability the value of q_o is dynamically adjusted. For initial phases value of q_o is kept low and high for finial phases.

Step 3: After serving customer demand ant returns to depot when either of the capacity or time window constraint of depot is satisfied.

Step 4: Check if all customers are visited or not. If all are served go to step 5 else send a new ant to visit the remaining customers.

Step 5: Continue till all customers served.

Step 6: Calculate total distance traveled and the number of vehicles used to compute the objective function value for the complete route.

Step 7: Repeat step 1 to step 6 until maximum number of iterations are reached.

Step 8: After construction of routes ants laid down artificial pheromones. In next route only best ant updates this pheromones trails and the value of pheromone is bounded. Therefore the pheromone updating rule is:

$$\tau_{ij}(t+1) = \tau_{ij}(t)(1-\rho) + \Delta\tau_{ij}$$

Where $\Delta_{ij} = n\_t\_path(i,j) \big/ nc * 100$

Here ρ is parameter that controls rate of evaporation of pheromone and n_t_path(i,j) gives the number of times the path from i to j is traversed by the vehicles. nc is maximum permissible value specified by the user. In order to prevent the local optimization and to increase the probability of getting quality of solution upper and lower values of pheromone are specified to be [18].

4. RESULTS AND DISCUSSIONS

To evaluate EN_ACS, it is tested on classical Solomon's 56 benchmark problems. These problems are classified into six categories: R1, R2, C1, C2, RC1, and RC2, each of which contains depot, vehicle capacity, time window and 100 customers. Problem sets C1 has clustered customers with large time windows and large vehicle capacity. Problems in C2 type have narrow time window and a small capacity of vehicle. On the other hand customers in R1 and R2 set are randomly generated with longer and shorter time horizon respectively.

Table 1. Computation results for type 1 problems

Problem	Best Known nv/dist	EN_ACS	Gap (%)
R101	19/1645.79 [19]	19/1670.7	1.49
R102	17/1468.12 [19]	17/1505.52	1.29
R103	13/1292.68 [19]	13/1330.01	2.81
R104	9/1007.24 [19]	9/1018.24	1.08
R105	14/1377.11 [19]	14/1380.3	0.23
R106	12/1252.98 [19]	13/1270.7	1.47
R107	10/1104.66 [19]	10/1112.85	0.74
R108	9/960.88 [19]	9/970.01	0.94
R109	11/1194.73 [19]	11/1194.16	-0.05
R110	10/1118.59 [19]	10/1125.5	0.61
R111	10/1096.72 [19]	10/1098.7	0.81
R112	9/982.14[19]	9/990.4	0.83
Average	**12/1209.89**	**12/1222.26**	**1.01**
C101	10/827.3 [20]	10/827.3	0.0
C102	10/827.3 [20]	10/827.3	0.0
C103	10/828.06 [20]	10/828.06	0.0
C104	10/824.78 [20]	10/824.78	0.0
C105	10/828.94 [20]	10/828.94	0.0
C106	10/827.3 [20]	10/827.3	0.0
C107	10/827.3 [20]	10/827.3	0.0
C108	10/827.3 [20]	10/827.3	0.0
C109	10/828.94 [20]	10/828.94	0.0
Average	**10/827.47**	**10/827.47**	**0.0**
RC101	14/1696.94 [19]	14/1700.82	0.23
RC102	12/1554.75 [19]	13/1580.97	1.66
RC103	11/1261.67 [19]	12/1140.00	-10.67
RC104	10/1135.48 [19]	10/1153.5	1.56
RC105	13/1629.44 [19]	13/1640.23	0.66
RC106	11/1424.73 [19]	11/1450.72	1.79
RC107	11/1230.48 [19]	11/1280.66	3.92
RC108	10/1139.82 [19]	10/1152.30	1.08
Average	**12/1384.16**	**12/1387.40**	**0.03**

The EN_ACS is coded in Matlab 7.0 at Intel Core 2 Duo 2.0 Ghz. All the problems were run for maximum of 2500 iterations, Table 1 and 2 presents a summary of our results and their comparison with best known results available in literature. Averaged results for each instance are reported and compared in table 3. One can conclude from table 3 that we are able to find better results for C class problems and comparable results for other instances. Fig 1 compares the convergence rate of ACS and EN_ACS for the problem C203. Fig 2 shows the schematic routes for Problem C203 after 500 iterations.

Problem Type	**MACS**		**EN_ACS**	
	nv	**dist**	**nv**	**dist**
R1	12	1217.70	12	1222.26
R2	2.73	967.70	2.91	968.88
C1	10	828.40	10	827.47
C2	3	590.90	3	598.86
RC1	11.63	1382.40	12	1387.40
RC2	3.25	1129.20	3.5	1105.99

5. CONCLUSION

In this paper an enhanced EN_ACS is proposed for solving vehicle routing problem with time window constraint. An extensive computation study for 56 Solomon benchmark problems for 100 customers is done to prove the efficiency of algorithm. The proposed algorithm have obtained equal results for C set for every instance and comparable results for R and RC set with less calculation over-head and with fast convergence rate. The results are encouraging and can be further improved if EN_ACS is hybridized with other metaheuristics.

Table 2. Computation results for type 2 problems

Problem	Best Known nv/dist	EN_ACS	Gap (%)
R201	4/1252.37 [20]	4/1260.72	0.7
R202	3/1198.45 [20]	4/1203.45	0.4
R203	3/942.64 [20]	3/963.77	2.2
R204	2/854.88 [20]	2/900.01	5.3
R205	3/1013.47 [20]	3/1100.23	8.6
R206	3/833 [20]	3/833	0.0
R207	3/814.78 [20]	3/837.32	2.8
R208	2/726.82 [20]	2/737.52	1.5
R209	3/855 [20]	3/888	3.9
R210	3/955.39 [20]	3/983.22	2.9
R211	2/910.09 [20]	2/950.43	4.4
R112	-	-	-
Average	**2.82/941.54**	**2.91/968.88**	**2.9**
C201	3/591.56 [20]	3/591.56	0.0
C202	3/591.56 [20]	3/591.56	0.0
C203	3/591.17 [20]	3/591.17	0.0
C204	3/590.6 [20]	3/590.6	0.0
C205	3/588.88 [20]	3/588.88	0.0
C206	3/588.49 [20]	3/588.49	0.0
C207	3/588.29 [20]	3/588.29	0.0
C208	3/588.32 [20]	3/588.32	0.0
C109	-	-	-
Average	**3/589.86**	**3/589.86**	**0.0**
RC201	4/1249 [20]	4/1279.13	2.4
RC202	4/1164.25 [20]	4/1188.43	2.1
RC203	3/1060.45 [20]	3/1090.22	2.8
RC204	3/798.46 [20]	3/805.71	0.9
RC205	4/1302.42 [20]	4/1340.1	2.9
RC206	3/1158.81 [20]	4/1203.11	3.8
RC207	3/1068.86 [20]	3/1102.27	3.1
RC208	3/833.4 [20]	3/838.22	0.6
Average	**3.38/1079.46**	**3.5/1105.99**	**2.4**

Table 3. Average number of vehicles (nv) and average distance (dist) computed by MACS and EN_ACS

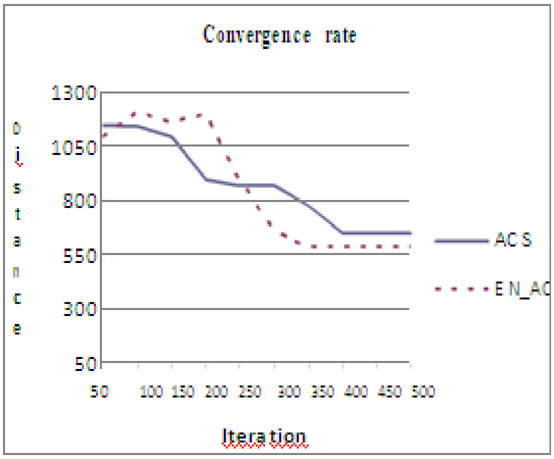

Fig.1 : Convergence Rate of ACS and EN_ACS

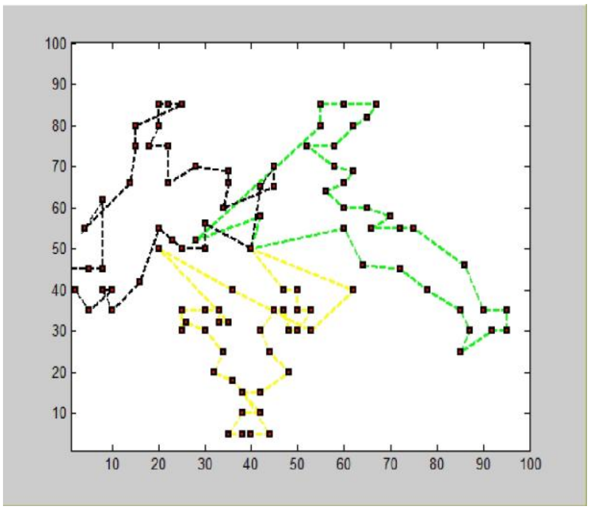

Fig.2: Schematic Routes for Problem C203 after 500 iterations

6. REFERENCES

[1] Laporte, Gilbert. "Fifty years of vehicle routing." Transportation Science 43.4 (2009): 408-416.

[2] Dantzig, George B., and John H. Ramser. "The truck dispatching problem." Management science 6.1 (1959): 80-91.

[3] Gambardella, Luca Maria, Éric Taillard, and Giovanni Agazzi. "Macs-vrptw: A multiple colony system for vehicle routing problems with time windows." New ideas in optimization. 1999.

[4] Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni. "Ant system: optimization by a colony of cooperating agents." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26.1 (1996): 29-41.

[5] Qi, Chengming, and Yunchuan Sun. "An improved ant colony algorithm for VRPTW." Computer Science and Software Engineering, 2008 International Conference on. Vol. 1. IEEE, 2008.

[6] Ma, Xin. "Vehicle Routing Problem with Time Windows Based on Improved Ant Colony Algorithm." Information Technology and Computer Science (ITCS), 2010 Second International Conference on. IEEE, 2010.

[7] Doerner, Karl F., and Verena Schmid. "Survey: metaheuristics for rich vehicle routing problems." Hybrid Metaheuristics. Springer Berlin Heidelberg, 2010. 206-221.

[8] Gambardella, Luca Maria, Éric Taillard, and Giovanni Agazzi. "Macs-vrptw: A multiple colony system for vehicle routing problems with time windows." New ideas in optimization. 1999.

[9] Zhu, Kenny Qili. "A new genetic algorithm for VRPTW." Proceedings of the International Conference on Artificial Intelligence. 2000.

[10] Jourdan, Laetitia, Matthieu Basseur, and E-G. Talbi. "Hybridizing exact methods and metaheuristics: A taxonomy." European Journal of Operational Research 199.3 (2009): 620-629.

[11] Dorigo, Marco, and Christian Blum. "Ant colony optimization theory: A survey." Theoretical computer science 344.2 (2005): 243-278.

[12] Soo, Raymond Kuo Yang, and Yong Haur Tay. "A survey on the progress of research on vehicle routing problem with time window constraints." Symposium on Progress in Information and Communication Technology (SPICT).

[13] Bullnheimer, Bernd, Richard F. Hartl, and Christine Strauss. "A new rank based version of the Ant System. A computational study." (1997).

[14] Bullnheimer, Bernd, Richard F. Hartl, and Christine Strauss. "An improved ant System algorithm for the vehicle Routing Problem." Annals of Operations Research 89 (1999): 319-328.

[15] Balseiro, S. R., Irene Loiseau, and Juan Ramonet. "An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows." Computers & Operations Research 38.6 (2011): 954-966.

[16] Hu, Xiangpei, et al. "An improved ant colony system and its application." Computational intelligence and security. Springer Berlin Heidelberg, 2007. 36-45.

[17] Hu, Xiangpei, et al. "An improved ant colony system and its application." Computational intelligence and security. Springer Berlin Heide-berg, 2007. 36-45.

[18] Sandhya et al. “Use of Ant Colony System Heuristic for solving Vehicle routing problem with time window constraint”, Scopros 2012 2nd international Conference on Soft Computing for Problem solving, proceedings in AISC series of Springer.

[19] http://www.sintef.no/Projectweb/TOP/VRPTW/Solomon-benchmark/100-customers/

[20] Chen, Chia-Ho, and Ching-Jung Ting. "A hybrid ant colony system for vehicle routing problem with time windows." Journal of the Eastern Asia Society for Transportation Studies 6.10 (2005): 2.