

A Modified Ant Algorithm for Solving the Vehicle Routing Problem

Chengming Qi

College of Automation, Beijing Union University, Beijing 100101, P.R.China

Abstract

The ant system is a metaheuristic developed for the solution of hard combinatorial optimization problems. In this paper, through an analysis of the constructive procedure of the solution in the Ant Colony System (ACS), a modified algorithm for solving Vehicle Routing Problem (VRP), based on the ACS Hybridized with Randomized Algorithm (HRACS), is proposed. In HRACS, only partial customers are randomly chosen to compute the transition probability. Experiments on various aspects of the algorithm and computational results for fourteen benchmark problems are reported. We compare our approach with some other meta-heuristics and show that our results are competitive.

Keywords: Ant colony system, Combinatorial optimization, Randomized algorithm, Vehicle routing problem.

1. Introduction

The Ant System (AS), introduced by Colnani, Dorigo and Maniezzo [1–3] is a new distributed meta-heuristic for hard combinatorial optimization problems and was first used on the well known Traveling Salesman Problem (TSP). Starting from Ant System, several improvements of the basic algorithm have been proposed [4–7]. Typically, these improved algorithms have been tested again on the TSP [8]. All these improved versions of AS have in common a stronger exploitation of the best solutions found to direct the ants' search process; they mainly differ in some aspects of the search control.

One of the most efficient ACO based implementations has been Ant Colony System (ACS) [9, 10], that introduced a particular pheromone trail updating procedure useful to intensify the search in the neighborhood of the best computed solution.

The Vehicle Routing Problem (VRP) is a well-known and complex combinatorial problem,

which has received considerable attention in recent years. The vehicle routing problem has been largely studied because of the interest in its applications in logistic and supply-chains management. Bullnheimer, Hartl, and Strauss [11, 12] applied an AS-like algorithm to VRP. Gambardella, Taillard, Agazzi [13], John E. Bell et al. [14] and Silvia Mazzeo et al. [15] have also attacked the VRP by means of an ACO algorithm. They also study the vehicle routing problem with time windows (VRPTW), which extends the VRP by introducing a time window within which a customer must be served. In all, Many different versions of this problem have been formulated to take into account many possible different aspects. One of the most interesting that has been raising in recent days, is to take into account the diversified road network traffic conditions during the course of the day.

Randomized algorithm is widely used in combinatorial optimization. It has been shown to outperform their deterministic counterparts in a number of interesting application domains. Up to date, they are becoming increasingly more important and popular for solving computationally hard combinatorial problems from various domains of AI and Operations Research, such as planning, scheduling, constraint satisfaction, satisfiability and other application domains. While doing so does not improve the algorithm in the worst case, it often makes very good algorithms in the average case.

In this paper, we apply randomized algorithms to ACS. The proposed algorithm computes the transition probability of random generated partial customers only, while ACS should compute that of all customers in the constructive procedure of the solution.

The paper is organized as follows. In Section 2, vehicle routing problems are introduced by presenting a formal definition of the VRP. Meanwhile, we introduce the solution construction mechanism used by the ACO metaheuristic. Section 3 presents a modified ant algorithm which Hybridized with Randomized Algorithm (HRACS). In Section 4 we provides an experimental results and comparisons

of HRACS with ACS. We conclude in Section 5 with a brief summary of the contributions of this paper.

2. Ant colony system for the VRP

Combinatorial optimization problems Almost all ACO algorithms have initially been tested on the Traveling Salesman Problems. In this paper we apply the similar idea to the VRP.

2.1. Vehicle routing problem

The vehicle routing problem is a very complicated combinatorial optimization problem that has been worked on since the late fifties, because of its central meaning in distribution management.

The vehicle routing problem can be described as follows[16]: n customers must be served from a (unique) depot. Each customer i asks for a quantity q_i of goods. A fleet of v vehicles, each vehicle a with a capacity Q_a , is available to deliver goods. A service time s_i is associated with each customer. It represents the time required to service him/her. Therefore, a VRP solution is a collection of tours.

The VRP can be modelled in mathematical terms through a complete weighted digraph $G = (V, A)$, where $V = \{0, 1, \dots, n\}$ is a set of nodes representing the depot (0) and the customers ($1, \dots, n$), and $A = \{(i, j) \mid i, j \in V\}$ is a set of arcs, each one with a minimum travel time tt_{ij} associated. The quantity of goods q_i requested by each customer i ($i > 0$) is associated with the corresponding vertex with a label. Labels Q_1, \dots, Q_v , corresponding to vehicles capacities, are finally associated with vertex 0 (the depot). The goal is to find a feasible set of tours with the minimum total travel time. A set of tours is feasible if each node is visited exactly once (i.e. it is included into exactly one tour), each tour starts and ends at the depot (vertex 0), and the sum of the quantities associated with the vertices contained in it, never exceeds the corresponding vehicle capacity Q_a .

Table 1 contains the data for the 14 vehicle routing problem instances[16, 17]:

2.2. Applying ant colony system to the VRP

To solve the VRP, the artificial ants construct vehicle routes by successively choosing cities to visit,

<i>Instance</i>	n	Q	L	<i>best publ</i>
C1	50	160	∞	524.61
C2	75	140	∞	835.26
C3	100	200	∞	826.14
C4	150	200	∞	1028.42
C5	199	200	∞	1291.45
C6	50	160	200	555.43
C7	75	140	160	909.68
C8	100	200	230	865.94
C9	150	200	200	1162.55
C10	199	200	200	1395.85
C11	120	200	∞	1042.11
C12	100	200	∞	819.56
C13	120	200	720	1541.14
C14	100	200	1040	866.37

n ... number of customers

Q ... vehicle capacity

L ... maximum tour length

best publ. ... best published solution

Table 1: Characteristics of the benchmark problem instances (C1–C10 are Random Problems and C11–C14 are Clustered Problems)

until each city has been visited. Whenever the choice of another city would lead to an infeasible solution for reasons of vehicle capacity or total route length, the depot is chosen and a new tour is started.

At each step, every ant k computes a set of feasible expansions to its current partial solution and selects one of these probabilistically, according to a probability distribution specified as follows. For ant k the probability p_{ij}^k of visiting customer j after customer i , the last visited customer, depends on the combination of two values[16]:

- the attractiveness u_{ij} of arc (i, j) , as computed by some heuristic indicating the a priori desirability of that move. Where $u_{ij} = tt_{ij}$ (a minimum travel time associated with each arc), i.e. it depends directly on the travel time between customer i and customer j ;
- the pheromone level τ_{ij} of arc (i, j) , indicating how proficient it has been in the past to visit j after i is a solution; it represents therefore an a posteriori indication of the desirability of that move.

This heuristic uses a population of m agents which construct solutions step by step. When all the ants have constructed their tour, the best solution are rewarded so as to encourage the identification of ever better solutions in the next cycles.

The most important component of an ant system is the management of pheromone trails. In a standard ant system, pheromone trails are used in conjunction with the objective function for constructing a new solution. The information contained in the pheromone trails and the use of this information is the key element of an ant system. Informally, pheromone levels give a measure of how desirable is to insert a given element into a solution. For the VRP, we have chosen to represent the set of the pheromone trails by a matrix $T = (\tau_{ij})$ of size $n \times n$, where the entry τ_{ij} measures the desirability of setting $\pi_i = j$ in the solutions visited by the ant system.

2.2.1. Construction of vehicle routes

ACS goal is to find a shortest tour. In ACS m ants build tours in parallel, where m is a parameter. Each ant is randomly assigned to a starting node and has to build a solution, that is, a complete tour. A tour is built node by node: each ant iteratively adds new nodes until all nodes have been visited. When ant k is located in node i , it chooses the next node j probabilistically in the set of feasible nodes N_i^k (i.e., the set of nodes that still have to be visited). The probabilistic rule used to construct a tour is the following: with probability q_0 a node with the highest $[\tau_{ij}]^\alpha [\eta_{ij}]^\beta$, $j \in N_i^k$ is chosen (exploitation), while with probability $(1 - q_0)$ the node j is chosen with a probability p_{ij} proportional to $[\tau_{ij}]^\alpha [\eta_{ij}]^\beta$, $j \in N_i^k$ (exploration).

With $\Omega = \{v_j \in V : v_j \text{ is feasible to be visited}\} \cup \{v_0\}$, city v_j is selected to be visited after city v_i according to a *random-proportional rule* [6] that can be stated as follows:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h \in \Omega} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta} & \text{if } v_j \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

This probability distribution is biased by the parameters α and β that determine the relative influence of the trails and the visibility, respectively. For the TSP Dorigo et al. [19] define the visibility as the reciprocal of the distance. The same is done for the VRP in [12] where the selection probability is then further extended by problem specific information. There, the inclusion of savings and capacity utilization both lead to better results. The latter is relative costly in terms of computation time (as it has to be calculated in each step of an iteration).

2.2.2. Pheromone trail update

After an artificial ant k has constructed a feasible solution, the pheromone trails are laid depending on the objective value L_k . For each arc (v_i, v_j) that was used by ant k , the pheromone trail is increased by $\Delta\tau_{ij}^k = 1/L_k$. In addition to that, all arcs belonging to the so far best solution (objective value L^*) are emphasized as if σ ants, so-called *elitist ants* had used them. One elitist ant increases the trail intensity by an amount $\Delta\tau_{ij}^*$ that is equal to $1/L^*$ if arc (v_i, v_j) belongs to the so far best solution, and zero otherwise. Furthermore, part of the existing pheromone trails evaporates (ρ is the trail persistence)[3]. Thus, the trail intensities are updated according to the following:

$$\tau_{ij}^{new} = \rho\tau_{ij}^{old} + \sum_{k=1}^m \Delta\tau_{ij}^k + \sigma\Delta\tau_{ij}^* \quad (2)$$

where m is the number of artificial ants

Concerning the initial placement of the artificial ants it was found that the number of ants should be equal to the number of cities in the TSP, and that each ant should start its tour from another city. The implication for the VRP is that as many ants are used as there are customers in the VRP (i.e. $m = n$), and that one ant is placed at each customer at the beginning of an iteration. After initializing the basic ant system algorithm, the two steps *construction of vehicle routes* and *Pheromone trail update*, are repeated for a given number of iterations.

The update of the pheromone trail is done in a different way than those of the standard model where all the ants update the pheromone trail. Indeed, this manner of updating the pheromone trail implies a very slow convergence of the algorithm [6]. For speeding-up the convergence, we update the pheromone trail by taking into account only the best solution produced by the search to date. First, all the pheromone trails are weakened by setting $\tau_{ij} = \rho \cdot \tau_{ij}$, ($1 \leq i, j \leq n$) where $0 < \rho < 1$ is a parameter that controls the *evaporation* of the pheromone trail: a value of ρ close to 1 implies that the pheromone trails remains active a long time, while ρ value close to 0 implies a high degree of evaporation and a shorter memory of the system.

3. ACS with randomized algorithm

In this section, we present a hybrid algorithm. The approach applies randomized algorithm to ACS.

The first ant system for vehicle routing problems has been designed only very recently by Bullnheimer et al.[11, 12] who considered the most elementary version of the problem: the capacitated vehicle routing problem (CVRP). It considers a more elaborated vehicle routing problem with two objective functions: (i) the minimization of the number of tours (or vehicles) and (ii) the minimization of the total travel time, where number of tours minimization takes precedence over travel time minimization.

An interesting aspect of the local updating is that while edges are visited by ants. Eq.(2) makes the trail intensity diminish, making them less and less attractive, and favoring therefore the exploration of not yet visited edges and diversity in solution generation.

Once a complete solution is available, it is tentatively improved using a local search procedure. We used a very simple greedy algorithm, which iteratively selects a customer and tries to move it into another position within its tour or within another tour. A maximum computation time for the local search, t_{ls} , must be specified.

Once the m ants of the colony have completed their computation, the best known solution is used to globally modify the pheromone trail. In this way a "preferred route" is memorized in the pheromone trail matrix and future ants will use this information to generate new solutions in a neighborhood of this preferred route. The pheromone matrix is updated as follows:

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \frac{\rho}{CostBest} \quad \forall (i,j) \in BestSol \quad (3)$$

where $\rho(0 \leq \rho \leq 1)$ is a parameter and $CostBest$ is the total travel time of solution $BestSol$, the best tour generated by the algorithm since the beginning of the computation.

The process is iterated by starting again m ants until a termination condition is met. The termination criterion is a maximum computation time of $< t_{acs}$ seconds.

Pseudo-code of the ACS procedure for the vehicle routing problem as following[16]:

```

BestCost := ∞;
For each arc (i,j)
     $\tau_{ij} = \tau_0$ 
EndFor
While (computation time <  $t_{acs}$ )
    For  $k:=1$  to  $m$ 
        While (Ant  $k$  has not completed its solution)

```

```

        Select the next customer  $j$ , according to Eq.(1);
        Update the trail level  $\tau_{ij}^{new}$ , according to Eq.(2);
    EndWhile
    Run a local search (maximum computation time =  $t_{ls}$ );
    Cost := Cost of the current solution;
    If ( $Cost < CostBest$ )
        CostBest := Cost;
        BestSol := current solution;
    EndIf
EndFor
For each move (i,j) in solution BestSol
    Update the trace level  $\tau_{ij}^{new}$  according to Eq.(3)
EndFor
EndWhile

```

In order to decrease the times of computing Eq.(1), we introduce randomized algorithm into ACS, i.e., HRACS. In Eq.(1), p_{ij}^k is the probability that the k -th ant in node i to choose node j . In HRACS, only random generated N customers are calculated, where N is part of the number of customers n . Usually, let N is almost 20% of n . In C3, VRP instance, for example, we let $N = 20$. Thus, HRACS adopt randomized approach only to compute the transition probability of N customers (ACS computes p_{ij}^k of all customers). We modify pseudo-code of inner *While* loop as follow:

```

For  $i:=1$  to  $n-N$  do
    Generate random number  $r \leftarrow \{r_1, r_2, \dots, r_N\}$ ,
    compute  $p_{ij}^k$  of  $N$  customers according to Eq.(1)
    Update the trail level  $\tau_{ij}^{new}$ , according to Eq.(2);
End-for
For  $i:=1$  to  $N$  do
    compute  $p_{ij}^k$  of last  $N$  customers
    Update the trail level  $\tau_{ij}^{new}$ , according to Eq.(2);
End-for

```

4. Numerical results

In this section we will present numerical results for our new approach (HRACS) and compare them with results from previous Ant Colony System algorithm (ACS)[11] for the VRP as well as TABUROUTE algorithm (TS)[20] and Simulated Annealing algorithm (SA)[21].

Instance	TS	SA	ACS	HRACS
C1	524.61	528	524.61	524.61
C2	835.77	838.62	870.58	872.41
C3	829.45	829.18	879.43	865.32
C4	1036.16	1058	1147.41	1042.73
C5	1322.65	1376	1473.40	1320.26
C6	555.43	555.43	562.93	566.31
C7	913.23	909.68	948.16	944.37
C8	865.94	866.75	886.17	885.54
C9	1177.76	1164.12	1202.01	1172.66
C10	1418.51	1417.85	1504.79	1480.27
C11	1073.47	1176	1072.45	1066.12
C12	819.56	826	819.96	821.16
C13	1573.81	1545.98	1590.52	1587.77
C14	866.37	890	869.86	868.91

Table 2: Experimental results for TS,SA,ACS and HRACS on VRP

The numerical analysis was performed on a set of benchmark problems described in [22]. In table 1 the set of benchmark problems consists of 14 instances containing between 50 and 199 customers and a depot. The first ten instances were generated with the customers being randomly distributed in the plane, while instances 11-14 feature clusters of customer locations. All instances are capacity constrained.

Our purpose is to show that our ant procedure is able to find good solutions in a moderate amount of time for problems presenting a structure.

Experiments were run on a Pentium IV, 256MB of RAM, 1.7 GHz processor. In order to assess the relative performance of ACS versus HRACS independently from the details of the settings, We choose the same settings. we used n artificial ants, initially placed at the customers v_i, \dots, v_n and set $\alpha = 1, \beta = 5$ and $\rho = 0.75$. For all problems maximum iteration times are $2n$.

Each run is guaranteed to be independent of others by starting with different random seeds. The result in Table 2 and Table 3 indicate that HRACS was able to find good results for larger problem instances. HRACS is superior to ACS except for three instances (C2, C6 and C12). For the instances of C5, C9 and C11, HRACS even shows a slightly better performance than TS.

5. Conclusion

In this paper we propose a hybrid ACO approach for solving vehicle routing problems. The main idea is to combine an ant colony system with a Random-

Instance	$ACS_{dev.}$	$HRAS_{dev.}$
C1	0.00%	0.00%
C2	4.23%	4.45%
C3	6.45%	4.74%
C4	11.57%	1.39%
C5	14.09%	2.23%
C6	1.35%	1.96%
C7	4.23%	3.81%
C8	2.34%	2.26%
C9	3.39%	0.87%
C10	7.80%	6.08%
C11	2.91%	2.30%
C12	0.05%	0.20%
C13	3.20%	3.03%
C14	0.40%	0.30%

Table 3: Deviation of ACS and HRACS on VRP

ized algorithm namely HRACS. Computational results shows the viability of the HRACS approach to generate very high quality solutions for the VRP, and proved that HRACS is an interesting novel stochastic approach to optimization of the VRP, especially to larger instances. Moreover, The idea developed in this paper is generic and applicable to other heuristics of the VRP.

There are several interesting directions for future work. On the one hand, it seems important to incorporate stochastic information to decide whether to accept a customer request. On the other hand, it is worth while to exploit the idea presented in this paper to solve dynamic real-world problems. As a consequence, applying these ideas to other domains is an important avenue for future research.

References

- [1] A. Coloni, M. Dorigo and V. Maniezzo. Distributed Optimization by Ant Colonies, in: *Proc. Eearop. Conf. Artificial Life*, ed. F. Varela and P. Bourguine, Elsevier, Amsterdam, 1991.
- [2] M. Dorigo, Optimization, Learning and Natural Algorithms, *Doctoral Dissertation*, Politecnico di Milano, Italy, 1992.
- [3] M. Dorigo, V. Maniezzo and A. Coloni. Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Trans. Sys., Man, Cybernetics* 26:29-41, 1996.
- [4] L.M. Gambardella and M. Dorigo. Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. In *Proceedings of*

- the Eleventh International Conference on Machine Learning*, pp. 252-260, 1995.
- [5] L.M. Gambardella and M. Darigo. Solving Symmetric and Asymmetric TSPs by Ant Colonies. *In Proceedings of the IEEE International Conference on Evolutionary Computation*, pp.622-627, 1996.
 - [6] M. Dorigo and L.M. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1:53-66, 1997.
 - [7] T. Stützle and H.H. Hoos. The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp.309-314,1997.
 - [8] M. Dorigo, L. M. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation*, pp.53-66,1997.
 - [9] L. M. Gambardella, M. Dorigo. Solving Symmetric and Asymmetric TSPs by Ant Colonies, *Proceedings of the IEEE Conference on Evolutionary Computation*, pp.622-627,1996.
 - [10] M. Dorigo, L. M. Gambardella. Ant Colonies for the Traveling Salesman Problem, *BioSystems* 43:73- 81, 1997.
 - [11] B.Bullnheimer, R. F.Hartl,& C.Strauss. An improved Ant System algorithm for the vehicle routing problem. (*Tech. Rep. POM-10/97*). Vienna, Austria: University of Vienna, Institute of Management Science, 1997.
 - [12] B.Bullnheimer, R. F.Hartl, & Strauss, C. Applying the Ant System to the vehicle routing problem. *Meta-heuristics: Advances and trends in local search paradigms for optimization*, pp. 109-120,1998.
 - [13] L.M. Gambardella, É. Taillard, and G. Agazzi. MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. *In New ideas in optimization*. pp.63-76, 1999.
 - [14] John E. Bell and Patrick R. McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*,18:41-48, 2004.
 - [15] Silvia Mazzeo and Irene Loiseau. An Ant Colony Algorithm for the Capacitated Vehicle Routing. *Electronic Notes in Discrete Mathematics*, 18:181-186, 2004.
 - [16] R. Montemanni, L. Gambardella, A. Rizzoli, and A. Donati. A new algorithm for a dynamic vehicle routing problem based on ant colony system. *In Second International Workshop on Freight Transportation and Logistics*, 2003.
 - [17] E. D.Taillard, Parallel iterative search methods for vehicle routing problems. *Networks* ,23:661-673,1993.
 - [18] Y.Rochat,and E. D.Taillard: Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics* 1:147-167, 1995.
 - [19] M. Dorigo, V. Maniezzo and A. Colorni. Ant System: Optimization by a Colony of Cooperating Agents,*IEEE Trans. Sys., Man, Cybernetics* 26,1996.
 - [20] Gendreau, M., Hertz, A. and Laporte, G.:A tabu search heuristic for the vehicle routing problem.*Management Science* 40:1276-1290,1994.
 - [21] I. H.Osman,:Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem.*Annals of Operations Research* 41:421-451,1993.
 - [22] N.Christofides,A.Mingozzi and P.Toth: The vehicle routing problem. *Combinatorial Optimization*. Wiley, Chicester,1979.