Meta-heuristics for vehicle routing and inventory routing problems

Yang, Z.

Author: Zhiwei Yang
Title: Meta-heuristics for vehicle routing and inventory routing problems
Issue Date: 2016-09-20

3

Ant Colony Optimization

In nature, ants stumble around randomly in order to find their food. If an ant finds food, it returns to the nest and deposits pheromone trails on the ground. The other ants follow the pheromone trails to get the food source. The higher amount of pheromone on the path, the higher probability of ants selecting it. If they succeeded to find the food source, more pheromone would be laid down on this path to the food source. The pheromone trails evaporates over time. If no food exists there, the amount of pheromones on the path would decrease gradually to a very low level and few ants would go there again. Since the pheromone on the shortest path has the highest amount of pheromones, all ants will find the shortest way to the food source.

The ant colony optimization (ACO) algorithm is inspired by the observation of the foraging behavior of ants in nature which is first proposed by Dorigo in his PhD thesis [Dorigo, 1992]. Since then, many researchers have been working on improving this algorithm which makes this algorithm able to compete with other metaheuristic algorithms in solving combinatorial problems both in theoretical and practical fields [López-Ibánez et al., 2015]. Many successful applications can be found such as scheduling problems in [Xiao et al., 2013, Chen and Zhang, 2013], routing problems in [Balaprakash et al., 2009, Toth and Vigo, 2014], assignment problems in [Gambardella et al., 1999b, D'Acierno et al., 2012], set problems in [Ren et al., 2010, Jovanovic and Tuba, 2013] and so on. In this chapter, we will introduce the canonical ACO algorithm and some of its variants.

The structure of this chapter is as follows: Section 3.1 introduces the details of the framework of the canonical ant colony algorithms. Section 3.2 describes several main ACO variants proposed by the researchers. Section 3.3 summaries the chapter.

3.1 · Canonical ACO Algorithm

ACO is a distributed metaheuristic technique which constructs solutions by combining the heuristic information and the pheromone trails. The heuristic information is learned from the problems or some empirical knowledge which can guide the algorithm to search the most promising search spaces. The pheromone trails are the knowledge attained from the previous search which has both reinforce and evaporate mechanisms that can lead the algorithm to high quality solutions by using the historical data.

Hereby, we introduce the most canonical ACO algorithm framework. The pseudo code of this algorithm can be seen in Algorithm 1. After initializing the parameters and pheromone trails, each ant generates its own solution for the problem and a **LocalPheromoneUpdate** rule is used to update the pheromone trails which would affect the next construction of the ant. When all ants generate their solutions, there is an option to apply the local search strategy to improve the obtained solutions. Then the **Global PheromoneUpdate** rule is implemented to update the pheromone trails for the next iteration. The iteration repeats until the termination condition is reached. The main steps of the algorithm are described in detail in the following.

Algorithm 1 Canonical Ant Colony Optimization Algorithm

```
Initialization
repeat
    for all ants do
        ConstructSolution
        LocalPheromoneUpdate (optional)
    end for
    LocalSearch (optional)
    GlobalPheromoneUpdate
until termination condition reached
return best solution found
```

Initialization: In this stage, all parameters of the algorithm are set based on the parameter tuning methods and the pheromone trails are given an initial value τ_0.

ConstructSolution: A solution of a problem is from an infinite set of solution components which can be transformed to a weighted fully connected graph $G = (V, A)$, where $V = \{v_0, v_1, \dots, v_n\}$ is a vertex set and $A = \{(v_i, v_j); i \neq j\}$ is an arc set, where (v_i, v_j) represents the edge from vertex i to vetex j with a corresponding weight τ_{ij} which means the pheromone trails laid on this edge. Each ant starts from an empty solution and extended by adding a solution component from the neighborhood set N, which can be defined based on the problem, without violating any constraints. If no solution components can be added to the solution anymore, the construction procedure stops.

There are many transition rules proposed to choose the next solution components from the neighborhood set $\mathcal{N}$. The next solution components are selected probabilistically using a probability distribution that combines the heuristic information and pheromone trails. The probability distribution can be seen in Equation 3.1:

$$p_j(v_i) = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{x \in \mathcal{N}_i} \tau_{ix}^{\alpha} \cdot \eta_{ix}^{\beta}} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

with τ_{ij} being the pheromone level on edge (v_i, v_j), η_{ij} the heuristic desirability of edge (v_i, v_j), α the influence of τ on the probabilistic value, β the influence of η on the probabilistic value, $\mathcal{N}_i$ the set of nodes that can be visited by an ant positioned at node v_i, and $\tau_{ij}, \eta_{ij}, \alpha, \beta \geq 0$.

The attractiveness η_{ij} of edge (v_i, v_j) indicates the desirability to chose v_j as the next solution component which shows the heuristic information that can be derived from the problem. Different problems could have their own η_{ij}. For example, in TSP and VRP problems, η_{ij} is usually set to $1/d_{ij}$, where d_{ij} is the length of the edge (v_i, v_j). The pheromone trail τ_{ij} represents how good the edge (v_i, v_j) performed in the history. The bigger τ_{ij} is, the higher chance the edge (v_i, v_j) is selected. The tradeoff of the heuristic information and pheromone trails can be tuned by changing the parameters α and β. Here, $\alpha = 0$ means a greedy approach that never uses any information of the previous solutions, whilst $\beta = 0$ represents a pure searching method without any known information about the problems. When running the algorithm, the parameters should be tuned based on the problems.

LocalSearch: This is an optional step which aims at improving the solutions generated by the construction step. There are many local search algorithms that can be integrated with ACO, such as 2-opt, 3-opt and cross-exchange. They can be applied to the solutions generated by all ants or the best one of them. Based on the research

Dorigo et al. [Dorigo and Blum, 2005], the ACO with local search algorithms performs better than the original ACO in many cases and can reach the best performance in most of the combinatorial problems.

PheromoneUpdate: The PheromoneUpdate rules shows how the pheromone trails are maintained in each iteration of the algorithm. There are mainly two parts of the pheromone updating. The first one is reinforcing the pheromone. After ants generate their solutions, the chosen ants deposit more pheromone on the edges they go through. By doing this, the algorithm has a higher probability to search the path of the chosen solutions. The reinforcing rule of edge (v_i, v_j) is defined in Equation 3.2:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij} \tag{3.2}$$

where τ_{ij} is the pheromone trail level on edge (v_i, v_j), and $\Delta\tau_{ij}$ is the newlty deposited pheromone.

Since the reinforcing mechanism leads to a rapid convergence of the algorithm towards a local optimum, another mechanism called evaporation of the pheromone is applied. After each iteration, a certain percentage of the pheromone trail levels on all edges vanishes over time. This mechanism increases the probability to get out of the local optima and explore the neighborhood of the newest solution. By adding the evaporation mechanism, the update rule is given in Equation 3.3:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij} \tag{3.3}$$

where ρ is the evaporation rate.

The PheromoneUpdate rules can be used in two phases. The first one is called **LocalPheromoneUpdate** rule that is applied when each ant generates a new solution. By doing this, the ants would have a higher chance to explore different edges which can prevent the algorithm from being trapped in the local optima too early. The update rule can be expressed as follows: Each time an ant has traversed an edge (v_i, v_j), it applies Equation 3.4:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho\Delta\tau_0 \tag{3.4}$$

where $\Delta\tau_0$ is the constant initial pheromone trail increment, ρ is the evaporation rate.

The other one is called **GlobalPheromoneUpdate** rule. It is used when all μ ants constructed their solutions. Each ant k deposits a certain amount of pheromone on the path it goes through. The global pheromone update rule is given in Equation 3.5 and Equation 3.6:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{\mu}\Delta\tau_{ij}^{k}\,, (v_i, v_j) \in T^k, i = 0, \ldots, N, j = 0, \ldots, N, \tag{3.5}$$

$$\Delta\tau_{ij}^{k} = Q/L^k \tag{3.6}$$

where m is the number of ants, Q is a constant, T^k is the solution that is generated by ant k and L^k is the length of T^k.

3.2 • ACO Variants

There are many ACO algorithm variants in the literature [Colorni et al., 1992]. In the next subsections, some of the main variants are described.

3.2.1 • Ant System

The ant system was proposed in [Colorni et al., 1992] and tested on the traveling salesman problem. In this algorithm, no **LocalPheromoneUpdate** and **Local Search** are applied. Equation 3.1 is used as the transition rule to construct a solution. Equation 3.5 and Equation 3.6 are used in the $GlobalPheromoneUpdate$. Although it is the basic ACO algorithm, it consistently finds a good solution without being trapped in the local optima too rapid.

3.2.2 • Elitist Ant System

The first improvement is the elitist strategy proposed in the thesis [Dorigo, 1992]. In this algorithm, the best solution is given an extra weight for depositing pheromones. In this way, the shorter path is given a higher probability to be searched. The formula can be seen in Equation 3.7 and Equation 3.8:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{\mu}\Delta\tau_{ij}^{k} + \sigma\cdot\Delta\tau_{ij}^{*}\,, (v_i, v_j) \in T^k, i = 0, \ldots, N, j = 0, \ldots, N, \tag{3.7}$$

$$\Delta\tau_{ij} = \begin{cases} Q/L^* & \text{if } (v_i, v_j) \in T^* \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

where T^* is the best tour found so far, L^* is the length of T^*, and σ is the extra weight given to the best tour.

Later, this strategy is extended to a new version [Dorigo and Gambardella, 1997]. In the new version, only the ant with the best solution could deposit the pheromone on the edges. $\Delta\tau_{ij}$ is multiplied by the pheromone decay parameter ρ. The formula of this update rule can be defined as in Equation 3.9 and Equation 3.10:

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij}\ , (v_i, v_j) \in T^*, i = 0, \ldots, N, j = 0, \ldots, N, \tag{3.9}$$

$$\Delta\tau_{ij} = 1/L^* \tag{3.10}$$

where T^* is the best tour found so far and L^* is the length of T^*.

3.2.3 • Ant Colony System

The Ant Colony System (ACS) was proposed by Dorigo and Gambadella in [Dorigo and Gambardella, 1997]. There are several improvements for this algorithm. The first one is the transition rule which can be seen in Equation 3.11:

$$p_j(v_i) = \begin{cases} \arg\max_{j \in \mathcal{N}_i} \{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}\} & \text{if } q \le q_0 \text{ and } j \in \mathcal{N}_i \\ \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{x \in \mathcal{N}_i} \tau_{ix}^{\alpha} \cdot \eta_{ix}^{\beta}} & \text{if } q > q_0 \text{ and } j \in \mathcal{N}_i \\ 0 & \text{if } j \notin \mathcal{N}_i \end{cases} \tag{3.11}$$

with τ_{ij} being the pheromone level on edge (v_i, v_j), η_{ij} the heuristic desirability of edge (v_i, v_j), α the influence of τ on the probabilistic value, β the influence of η on the probabilistic value, $\mathcal{N}_i$ the set of nodes that can be visited by an ant positioned at node v_i, and $\tau_{ij}, \eta_{ij}, \alpha, \beta \ge 0$. Moreover, q denotes a random number between 0 and 1 and $q_0 \in [0, 1]$ a threshold.

The other improvement is that only the best solution is used in the global pheromone update rule. The formula can be seen in Equation 3.9 and Equation 3.10.

The last but not the least improvement is that a local updated rule is applied for the construction of the solution. The formula can be seen in Equation 3.4. Each time an ant generates a solution, the pheromone trails on the edges it goes through decrease based on the equations. It will encourage the other ants to search for a different solution rather than the same one.

3.2.4 • Max-Min Ant System

The Max-Min Ant System, also called MMAS, was introduced by Stützle and Hoos [Stützle and Hoos, 1996, Stützle and Hoos, 2000]. In this variant, the pheromone trails level is given an upper bound τ_{max} and a lower bound τ_{min}, which means $\tau_{max} \leq \tau_{ij} \leq \tau_{min}$. The initial pheromone trails are set to the upper bound so that the algorithm has a higher exploration at the beginning. By properly controlling the upper bound and lower bound, the algorithm is able to avoid premature convergence.

3.2.5 • Multiple Ant Colony System

The Multiple Ant Colony System (MACS) is the state-of-the-art ant colony optimization algorithm which was proposed by Gambadella et al. [Gambardella et al., 1999a]. In this algorithm, two ant colonies are used to search for two objectives. The colonies communicate with each other by sending their information to a controller. The controller stores the information of the pheromone trails of different colonies and also decides when to rerun the colonies. Through the cooperation of different colonies, the algorithm performs better in most of the tested problems. Moreover, since the colonies are independent of each other, it is very convenient to implement it in parallel. The implementation of this algorithm on the vehicle routing problem can be seen in Chapter 4, Chapter 5 and Chapter 6.

3.3 • Summary

Ant colony optimization techniques, following the concept learned from the foraging behaviors of the ant colony, are applied in various combinatorial problems. In this chapter, we introduced the canonical ant colony optimization algorithm and some of its variants. The ant colony algorithm contains the initialization, construction of solutions, pheromone updating and local search methods. Our algorithm used in this thesis is inspired by these concepts. The state-of-the-art MACS is extend to solve the dynamic vehicle routing problem with time window and also for the version of the problem with multiple priority levels.