

Ant Colony System for Optimizing Vehicle Routing Problem with Time Windows (VRPTW)

Xuan Tan^{1,2}, Xiaolan Zhuo¹, and Jun Zhang^{1,2,*}

¹ Department of Computer Science, SUN Yat-sen University, P.R. China

² Guangdong Key Lab of Information Security
junzhang@ieee.org

Abstract. Research on the optimization of Vehicle Routing Problem with Time Windows (VRPTW) is a significant investigation area of ant colony system (ACS). This paper proposes an enhanced ACS, which embeds the sequential insertion heuristic method, to solve VRPTW. The main idea is to organize two respective ant colonies to successively achieve a multiple objective minimization. Experiments on a series of benchmark problems demonstrate the excellent performance of ACS when compared with other optimization methods.

1 Introduction

Vehicle Routing Problem with Time Windows is a well known combinatorial optimization problem related to many real-life applications, especially in the field of industry. In the past few years, great efforts have been put forward to solve this kind of problems. In 2001, Arbelaitz *et al* [1] proposed a parallelizable system based on simulated annealing to solve VRPTW. The combination of simulated annealing and parallelizable makes a fast and low cost parallel system and is able to reach 85% of the optimal solutions for the Solomon's benchmark problems [1]. In 2004, a multi-agent model based on dynamic generation coalition was proposed, which shows an encouraging performance [2]. Besides, A. Lim *et al* focused their research on an even more restricted VRPTW, with the number of vehicles being constrained. They design an improved Greedy Randomized Adaptive Search Procedure (GRASP) framework. This framework, which embeds a technique of a smoothed dynamic tabu search, proves to be an effective method that outperforms many published algorithms in accuracy [3]. However, the first paper to introduce ACS to the solving of VRPTW appeared earlier than all the above work. In 1999, L. M. Gambardella *et al* presented a multiple ant colony system called MACS-VRPTW [4]. They reduce the complex VRPTW problem to the optimization of a multiple objective function. MACS-VRPTW organizes a hierarchy of artificial ant colonies, each of which uses independent pheromone trails but collaborate each other by exchanging information. Inspired by the idea of MACS-VRPTW, this paper presents an enhanced ACS model, which makes use of two ant colonies with respective goals. The sequential insertion heuristic method is also added

* Corresponding author: This work was supported in part by NSF of China Project No.60573066; NSF of Guangdong Project No. 5003346 and Guangdong Key Lab of Information Security.

to improve its performance. In addition, we use the Nearest Neighbor Heuristic (NNH) to generate the initial solution so that ants can start from a favorable beginning. Experimental results show that this ACS outperforms many published methods in solving the benchmark problems.

2 Ant Colony System for VRPTW

Like the neural network [5][6], ACS is a heuristic method derived from natural behaviors. The main idea of ACS is to have a set of ants search in parallel for solutions and cooperate through pheromone-mediated indirect and global communication [7]. This section gives the detailed descriptions of the proposed ACS for VRPTW.

2.1 Definition of Vehicle Routing Problem with Time Windows (VRPTW)

VRPTW represents an important component of many distribution and transportation system services. Accurately, VRPTW refers to a class of problems in which a set of routes for a fleet of vehicles based at a depot must be determined for n geographically dispersed customers. Let $C=\{1, \dots, n\}$ be a set of customers and 0 stands for the depot. Each customer c_i involves a service time S_i and asks for a quantity q_i of goods ($i=1, \dots, n$). Moreover, there is a time window $[b_i, e_i]$ during which c_i has to be served (with b_0 be the earliest starting time and e_0 the latest returning time of each vehicle). $V=\{1, \dots, t\}$ is a set of vehicles, each of which is associated with a capacity constraint Q . In addition, all the tours should begin at the depot and return to the depot.

2.2 Development of the ACS Model for VRPTW

The major goal of this elaborated ACS model is related to two objectives. We use two ant colonies to achieve this multiple objective minimization [4]. The first ant colony named ACS_vehicle is to minimize the vehicle number. The other one called ACS_time is to optimize the solution so that less traveling time is required.

2.3 Ants Search for Solutions in ACS_Vehicle

The ACS_vehicle colony, whose searching process is given in fig.1 (a), seeks for solutions by minimizing the vehicle number. It begins with an initial solution (possibly unfeasible), using one vehicle less than what the Nearest Neighbor Heuristic (NNH) requires. At the beginning, all ants set out from the depot. For ant k , it chooses the next customer according to the state transition rule (1).

$$\begin{cases} j = \arg \max_{j \in \text{fea}[k]} ([\tau_{ij}] [\eta_{ij}]^\beta), & \text{if } q \leq q_0 \\ p_{ij}^k = \frac{[\tau_{ij}] [\eta_{ij}]^\beta}{\sum_{t \in \text{fea}[k]} [\tau_{it}] [\eta_{it}]^\beta}, & j \in \text{fea}[k], \text{ otherwise} \end{cases} \quad (1)$$

τ_{ij} is the amount of pheromone on edge (i, j) and η_{ij} is defined by equation (2). In equation (2), cur_time refers to when the current vehicle will finish serving customer i . $\max(\text{cur_time} + t_{ij}, b_j)$ is the time to serve j and thus $(\max(\text{cur_time} + t_{ij}, b_j) - \text{cur_time})$

calculates the time span from now to when j begins to be served. ($e_j\text{-cur_time}$) is the measurement of the urgency of j to be served, indicating that those whose time window will expire soon will have the priority to be served first. IN_j is a variable to record the number of times that j fails to be inserted in each iteration [4]. The set $fea[k]$ records the current available customers.

$$\eta_{ij} = 1.0 / ((\max(cur_time + t_{ij}, b_j) - cur_time) * (e_j\text{-cur_time}) - IN_j) \quad (2)$$

Once the current vehicle can not serve more customers, ant k starts a new route by adding another vehicle, unless the allowed vehicle number has been reached. By repeatedly applying (1), ant k constructs a solution Ψ_k . Considering the vehicle number constraint and other restrictions, it is possible that not all customers are included in Ψ_k . The sequential insertion heuristic method is then added to help serve as more customers as possible [8]. If Ψ_k is a feasible solution, the allowed vehicle number will decrease.

2.4 Pheromone Updating Rules in ACS_Vehicle

The Local Updating Rule. This rule (3) is used to update the pheromone on the edge which has just been selected by an ant. τ_0 is the initial pheromone amount.

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad (0 < \rho < 1) \quad (3)$$

The Global Updating Rule. This rule (4) is applied twice: one for the solution with the maximum goods and the other for the global best feasible solution [4].

$$\tau_{ij} \leftarrow (1 - \alpha) \cdot \tau_{ij} + \alpha \cdot \Delta\tau_{ij} \quad (4)$$

For the solution with the most goods, $\Delta\tau_{ij}$ is calculated by (5) where L_1 is the sum of distance of all the tours in the solution with the most goods.

$$\Delta\tau_{ij} = \begin{cases} (L_1)^{-1} & \text{if } (i, j) \in \text{the_most_goods_solution} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Likewise, for the global best feasible solution, $\Delta\tau_{ij}$ is calculated by (6) where L_{gb} is the sum of distance of this solution.

$$\Delta\tau_{ij} = \begin{cases} (L_{gb})^{-1} & \text{if } (i, j) \in \text{global_best_solution} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

2.5 Ants Search for Solutions in ACS_Time

The colony of ACS_time is designed in the purpose of minimizing the total transportation time. This process is described in fig.1 (b). Considering that the length of the distance in a solution is proportional to that of the transportation time, we take the total distance of a solution to evaluate its quality. In this way, the searching process of

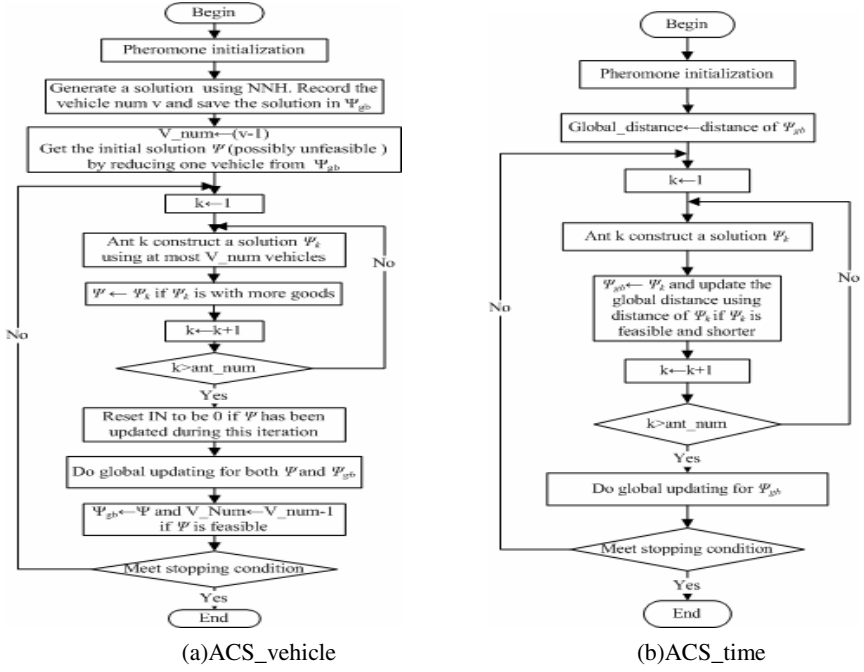


Fig. 1. Flow charts of the searching process for ACS_vehicle and ACS_time respectively

ACS_time is quite similar to that of the ant colony for solving the traveling salesman problem [7]. Therefore, ACS_time applies the same state transition rule and pheromone updating rules as those of the conventional ACS [7].

3 The Numerical Result of ACS for VRPTW

The proposed ACS for VRPTW has been tested on a set of benchmark problems, which are divided into three groups as C , R and RC . Group C consists of clustered customers whose time windows are generated based on a known solution. Group R have customers' location generated uniformly randomly over square. Group RC are combinations of randomly placed and clustered customers. The difference of these three types is the width of time windows for each customer.

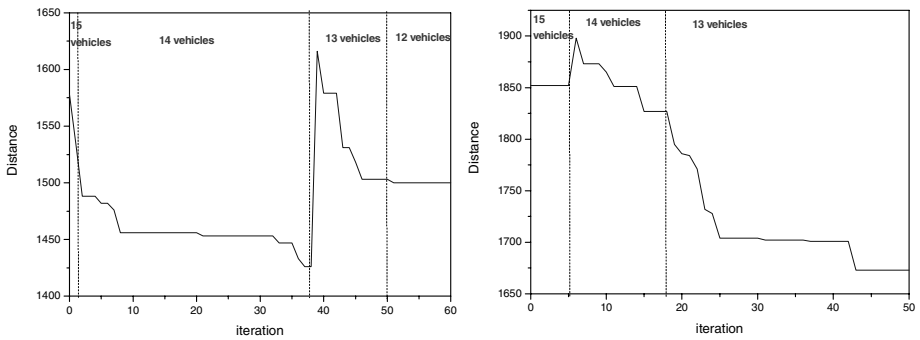
3.1 Results on Benchmark Problems

Table.1 shows some satisfying experimental results. Each test is performed 10 times. We can see that for R103, R104 and RC102, ACS requires less vehicles than all the listed algorithms while for C101, C107, R106 and RC101, it can obtain solutions almost close to that of the other algorithms.

Fig.2 shows the computing process in the respective tests for R103 and RC102, from which we can see the process of the total distance of the optimal solution against the iteration count. As the iteration increases, the vehicle number decreases. Moreover, during those iterations with the same vehicle number, the total distance decreases.

Table 1. The experimental results of ACS for some benchmark problems

Problem	Optimal Solution by other algorithms			Optimal Solution by ACS	
	Number	Distance	Author	Number	Distance
C101	10	828.94	RT [9]	10	833
C107	10	827.3	KDMSS [10]	10	837
R103	13	1292.68	LLH[11]	12	1500
R104	11	971.5	IV[12]	10	1217
R106	13	1234.6	CR[13]+KLM [14]	13	1259
RC101	15	1619.8	KDMSS [10]	15	1623
RC102	14	1457.4	CR[13]+KLM [14]	13	1673

**Fig. 2.** Schematic illustrations of the computing process for R103 and RC102 respectively

4 Conclusion

In this paper, an enhanced ACS is presented to solve the VRPTW. This ACS model makes use of two respective ant colonies which are designed to obtain respective goals. With the initial solution generated by the Nearest Neighbor Heuristic, ACS is able to start from a favorable beginning. In addition, the sequential insertion heuristic method is embedded as an auxiliary skill. An extensive computational study on a set of benchmark problems has been conducted and the experimental results show that the proposed ACS is competitive when compared with other methods.

References

1. Arbelaitz, O., Rodriguez, C., Zamakola, I.: Low Cost Parallel Solutions for the VRPTW Optimization Problem. *Parallel Processing Workshops (2001)* 176 – 181
2. Boudali, I., Fki, W., Ghedira, K.: How to Deal with the VRPTW by Using Multi-Agent Coalitions. *Hybrid Intelligent Systems. Proceedings (2004)* 416 – 421
3. Lim, A., Wang, F.: A Smoothed Dynamic Tabu Search Embedded GRASP for m-VRPTW. *Tools with Artificial Intelligence. ICTAI 2004.(2004)* 704 –708

4. 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* (1999) 63–76
5. Huang, D.S.: A Constructive Approach for Finding Arbitrary Roots of Polynomials by Neural Networks. *IEEE Transactions on Neural Networks*. 15(2004) 477–491
6. Huang, D.S., Horace, H.K., Chi, .Z.: Zeroing Polynomials Using Modified Constrained Neural Network Approach. *IEEE Trans On Neural Networks*, 16(2005) 721–732
7. Dorigo, M, Gambardella, L M: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Trans on evolutionary computation*. 1(1997) 53–66
8. Solomon, M.: Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. *Operations Research*. 35 (1987) 254–265
9. Rochat, Y., Taillard, E.D.: Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics*. 1 (1995) 147–167
10. Kohl, N., Desrosiers, J., Madsen, O. B. G., Solomon, M. M., Soumis, F.: 2-Path Cuts for the Vehicle Routing Problem with Time Windows, *Transportation Science*. (1999) 101–116
11. Li, H., Lim, A., Huang, J.: Local Search with Annealing-like Restarts to Solve the VRPTW. Working Paper, Department of Computer Science, National University of Singapore (2001)
12. Irnich, S., Villeneuve, D.: The Shortest Path Problem with K-cycle Elimination ($k \geq 3$): Improving a Branch-and-Price Algorithm for the VRPTW. *INFORMS Journal of Computing*.
13. Cook, W., Rich, J. L.: A Parallel Cutting Plane Algorithm for the VRPTW. Working Paper, Computational and Applied Mathematics, Rice University, Houston, TX (1999)
14. Kallehauge, B., Larsen, J., Madsen, O.B.G.: Lagrangean Duality and Non-Differentiable Optimization Applied on Routing with Time Windows - Experimental Results. Internal report IMM-REP-2000-8, Technical University of Denmark, Lyngby, Denmark (2000)