

Use of Ant Colony System in Solving Vehicle Routing Problem with Time Window Constraints

Sandhya Bansal, Rajeev Goel and C. Mohan

Abstract Vehicle routing problem with time window constraints (VRPTW) is an extension of the original vehicle routing problem (VRP). It is a well-known NP-hard problem which has several real-life applications. Meta-heuristics have been often tried to solve VRTPW problem. In this paper, an attempt has been made to develop a suitable version of Ant colony optimization heuristic to efficiently solve VRPTW problem. Experimentation with the developed version of Ant colony optimization has shown that it succeeds in general in obtaining results obtained with earlier version and often even better than the results that are better than the corresponding results available in literature which have been obtained using even previously developed hybridized versions of ACO. In many cases, the obtained results are comparable with the best results available in literature.

Keywords Vehicle routing problem · Ant colony optimization · Heuristics · Optimization

1 Introduction

Vehicle routing problem (VRP) [21, 22] lies at the heart of logistics and distribution management that is presently being used by the companies engaged in delivery and collection of goods. Since the conditions and constraints vary from one situation to another, several variants [3, 11, 18] of basic problem have been proposed in

S. Bansal (✉)
Maharishi Markandeswar University, Mullana, India
e-mail: Sandhya12bansal@gmail.com

R. Goel · C. Mohan
Ambala College of Engineering and Applied Research, Ambala, India
e-mail: rajiv_2709@rediffmail.com

literature. This paper addresses itself to developing an efficient Ant-colony-system-based heuristic for solving VRPTW.

VRPTW [23, 26] problem consists of finding the minimum set of routes for identical capacity vehicles originating and terminating at a central depot such that each customer is served once and only once, given that the exact number of customers and their demands are known. There are also constraints of time windows in that each customer must be served in a specified slot of time. Objective of VRPTW is to find the minimum of total distance travelled and/or the minimum number of vehicles required which can accomplish this job. It is a NP-hard problem where the number of feasible solutions grows exponentially as the number of customer's increases. The work on this problem available in literature can be divided into two classes: exact optimization techniques and heuristic-based (approximate) algorithms. In the first category, the works by [21, 24, 27] can be cited. The methods developed in these papers have been able to efficiently solve some of the Solomon benchmark problems [5, 20, 31]. In the second category, a very large number of heuristic approaches such as tabu search, genetic algorithms, ant colony algorithms, simulated annealing, large neighborhood search, variable neighborhood based algorithms, and multi-phase approaches have been tried [6, 7, 20, 26, 29].

Among heuristic-based optimization techniques, ACO is a more recent optimization heuristic proposed by Dorigo et al. [4, 12–16]. ACO imitates real ant behavior to search for optimal solutions. ACO-based optimization techniques tried thus far for solving VRPTW problem generally use distance and pheromone as search guide parameters. This paper proposes a version of ACO which incorporates besides these two parameters waiting time, urgency to serve and the bias factor also as search guide parameters. Our objective has been to see whether inclusion of these additional parameters can further improve the performance of ACO algorithm for solving VRPTW.

The rest of the paper is organized as follows. Section 2 presents mathematical model of VRPTW problem. Conventional use of ACO heuristic in solving VRPTW problem is presented in Sect. 3. Proposed modifications in this algorithm are presented in Sect. 3.2. Computational experimentation using the modified ant colony system (ACS) algorithm is presented in Sect. 4. Comparison of the obtained results with those earlier available in literature obtained through is done Sect. 5 and certain conclusions drawn.

2 Mathematical Description of VRPTW

In this section, we briefly describe the mathematical model of VRPTW.

The VRP is a complicated combinatorial optimization problem. It has received considerable attention in the past decades because of its practical importance in the fields of transportation, distribution, and logistics. VRPTW is a generalization of the classical VRP with the additional restriction of time window constraints. The

VRPTW can be modeled in mathematical terms through a complete weighted digraph as follows.

Let $G = (V, A)$ where $V = \{v_0, v_1, v_2 \dots v_n\}$ be a set of nodes, where v_0 represents the depot that holds a fleet of vehicles and $v_1, v_2 \dots v_n$ denote a set of n customers which are to be served by these vehicles. Each customer has an associated demand q_i , service time s_i , a service time window $[e_i, l_i]$. Also $A = [\{v_i, v_j\} (i, j = 0, 1, 2, \dots, n, i \neq j)]$ is the set of arcs connecting various nodes, having distance d_{ij} as weights. If a vehicle reaches a customer v_i before specified time e_i , it needs to wait until e_i in order to service the customer. The service has to be provided before close time of window at l_i . The depot has also time window $[e_0, l_0]$. No vehicle is to leave the depot before e_0 and all should come back before l_0 . The load-carrying capacity of all vehicles is same and all travel with identical constant speed. The objective of the VRPTW is to service all the customers without violating vehicle capacity constraints and time window constraints using minimum number of vehicles that travel minimum possible total distance.

Mathematically, the problem is usually expressed as:

Minimize

$$F = \sum_{i=0}^N \sum_{j=0}^N \sum_{K=1}^V c_{ij} x_{ij}^v \quad (1)$$

Subject to:

$$\sum_{v=1}^V \sum_{j=1}^N x_{ij}^v \leq V \quad \text{for } i = 0 \quad (2)$$

$$\sum_{v=1}^V x_{ij}^v = \sum_{j=1}^N x_{ji}^v \leq 1 \quad \text{for } i = 0 \text{ and } v \in \{1, \dots, V\} \quad (3)$$

$$\sum_{v=1}^V \sum_{j=0}^N x_{ij}^v = 1 \quad \text{for } i \in \{1, \dots, N\} \quad (4)$$

$$\sum_{v=1}^V \sum_{i=0}^N x_{ij}^v = 1 \quad \text{for } j \in \{1, \dots, N\} \quad (5)$$

$$\sum_{i=0}^N c_i \sum_{j=0}^N x_{ij}^v \leq q_v \quad \text{for } v \in \{1, \dots, V\} \quad (6)$$

$$\sum_{v=1}^V \sum_{i=0}^N x_{ij}^k (t_i + t_{ij} + s_i + w_i) = t_j \quad \text{for } j \in \{1, \dots, N\} \quad (7)$$

$$e_i \leq (t_i + w_i \leq l_i) \quad \text{for } i \in \{1, \dots, N\} \quad (8)$$

$$t_0 = w_0 = s_0 = 0 \quad (9)$$

$x_{ij} = 1$ if vehicle k travels from customer i to customer j , and 0 otherwise ($i \neq j$; $i, j = 0, 1, \dots, N$).

Here,

V denotes total number of vehicles,

N total number of customers,

c_i customer i ($i = 1, 2, \dots, N$) and c_0 delivery depot,

d_{ij} traveling distance between customer i and customer j ,

t_{ij} travel time between customer i and customer j ,

q_i demand of customer i ,

q_v loading capacity of each vehicle, (loading capacity of all vehicles are identical).

e_i earliest permitted arrival time of vehicle at customer i ;

l_i latest permitted arrival time of vehicle at customer i ;

s_i service time of customer i ;

This is an optimization problem in which, (1) is the objective function of the problem which is to be minimized subject to constraints (2)–(9).

In this optimization model, decision variables are as follows:

V total number of vehicles required;

t_i arrival time of vehicle V at customer i ;

w_i waiting time of vehicle at customer i before service can be started;

Objective function (1) ensures that total distance travelled by all the vehicles is minimized. The first constraint (2) specifies that there are at the most V vehicles going out of the depot. The set of constraint (3) ensures that every vehicle starts from and ends at the delivery depot. The next two sets of constraints (4) and (5) restrict the assignment of each customer to exactly one vehicle route. The next set of constraints (6) ensures that the loading capacity of no vehicle is exceeded. The constraints of

set (7) are the maximum travel time constraint. Remaining constraints (8) guarantee schedule feasibility with respect to time windows.

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

- Identical vehicles with known capacities Q are used.
- All vehicles travel with identical constant velocity.
- Every vehicle leaves the depot and returns to the depot within specified time window $[l_0, s_0]$.
- Demand of each customer is q_i is known.
- Each customer is serviced by one and only one vehicle.
- The total demand of any customer is not more than the capacity of the vehicle.

3 Use of ACS in Solving VRPTW

VRPTW being an NP-hard problem, its exact solution is not known in general. Therefore, large numbers of alternative algorithms have been proposed for solving it. In this section, we first present conventional ACS-based approach for solving VRPTW problem and then present our proposed modification in it. The basic philosophy of ACS approach is to use a suitable positive feedback mechanism to reinforce those arcs of the graph that belong to a good solution. This mechanism is implemented associating pheromone levels with each arc which are then updated in proportion to the goodness of solutions found.

While presenting ACS-based algorithm, for solving VRPTW, we shall use the term 'tour' to denote a set of routes followed by all ants (vehicles) which are able to serve all the nodes of the graph as per their requirements under specified conditions. Our problem is to determine an optimal tour. The ACS algorithm commonly used for solving VRPTW is given below.

3.1 ACS Algorithm for Solving VRPTW:

Step 1. Construction of an initial feasible route:

- (a) Each ANT starts from the depot and the set of customers included in its route is empty.
- (b) The ant selects the next customer to visit from the list of feasible customers based upon the probabilistic formula (10).
- (c) After serving the customer storage capacity and the time used thus far of the Ant is updated and the process continued. Ant returns to the depot when either of the capacity constraint or time window constraint of the depot is satisfied.

- (d) We next check whether all the customers have been served or not. If all the customers have been served, stop else send a new ant (vehicle) to visit the remaining destinations.
- (e) Continue till all customers served.
- (f) Calculate total distance travelled and the number of vehicles used and compute the objective function value for the complete route using (1) (which gives total distance travelled by all used vehicles).

Step 2. Construct a specified number of feasible tours as in step 1.

Step 3. From among these constructed feasible tours, choose the tour which uses minimum number of ants (vehicles). In case of a tie, choose the tour in which total travelled distance is minimum (or vice-versa depending upon whether greater priority is to be given to minimize the number of vehicles used or total distance travelled).

3.1.1 Selection of Next Customer

In setting up of a feasible tour, each ant constructs a path that visits certain customer before returning back to depot. In the previous studies using ACO/ACS for solving VRPTW, the ant (vehicle), currently located at node i , selects the next node j to move to using the transition rule,

$$j = \begin{cases} \arg \max_{j \in \phi_i} \left\{ \tau_{ij}^\alpha \eta_{ij}^\beta \right\} & \text{if } q \leq q_o \\ J & \text{otherwise} \end{cases} \quad (10)$$

where

$$\eta_{ij} = 1/d_{ij} \text{ is a heuristic-based parameter.} \quad (11)$$

and $J \in \phi_i$ is randomly chosen according to the probability

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{u \in \phi_i} \tau_{iu}^\alpha \eta_{iu}^\beta} \quad (12)$$

Here, set ϕ_i contains the cities not visited so far.

In (10), $q \sim U(0, 1)$, and $q_o \in [0, 1]$ is a user-specified value of parameter q .

In (11), d_{ij} is Euclidian distance between i and j and τ_{ij} is the amount of pheromone on the path between current location i and next possible location j . Also α, β are the positive constants that determine the importance of η verses τ .

The transition rule (10) creates a bias toward customers connected by short distances and having large amount of pheromone. The parameter q_o balances exploration and exploitation. if $q \leq q_o$, the algorithm exploits (favoring the best nearest customer). Otherwise if $q > q_o$, the algorithm explores selecting node $j \in \phi_i$ randomly.

3.1.2 Pheromone Update for New Tour

After construction of a complete feasible route, the pheromones are laid for the next path depending upon the total traveled distance (L) of that route. For each arc $v_i \rightarrow v_j$ that was used in the previous feasible route, the pheromone trail is increased by $\Delta\tau_{ij}$. Furthermore, part of existing pheromone is also allowed to evaporate [4, 11, 14, 17, 34]. Thus in the next route, pheromones are updated according to the following

$$\tau_{ij} = \tau_{ij} (1 - \rho) + \Delta\tau_{ij} \quad (13)$$

$$\text{where } \Delta\tau_{ij} = Q/L \quad (14)$$

Here, ρ is parameter that controls rate of evaporation of pheromone.

3.2 Proposed Modifications in ACS Algorithm

Following modifications have been introduced in the conventional algorithms for solving VRPTW problem using ACS heuristics.

1. Whereas earlier approaches using ant colony technique for solving VRPTW problem have primarily given importance to distance and pheromone only to guide the heuristic [3, 32, 34], we in our present study have used besides these two parameters such as urgency to serve, waiting time and bias parameter also for this purpose. In (11), heuristic-based parameter η_{ij} only gives importance to the distance in determining the heuristic parameter. However, it was observed on experimentation that in addition to the distance waiting time, urgency to serve, and biasing should also be given importance in deciding the choice of next customer [11, 32].

As a result in our present study choice of η_{ij} defined in (11) has been modified as:

$$\eta_{ij} = \frac{1}{(d_{ij} + w_j)^\lambda} * \frac{1}{(l_j - a_j)^\gamma} * \frac{1}{(I_{\max} - I_j)} \delta \quad (15)$$

In (13), a_j is arrival time of vehicle at customer j and w_j defined as

$$w_j = \begin{cases} e_j - a_j & \text{if } e_j > a_j \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

is the waiting time at customer j before service can be started. Also $l_j - a_j$, $a_j < l_j$, is the difference between the latest arrival time l_j and actual arrival time a_j at customer j . It is a measure of urgency of customer j to be served, emphasizing that those customers whose time window is going to expire soon be given priority. Also $I_{\max} - I_j$, (where I_j is the number of consecutive times the customer j who could be next visited from the present customer has not been visited and I_{\max} is a user-specified maximum permissible value of I_j ($I_j < I_{\max}$) is a measure of bias factor).

2. In order to prevent the slow convergence of the algorithm when specified number of initial tours have been generated, we update the pheromone for the next tour using the best solution among the solution provided by m feasible routes [17]. In order to prevent local optimization and increase the probability of obtaining higher quality of solution upper and lower values of pheromone have been specified as $1/\sum 2d_{0i}$ and $1/\min(d_{ij})$ respectively, where d_{0i} is distance from the depot to the customer.
3. In conventional studies, total travelled distance has been minimized irrespective of vehicles needed. However keeping in view the fact that cost of obtaining a vehicle and its maintenance is generally much more than fuel cost we have tried to minimize total number of vehicles required as a first priority and total travelled distance as a second priority.

4 Implementation of the Modified ACS Algorithm

In this section, we summarize our computational experience of using the modified ACS algorithm for solving some of the Solomon benchmark [5] problems.

Solomon generated a set of 56 problems which have been frequently used in literature to check the performance of the developed algorithm. This set is divided into three categories namely C, R, and RC. In C category problems, customers are clustered either geographically or according to the time windows. R types of problems have uniform distribution of customers. Category RC is hybrid of problems of R and C set. In our present study, we have chosen 15 problems of 25 customers, 10 problems of 50 customers, and 10 problems of 100 customers from all these three sets. To solve these problems, the proposed algorithm was coded in MATLAB 7.0 at Intel Core 2 Duo 2.0 GHz. After experimentation, it was observed that the following values of parameters proved most suited for solving these problems. The number of initial feasible tours = 10, $\alpha = 1$, $\beta = 1$, $\lambda = 5.5$, $\gamma = 3.5$, $\delta = 1$, $Q = 250$ and $q_0 = 0.9$. All problems were run for maximum of 2,500 iterations, Tables 1, 2 and 3 present a summary of our results and their comparison with the best-known routing solutions compiled from different heuristics available in literature as per our information.

It may be noticed that whereas in our study, we have used only ACS, most of the earlier studies with ACO/ACS usually are hybrid in the sense that after completion of search with ACO/ACS, search is further carried out with some other optimization

Table 1 Comparison of best-known results with the results generated by proposed algorithms for Solomon's 25 customers set problem

Problem	Best	Worst	Using conventional ACO's	Best known [Ref.]
C201	215.54/2	215.54/2	222.53/2	214.7/2 [10]
R101	618.33/8	619.17/8	625.23/8	617.1/8 [20]
R102	563.35/7	573.15/7	605.45/7	547.1/7 [20]
R105	556.72/5	556.72/5	600.13/5	530.5/6 [20]
R109	442.63/5	448.54/5	510.31/5	441.3/5 [20]
RC101	462.15/4	462.15/4	507.87/4	461.4/4 [20]
RC105	416.16/4	416.88/4	435.97/4	411.3/4 [20]
RC106	346.51/3	346.51/3	402.11/3	345.5/3 [20]
RC201	432.30/2	432.30/2	412.34/3	360.2/3 [10]
RC202	376.61/2	381.75/2	400.72/2	338.0/3 [18]
RC203	433.94/1	433.94/1	454.78/2	326.9/3 [19]
RC204	331.29/1	333.36/1	370.56/1	299.7/3 [8]
RC205	386.15/2	386.15/2	413.37/3	338.0/3 [23]
RC207	358.92/2	367.92/2	387.16/2	298.3/3 [19]
RC208	309.85/1	309.85/1	313.76/1	269.1/2 [8]

Table 2 Comparison of best-known results with the results generated by proposed algorithms for Solomon's 50 customers set problem

Problem	Best	Worst	Using conventional ACO's	Best known [Ref.]
C101	363.25/5	363.25/5	363.25/5	362.5/5 [20]
C201	444.96/2	444.96/2	402.43/3	360.2/3 [20]
C205	444.57/2	444.57/2	407.58/2	360.2/3 [20]
R101	1053.04/12	1054.84/12	1107.18/12	1044/12 [20]
R201	882.32/2	893.56/3	900.72/3	791.9/6 [20]
R202	869.42/2	870.06/2	898.68/3	791.9/6 [20]
R203	741.3/2	764.3/2	612.32/5	605.3/5 [20]
R206	711.6/2	711.6/2	645.56/4	632.4/4 [20]
R209	722.24/2	735.20/2	619.23/4	600.6/4 [20]
RC101	951.07/8	962.80/8	987.97/8	944/8 [20]

heuristic also (such as local search, genetic algorithm [7, 21, 27]). In order to compare our present results with performance of earlier versions of ACO only (without use of any hybrid), we repeated our experimentation with those versions without using any other add on optimization heuristic. The results of this study are also presented in the Table 1 (column 4) for comparison.

The proposed algorithm has produced some improved results with lesser number of vehicles used (however, with some increase in routing length compared to the best available in literature. This is due to the priority that we assigned to minimize the total number of vehicles used visa vis total distance travelled).

Table 3 Comparison of best-known results with the results generated by proposed algorithms for Solomon's 100 customers set problem

Problem	Best	Worst	Using conventional ACO's	Best known [Ref.]
C101	828.94/10	828.94/10	830.37/10 (828.94/10)* [9]	828.94/10 [30]
C102	874.20/10	875.36/10	917.53/10 (828.94/10)* [34]	828.94/10 [30]
C105	828.94/10	828.94/10	830.37/10 (828.94/10)* [9]	828.94/10 [30]
C106	856.18/10	857.91/10	875.71/10 (828.94/10)* [9]	828.94/10 [30]
C107	830.60/10	838.42/10	842.67/10 (828.94/10)* [34]	828.94/10 [30]
C201	591.56/3	591.56/3	594.23/3 (591.56/3)* [34]	591.56/3 [20]
C205	591.5/3	595.33/3	598.28/3 (588.88/3)* [25]	588.88/3 [20]
R101	1714.26/19	1725.65/19	1845.12/19 (1670.66/19)* [25]	1645.79/19 [30]
R102	1558.19/17	1575.69/17	1613.34/18 (1535.52/17)* [25]	1486.12/17 [30]
R105	1519.55/14	1544.86/14	1853.45/18 (1365.23/15)* [25]	1377.11/14 [30]

Note * indicates results available in literature using hybrid versions of ACO's

5 Conclusions

In this paper, a modified version of ACS is proposed.

In our proposed algorithm, we have modified the heuristic-based parameter and pheromone update rules, used in conventional ACO for solving VRPTW. An extensive computational study on a set of benchmark test problems has been conducted. The experimental results show that the proposed algorithm even when used by itself is competitive with the earlier versions of ACO even when these are hybridized with certain other heuristics. We have obtained certain results in which lesser number of vehicles are needed than those reported in literature. However, in most of such cases, total distance travelled is slightly greater. Lesser number of vehicles means less initial investment in purchase of vehicles and less maintenance cost. (However, there is slight increase in fuel cost if total distance travelled is more).

The results are encouraging and we propose to direct further study toward use of proposed algorithm for solving the dynamic VRPTW

References

1. Andres, F.M.: An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transp. Res. Part B* **43**, 438–447 (2010)
2. Azi, N., Gendreau, M., Potvin, J.Y.: An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *Eur. J. Oper. Res.* **202**(3), 756–763 (2010)
3. Balseiro, S.R., Loiseau, I., Ramone, J.: An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Comput. Oper. Res.* **38**, 954–966 (2011)
4. Bell, J.E., McMullen, P.R.: Ant colony optimization techniques for the vehicle routing problem. *Adv. Eng. Inform.* **18**, 41–48 (2004)
5. Best solutions to Solomon problems identified by heuristics. (SINTEF VRP page): <http://www.sintef.no/static/am/opti/projects/top/vrp/bknown.html>

6. Bianchessi, N., Righini, G.: Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Comput. Oper. Res.* **34**, 578–594 (2007)
7. Bräysy, O.: Local search and variable neighborhood search algorithms for the vehicle routing problem with time windows. University of Vaasa, Finland, Doctoral Dissertation (2001)
8. Chabrier, A.: Vehicle routing problem with elementary shortest path based column generation. *Comput. Oper. Res.* **33**(10), 2972–2990 (2006)
9. Chen, C.-H., et al.: A hybrid ant colony system for vehicle routing problem with time windows. *J. East. Asia Soc. Transp. Stud.* **6**, 2822–2836 (2005)
10. Cook, W., Rich, J.L.: A parallel cutting plane algorithm for the vehicle routing problem with time windows. Working Paper of Computational and Applied Mathematics, Rice University, Houston, TX (1999)
11. Donati, A.V., et al.: Time dependent vehicle routing problem. *Eur. J. Oper. Res.* (2006). doi:[10.1016/j.ejor.06.047](https://doi.org/10.1016/j.ejor.06.047)
12. Dorigo, M., Birattari, M., Stützle, T.: Ant colony optimization: Artificial ants as a computational intelligence technique. *IEEE Comput. Intell. Mag.* **1**, 28–39 (2006)
13. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. *Theor. Comput. Sci.* **334**, 243–278 (2005)
14. Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1**, 53–66 (1997)
15. Dorigo, M., Maniezzo, V., Colomi, A.: Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B. Cybern.* **26**, 29–41 (1996)
16. Gambardella, L.M., Taillard, E., Agazzi, G.: MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows *New Ideas in Optimization*. McGraw-Hill, London (1999)
17. Garcia-Najera, A., Bullinaria, J.A.: An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Comput. Oper. Res.* **38**(1), 287–300 (2011)
18. Irnich, S., Villeneuve, D.: The shortest path problem with k-cycle elimination ($k > 3$): Improving a branch-and-price algorithm for the VRPTW *INFORMS. J. Comput.* **18**(3), 391–406 (2003)
19. Kallehauge, B., Larsen, J., Madsen, O.B.G.: Lagrangean duality and non-differentiable optimization applied on routing with time windows. *Comput. Oper. Res.* **33**(5), 1464–1487 (2006)
20. Kohl, N., Desrosiers, J., Madsen, O.B.G., Solomon, M.M., Soumis, F.: 2-path cuts for the vehicle routing problem with time windows. *Transp. Sci.* **33**(1), 101–116 (1999)
21. Laporate, G.: The vehicle routing problem: An overview of exact and approximate algorithms. *Eur. J. Oper. Res.* **59**, 345–358 (1997)
22. Laporte, G.: Fifty years of vehicle routing. *Transp. Sci.* **43**, 408–416 (2011)
23. Larsen J.: Parallelization of the vehicle routing problem with time windows. Ph.D. thesis IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark (1999)
24. Lau, H.C., et al.: Vehicle routing problem with time windows and a limited number of vehicles. *Eur. J. Oper. Res.* **148**, 559–569 (2003)
25. Ma, X.: Vehicle routing problem with time windows based on improved ant colony algorithm. In *Proceedings of IEEE Computer Society Second International Conference on Information Technology and Computer Science*, Washington, DC, USA, pp. 94–97 (2010)
26. Masrom, S., Nasir, A.M., Abidin, S.Z., Rahman, A.S.A.: Software framework for vehicle routing problem with hybrid metaheuristic algorithms. In *Proceedings of the Applied Computing Conference*, Angers, pp. 55–61 (2011)
27. Mester, D.: An evolutionary strategies algorithm for large scale vehicle routing problem with capacitate and time windows restrictions. In *Proceedings of the Conference on Mathematical and Population Genetics*, University of Haifa, Israel (2002)
28. Qi, C., Sun, Y.: An improved ant colony algorithm for VRPTW. In *IEEE International Conference on Computer Science and Software Engineering* (2008)
29. Rizzoli, A.E., Montemanni, R., Lucibello, E., Gambardella, L.M.: Ant colony optimization for real-world vehicle routing problems. *Swarm Intell* **1**(2), 135–151 (2007)

30. Rochat, Y., Taillard, E.: Probabilistic diversification and intensification in local search for vehicle routing. *J. Heuristics* **1**, 147–167 (1995)
31. Solomon, M.M.: Algorithms for vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **35**, 254–265 (1987)
32. Yu, B., Yang, Z.Z., Yao, B.: An improved ant colony optimization for vehicle routing problem. *Eur. J. Oper. Res.* **196**, 171–176 (2009)
33. Yuichi, N., Olli, B.: A powerful route minimization heuristic for the vehicle routing problem with time windows. *Oper. Res. Lett.* **37**, 333–338 (2009)
34. Zhang, X., Tang, L.: A new hybrid ant colony optimization algorithm for the vehicle routing problem. *Pattern Recogn. Lett.* **30**, 848–855 (2009)