Ant Colony Optimization

Student(s) : Mehmet Ali Uyar
Student ID: 118200066

Introduction

Ants are really hardworking, smart creatures but they are almost blind. They communicate with each other with their pheromones. While an ant looking for food, it follows ants pheromones. It follows other ants routes. Far or near, it does not matter. But pheromone vanishes over time just like everything in world. That means more ants on the same route, powerful pheromone level and more chance to find food.

Ant system is the first step of ACO algorithms. This algorithm inspired by behaviours of ants and their scavenging. ACO algorithm uses parallel search idea on problem and provides that a problem's solution ways and quality of previously obtained solutions.

ACO (Ant Colony Optimization)

When an ant move from A (start point) to B (destination), an ant leave pheromone for marking to help other ants to reach food. Algorithm uses that idea to create solution ways and decides which solution is efficient based on pheromone. An approach of algorithm can be shown in Figure 1.

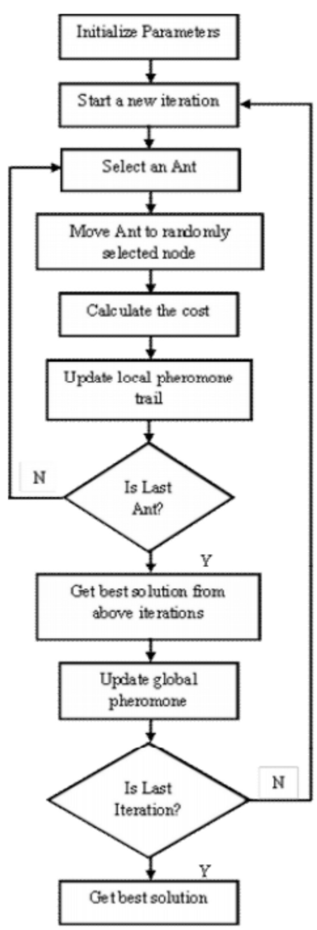

Figure 1: Steps of ACO

While an ant going along the road, it can follow other ants roads or can create a different route itself. Every road has it's own probability for each ant depends on pheromone level on the path and distance between locations. This approach called Ant system.

First off, pheromone level should be calculated. In Figure 2 it can be seen as :

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \Delta\tau_{i,j}$$

Figure 2: Formula of Pheromone Update

where P is a parameter called evaporation rate and changes between 0 and 1. Left side of the sum is represents pheromone loss over time. $T_{i,j}$ is stands for previous phereomone levels on the path.For every path from i to j, pheromone level increases when ant passes. But pheromone level will decreases overtime.

Right side of the sum is represents positive pheromone level which ant's left in road while passing. It is sum of all ants in colony where Lk is tour length of k-th ant in formula :

$$\Delta\tau_{i,j}^{k} = \begin{cases} 1/L_k & \text{if ant } k \text{ travels on edge } i,j \\ 0 & \text{otherwise} \end{cases}$$

Figure 3: Positive Pheromone Update

That means ants deposits more phoromone on shorter distances at same amount of time then long ones.

After all, probability of road can be calculated by :

$$p_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}$$

Figure 4: Probability of Possible Roads

where T pheromone level from i to j, n is distance between i to j, alpha and beta is constant values.

Ant Colony System is similar to Ant System but it has differences. Ants in ACS use the preudorandom proportional rule. Probability of city depends on random variable q which changes between 0 and 1 and parameter q0. Other than that system also uses local pheromone update which updates pheromone level on the path after each ant starts to move.

Algorithm can be apply many general problems like bus routes, telecommunication, protein folding, scheduling etc.

Implementation of the ACO

Environment:

Environment is a method that returns distance matrix which represents length between cities and pheromone matrix for pheromone level between paths.

.Distance matrix created with data frames and all cities distance is 10 by default. But ***path*** parameter gives information about which ways are shorter so method updates this path distance by 1.

Pheromone matrix is also created with data frame and filled with default value 0.0001 that ***eps*** holds which stands for level of pheromone at start.

Ant Class:

Using for represent ants. Each ant has a unique number, current city that ant stands , start and end point and route that take along the destination point.

Ants can move to the next city using via ***move()*** method. First method checks if current city is end point of route. If not , it takes distance matrix's current city column and from pheromone matrix column of possible cities and puts in formula. Outcome gives probabilities of next destination and choosen depends on probability values randomly. After selecting next city, it becomes current city and updates pheromone level on taken path.

To update pheromone level on path, use ***positive_pheromone_level(delta = 1)*** method. Delta parameter is constant in the formula. It calculates pheromone level of ant then adds it to pheromone matrix on certain path.

Cost() method can be used to choose efficient way of solution. Method calculates taken distance of every ant took after it reaches the destination point. Less cost means better way to solve problem.

Ant Colony Class:

Class stands for grouping all ants and follow their moves. Properties of the ant colony class is list of ants that puts all ants with their information in to the list and counts of ants.

start(time = 10) method moves each ant in the colony to next city and repeats it and updates pheromone level. Next ant takes its first move after previous ant finishes its route. Time for how many times will ant move.(Default 10)

Method ***negative_pheromone_update(ant , env_rate=0.01)*** updates pheromone level of all previous cities that ant take . env_rate is stands for environment decay rate. Unlike ant move, it effects pheromone level negatively and multiplies previous phereomone level on the path by (1-env_rate).

route() method returns route of all ants in colony.