An Ant Colony System for the Open Vehicle Routing Problem

Xiangyong Li and Peng Tian

Antai College of Economics & Management
Shanghai Jiaotong University, Shanghai, P.R. China
lixiangyong@163.com, ptian@sjtu.edu.cn

Abstract. This paper studies the open vehicle routing problem (OVRP), in which the vehicles do not return to the starting depot after serving the last customers or, if they do, they must make the same trip in the reverse order. We present an ant colony system hybridized with local search for solving the OVRP (ACS-OVRP). Additionally, a Post-Optimization procedure is incorporated in the proposed algorithm to further improve the best-found solutions. The computational results of ACS-OVRP compared to those of other algorithms are reported, which indicate that the ACS-OVRP is another efficient algorithm for solving the OVRP.

1 Introduction

The open vehicle routing problem (OVRP) is a special variant of the standard vehicle routing problem (VRP). The most important feature consists in that the route of VRP is *hamiltonian cycle*, whereas that of OVRP is *hamiltonian path*. Such a different can be attributed to that the vehicles in the OVRP are not required to return to the depot, or if they are required to do so, they must return exactly along the same trip in the reverse order. The OVRP is a basic distribution management problem that can be used to model many real-life problems. It can be encountered in many real-word problems, for example, a delivery company without its vehicle fleet contracts its delivery to the hired vehicles. In such instance, the delivery company is not concerned with whether the vehicles return the depot and does not pay any traveling cost between the last delivery customer and the depot. It can be modeled as a OVRP. Other applications fitting the OVRP framework include the newspaper home delivery problem[1] etc..

The earliest publication about the OVRP can trace back to the article by Schrage in 1981[2]. But in the following 20 years since 1981, OVRP received little study and there is no related publication. Since 2000 some researchers began to study the solutions and several methods have been developed. Sariklis and Powell[3] proposed a two-stage heuristic, i.e. clustering and routing phases. In the clustering phase, the customers were first assigned to the clusters by taking into account vehicle capacity constraints and then the clusters were improved by reassignments of customers among these clusters. In the second phase, the clusters are transformed into an open vehicle route by solving a minimum spanning tree

M. Dorigo et al. (Eds.): ANTS 2006, LNCS 4150, pp. 356–363, 2006.

problem (MSTP). A tabu search algorithm was presented in[4]. Brandão generated initial solutions by using the nearest neighbor heuristic and the heuristic based on a pseudo lower bound. The initial solution was then submitted to either the nearest neighbor method or the unstringing and stringing procedure to improve the solution cost. The neighborhood structure was defined based on insert and swap operators. Fu et al.[5] built another tabu search. The initial solution was generated by a farthest first heuristic and the neighborhood structure was defined on four different neighborhood moves, i.e., vertex reassignment, vertex swap, 2-Opt, and trail swap. Tarantilis et al.[6] developed an adaptive memory programming algorithm for the OVRP. The set of the OVRP solutions was stored in an adaptive memory that was dynamically updated during the search process. The sequences of vertices of these solutions were periodically extracted from the adaptive memory, giving a larger weight to the routes belonging to the best solution. The algorithm had two phases. In pool generation phase, the initial pool of routes was generated using the weighted savings. The solutions were then improved using a standard tabu search. In pool exploitation phase, promising sequences of vertices of the solution are extracted, a solution was generated and improved using tabu search, and the set of solutions was updated. In[7], a list based threshold accepting (LBTA) was presented for solving the OVRP. It was an annealing-based method using a list of threshold values to guide intelligently local search. Local search are performing by using 2-opt, 1-1 exchanges (swap two customers from either same or different routes), and 1-0 exchanges (move a customer from its position on one route to another position on the same route or a different route). Recently Li et al.[1] develops a variant of the record-to-record travel algorithm developed to handle the very large standard VRP[8] to solve the OVRP. The record-to-record travel was a deterministic variant of simulated annealing. For the detailed description, we refer the reader to Li et al.[8,1].

Because of the intrinsic complexity, it is impractical to find an optimum for many combinatorial optimization problems, e.g. OVRP in a moderate computation cost. A reasonable choice is to apply metaheuristic to quickly produce good solutions. One of the most successful metaheuristics is Ant Colony Optimization (ACO), which was a common framework for the existing applications and algorithmic variants of ant algorithms. Ant Colony System (ACS), a particular instance of ACO, has proved to be competitive compared with other metaheuristics. In this paper, we apply ACS to OVRP and propose an ACS hybridized with local search and a post-optimization procedure for solving the OVRP (ACS-OVRP). The rest of this paper is organized as follows. Section 2 describes ACS-OVRP in detail. The experimental results are reported in section 3. Section 4 concludes this paper.

2 Ant Colony System for the OVRP

2.1 Problem Definition

The OVRP is a relaxation of the classical VRP. From a point of view of graph, the OVRP can be defined by a complete weighted graph G=(V, E), where

$V = \{0, 1, \cdots, n\}$ is the node set, and $E = \{(i, j)|i, j \in V, i \neq j\}$ is the edge set. Node 0 is the depot, and $C = \{1, 2, \cdots, n\}$ denotes customer set. n is the number of customers. Each arc $(i,\ j)$ is associated with a traveling distances d_{ij}. Each customer i, has a fixed demand q_i, to be delivered and a service time δ_i. Each vehicle serves a subset of customers on its route, which begins at the depot and ends at the last customer. Each vehicle has the same capacity Q and the traveling cost constraint L. The objective of the OVRP is to determine a set of minimum routes with minimum total traveling cost, which satisfy the constraints: (i) Each vehicle starts at the depot. It doesn't return to the depot, or if it do so, it must go back to the depot along the same trip in the opposite order; (ii) The service of each customer can only be fulfilled by one vehicle; (iii) The total demand of each route can not exceed the vehicle capacity; (iv) The total traveling cost of each route can not exceed the restriction L. The OVRP has multiple objectives, i.e. minimizing not only the number of vehicles required, but also the corresponding total traveled distanced. In general, a hierarchical objective is considered in the literatures, the number of routes is the primary objective and then the total travel distance is minimized for the obtained number of vehicle routes.

2.2 Ant Colony System for Solving the OVRP

Ant Colony System (ACS) is first proposed by Dorigo and Gambardella[9] and has proved to be one of the most promising ACO metaheuristics. In this section, we propose an algorithm based on ACS for OVRP. It works as follows. At each iteration, first a set of artificial ants probabilistically build the solutions, exploiting a given pheromone model. Then the generated solution can be improved by applying local search. After each ant builds its solution, the best-so-far solution is used to update the pheromone trail. Unlike AS, a local pheromone update procedure is included in ACS. The main procedures are iterated until termination condition is met. Additionally, a post optimization procedure is implemented to further improve the obtained optima. The main procedures are as follows.

Solution Construction. In ACS, m ants concurrently build the solutions of the OVRP by exploiting a probability model indicated by pheromone trail and heuristic information. Each ant starts from the node 0 (the depot) and probabilistically choose the next node until all customer nodes have been visited. When ant k is located at customer node i, it probabilistically chooses next city j to visit in the set of feasible nodes by using a pseudorandom proportional rule[9]:

$$j = \begin{cases} \arg\max_{l \in N_i^k} \{\tau_{il}^{\alpha} \eta_{il}^{\beta}\}, & \text{if } q \leq q_0 \\ J, & \text{if } q > q_0 \end{cases} \tag{1}$$

J is the customer node determined over the following probability distribution:

$$p_{ij}^k = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\Sigma_{l \in N_i^k} \tau_{il}^{\alpha} \eta_{il}^{\beta}}, \qquad \text{if } j \in N_i^k \tag{2}$$

where N_i^k is the set of all feasible nodes j still to be visited. τ_{ij} is the pheromone trail indicating the desirability of visiting customer j directly after customer i. η_{ij} is the heuristic information and it shows the heuristic desirability of choosing customer j as the next city. In the context of OVRP, we consider $\eta_{ij} = 1/d_{ij}$ as in TSP, where d_{ij} is the travel distance between node i and j. α and β are two parameters indicating the relative importance of the pheromone trail and the heuristic information. q is a random number uniformly distributed in the interval $[0, 1]$; q_0 is a control parameter and $0 \leq q_0 \leq 1$. From equation (1), the node j maximizing the product $\tau_{il}^{\alpha}\eta_{il}^{\beta}$ will be chosen as the next city with probability q_0. While with probability $1-q_0$, the city j is chosen with the probability defined in equation (2). q_0 is an important parameter, which controls the tradeoff between the exploration and exploitation[9].

Local Search. The vast literatures indicate that ACO hybridized with a local search can obtain better solution. In this section, we consider some local search algorithm embedded in the ACS-OVRP. Our implementation of local search oscillates between the inter-route as well as intra-route improvements. For the intra-improvement, we use a method similar to 2-Opt for each vehicle route of the OVRP. When implementing 2-Opt to each vehicle route, the termination condition used is the best-accept (BA) strategy, i.e. all the neighbors are examined until no improvement can be obtained. In order to get an inter-route improvement, an exchange similar to 2-Opt*[10] is used. But it is modified in consideration of the property of OVRP, namely the route is not a tour. Another local search is swap operator, which performs by exchanging the customer in one route with other customer in another route. In the implementation for the OVRP, one customer or two adjacent customers are considered for swapping. We also use another local search, relocate operator. The basic idea is to eject a small sequence of customers (here we consider one customer) at the current location and try to improve the solution by reinserting the sequence at another location. Relocate operator is suitable for inter-route and intra-route improvements.

Except for the 2-Opt, other operators are implemented with FA criteria, i.e. they will stop if the first improvement is obtained. Local search is implemented in a random order. After the ants finish solution construction, we first produce a random permutation of integers 1-4 and then the local search is implemented in turn according to the permutation. Here we gradually increase the number of ants applying local search, to obtain a tradeoff between solution quality and computation time. ACS-OVRP first allows only the iteration-best ant to apply local search. Then the number *Elitist_num* of ants applying local search is increased by one every *num_iter* iterations and a upper bound is set to 10. When triggering local search, the ants are first ranked in terms of travel cost, and then only the *Elitist_num* best-ranked ants are chosen to apply local search.

Pheromone Trail Update. In the ACS-OVRP, the pheromone trail is updated by two updating procedures, local updating and global updating.

The local pheromone trail update rule is triggered immediately after the ant chooses a next city. The updating equation is as follows:

$$\tau_{ij} = (1 - a) \cdot \tau_{ij} + a \cdot \tau_0 \tag{3}$$

where a $(0 < a \leq 1)$ is the evaporation rate. τ_0 is the initial pheromone trail and a good choice is to set $\tau_0 = 1/(n \cdot L_f)$. L_f is the length of initial solution generated by the nearest neighbor heuristic, and n is the number of customers. The effect of local updating is to reduce the pheromone trail on the visited arcs at current iteration, making these arcs less attractive for the following ants.

After all ants finish constructing their solutions the best-so-far ant is then allowed to deposit additional pheromone trail. The basic idea is that the information of the best-so-far solution is indicated by the pheromone trail and the arcs included in the best-so-far solution will be biased by other ants in the following iterations. The updating equation is as follows:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho / L_{bs} \tag{4}$$

where L_{bs} is the length of the best found solution. $0 < \rho \leq 1$ is pheromone evaporation rate. In this updating equation, both pheromone evaporation and new pheromone deposit are only applied to the arc (i, j) included in the best solution. In consideration of property of OVRP, pheromone trail on the arc connecting the last customer and the depot, i.e. the arc $(i, 0), i \in C$, will not be updated.

Post-Optimization. In our ACS-OVRP, we consider a post-optimization procedure to further improve the best found solutions. When the termination condition of the main loop is met, the post-optimization procedure is performed to further improve the solution. Our Post-Optimization procedure consists of the above four operators mentioned in the subsection *"Local search"*. The operators are implemented in different orders to obtain the best possible improvement.

3 Experiment and Results

3.1 Experiment Setting

In this section, ACS-OVRP was tested on a set of problems. The problems C1-C14 were taken from Christofides et al.[11] and the data can be downloaded from the web (http://people.brunel.ac.uk/~mastjjb/jeb/info.html). The properties of the problems are summarized in Table 1. L is the maximum route length restriction which is the original route length restriction multiplied by 0.9 because the original problems were designed for VRP, not for OVRP[4,5]. For the vehicle without driving time limit, we set $L = \infty$. n is the number of customers. Q is the vehicle capacity. K_{min} is the estimated minimum number of route by $\lceil \Sigma_{i=1}^{n} d_i / Q \rceil$. δ is service time. All the experiments were implemented on a Pentium IV 1.8GHz with Matlab 7.0 simulation platform.

In the preliminary experiment, we checked the performance of ACS-OVRP under different parameters and found that ACS-OVRP performed better under the following parameters: $\rho = a = 0.1$, number of ants, 20; the size of candidate list, $\lceil n/5 \rceil$; the maximum iterations, 300. $num\_iter = 50$, $\alpha = 1$, and $\beta = 2$.

Table 1. Characteristics of the testing problems

No	n	Q	L	δ	K_{min}	No	n	Q	L	δ	K_{min}
C1	50	160	∞	0	5	C8	100	200	207	10	8
C2	75	140	∞	0	10	C9	150	200	180	10	12
C3	100	200	∞	0	8	C10	199	200	180	10	16
C4	150	200	∞	0	12	C11	120	200	∞	0	7
C5	199	200	∞	0	16	C12	100	200	∞	0	10
C6	50	160	180	10	5	C13	120	200	648	50	7
C7	75	140	144	10	10	C14	100	200	936	90	10

3.2 Computational Results and Comparison with Other Algorithms

We tested the performance of ACS-OVRP and compared it with other methods in the literatures. The algorithms compared with ACS-OVRP are as follows. SP: the method of cluster first, route second in[3]. TSB1 and TSB2: tabu search algorithm in[4]. TSB1 and TSB2 denote the tabu search algorithms with different initial solutions, namely the initial solutions produced by K-tree with unstringing and stringing, and by nearest neighbor heuristic. AMP: adaptive memory programming algorithm (AMP)[6]. BATA: the backtracking adaptive threshold accepting algorithm[12]. LBTA: the list based threshold accepting (LBTA) algorithm[7]. TSF and TSR: tabu search algorithm with different initial solutions[5]. In TSF, the initial solutions are produced by a farthest first heuristic. The ones of TSR are randomly generated. We note that some results are not correct in[5] because of some errors in program. We got the corrected results and reported them here. ORTR: the record-to-record travel algorithm[1]. All methods consider the number of routes as the primary objective. As in Brandão[4], the reported solutions contain only the traveling cost without service time.

We first studied the effect of post-optimization procedure. The results are reported in Table 2. It can be seen that the effect of the post-optimization is slightly significant and the obtained improvement is quite obvious, especially for large-scale testing problem. In particular, the solution quality of problem C1 and C2 can not be further improved. It is probably due to the fact that ACS-OVRP has found the best solutions for these problems. But it is an interesting observation that the solution quality of big problem, e.g. C4 and C5, has been improved by using post-optimization procedure. The results show that it is valuable to incorporate post-optimization procedure to further improve the solution quality.

Table 2. Summary of the effects of Post-Optimization

	C1	C2	C3	C4	C5	C6	C7
Post-Opt	5/416.1	10/571.7	8/649.0	12/748.4	16/1017.3	6/413.0	11/568.5
No Post-Opt	5/416.1	10/571.7	8/654.9	12/917.3	16/1083.2	6/413.1	11/568.5
	C8	C9	C10	C11	C12	C13	C14
Post-Opt	9/647.9	14/764.2	17/903.1	7/685.3	10/536.3	11/903.8	11/593.1
No Post-Opt	9/653.7	14/770.5	17/936.4	7/692.0	10/538.3	12/924.4	11/597.6

Table 3. Comparison of the optimums produced by ACS-OVRP and other algorithms

Problem	SP	TSB1	TSB2	AMP	BATA
C1	5/488.20	5/416.1	5/438.2	6/412.96	6/412.96
C2	10/795.33	10/574.5	10/584.7	11/564.06	11/564.06
C3	8/815.04	8/641.6	8/643.4	9/641.77	8/642.42
C4	12/1034.14	12/740.8	12/767.4	12/735.47	12/736.89
C5	16/1349.71	16/953.4	16/1010.9	17/877.13	**16/879.37**
C6		**6/412.96**	6/416.00		
C7		**10/634.54**	11/580.97		
C8		**9/644.63**	9/652.09		
C9		**13/785.2**	14/827.6		
C10		17/884.63	17/946.8		
C11	7/828.25	7/683.4	7/713.3	10/679.38	9/679.60
C12	10/882.27	10/535.1	10/543.2	**10/534.24**	**10/534.24**
C13		**11/943.66**	11/994.26		
C14		11/597.3	12/651.92		
Problem	LBTA	TSF	TSR	ORTR	ACS-OVRP
C1	6/412.96	5/416.1	**5/416.06**	**5/416.06**	**5/416.06**
C2	11/564.06	10/569.8	**10/567.14**	**10/567.14**	10/571.70
C3	9/639.57	8/641.9	8/643.05	**8/639.74**	8/649.02
C4	12/733.68	12/742.4	12/738.94	**12/733.13**	12/748.40
C5	17/870.26	17/879.9	17/878.95	16/924.96	16/1017.28
C6		6/413.0	**6/412.96**	**6/412.96**	**6/412.96**
C7		11/568.5	11/568.49	11/568.49	11/568.49
C8		9/648.0	9/647.26	**9/644.63**	9/647.94
C9		14/767.1	14/761.28	**14/756.38**	14/764.15
C10		17/904.1	17/903.10	**17/876.02**	17/903.10
C11	10/678.54	7/717.2	7/724.46	**7/682.54**	7/685.32
C12	**10/534.24**	10/537.8	10/534.71	**10/534.24**	10/536.33
C13		12/917.9	12/922.28	12/896.50	12/903.82
C14		11/660.2	11/600.66	**11/591.87**	11/593.08

In Table 3, we compared the best found solutions of ACS-OVRP with those of other algorithms. The best solution corresponding to each problem among all the algorithms is marked in bold face. The results show that:

(a) No algorithm can dominate other algorithms in terms of the number of the vehicles and the corresponding traveling cost over all the problems. But ORTR performs well over most of the problems.

(b) ACS-OVRP outperforms some algorithms in some problems. For example, for C2 ACS-OVRP found better solutions with less vehicle than AMP, BATA and LBTA. Similar results can be observed on C3 and C5. Especially for problem C11, the best solutions found by AMP and LBTA respectively have 3 and 2 vehicles more than the one by ACS-OVRP. It can also be seen that our proposed algorithm outperforms TSR and TSF on some problems e.g. C5, C11, C13 and C14.

(c) ACS-OVRP appears to be very competitive with other algorithms on many problems. But for some large problem, for example, C5 and C10, ORTR outperforms our approach. The best found solution of ACS-OVRP has the same fleet size as that of ORTR, but higher traveling cost. It may be due to that ORTR is specially designed for large-scale VRPs and OVRPs.

4 Conclusions

In this paper, we propose a metaheuristic coupled ACS with local search for the OVRP. Moreover, a Post-Optimization procedure is introduced to further improve the solutions. The effects of Post-Optimization procedure were proved by the computation results. Finally, we compare the performance of ACS-OVRP with those of other algorithms. The comparison results show that ACS-OVRP is an efficient algorithm for solving the OVRP. During the implementation, we found that the computation cost of local search accounted for a large proportion of total time consumption because of their intrinsic computation complexity. How to speed up the implementation of local search in ACS-OVRP and other efficient local search need to be studied in the future work.

References

1. Li, F., Golden, B., Wasil, E.: The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. Computers and Operations Research (In press)
2. Schrage, L.: Formulation and structure of more complex realistic routing and scheduling problem. Networks **11** (1981) 229–232
3. Sariklis, D., Powell, S.: A heuristic method for the open vehicle routing problem. The Journal of the Operational Research Society **51**(5) (2000) 564–573
4. Brãndao, J.: A tabu search algorithm for the open vehicle routing problem. European Journal of Operational Research **157**(3) (2004) 552–564
5. Fu, Z., Eglese, R., Li, L.: A new tabu search heuristic for the open vehicle routing problem. Journal of the Opeartional Research Society **56**(2) (2005) 267–274
6. Tarantilis, C., Diakoulaki, D., Kiranoudis, C.: Combination of geographical infromation system and efficient routing algorithms for real life distribution operations. European Journal of Opeartions Research **152**(2) (2004) 437–453
7. Tarantilis, C., Ioannou, G., Kiranoudis, C., Prastacos, G.: Solving the open vehicle routing problem via a single parameter metaheuristic algorithm. Journal of the Operational Research Society **56** (2005) 588–596
8. Li, F., Golden, B., Wasil, E.: Very large-scale vehicle routing: new test problems, algorithms, and results. Computers and Operations Research **32**(5) (2005) 1165–1179
9. Dorigo, M., Stützle, T.: Ant Colony Optimization. The MIT Press, Massachusetts (2004)
10. Potvin, P., Rousseau, J.: An exchange heuristic for routing problem with time windows. Journal of the Operational Research Society **46**(1433-1446) (1995)
11. Christofides, N., Mingozzi, A., Toth, P.: The vehicle routing problem. In Christofides, N., Mingozzi, A., Toth, P., Sandi, C., eds.: Combinatorial optimization. Chichester, UK: Wiley (1979) 315–338
12. Tarantilis, C., Ioannou, G., Kiranoudis, C., Prastacos, G.: A threshold accepting approach for the open vehicle routing. RAIRO Opeartions Research **38** (2004) 345–360