TRAVELLING SALESMAN PROBLEM USING ANT COLONY OPTIMIZATION

Heli Shah
SU ID: 733914949
Syracuse University
Syracuse, USA
hpshah@syr.edu

Suket Singh
SU ID: 923377656
Syracuse University
Syracuse, USA
ssingh45@syr.edu

Abstract

The classical Travelling Salesman Problem involves finding the shortest path between a set of points and locations that must be visited. These points are the cities that a salesperson might visit. This problem is focused on optimization and often used to find most efficient route for data to travel between nodes. TSP is often concerned to find the cheapest solution.
Here, in our study, we present the solution of this Travelling Salesman Problem using the Ant Colony Optimization method and then do a comparative study with the Genetic Algorithm.
This document is intended for discussing the Ant Colony approach for the TSP and explaining the whole process.

Introduction

We describe artificial ant colonies that can solve the Traveling Salesman Problem. Ants of artificial colonies can use the information accumulated in the form of pheromone trajectories. These can be found on the TSP graph deposited on the edge to generate a possible tour. These tour can be assumed of getting progressively shorter. Computer simulations demonstrate that colonies of artificial ants are capable of producing excellent solutions for both type of instances of TSP: symmetric and asymmetric. This method is an example, like neural network, simulated annealing and evolutionary computation. It depicts the success of using natural metaphors to design optimization algorithms.

Artificial Ants Phenomenon

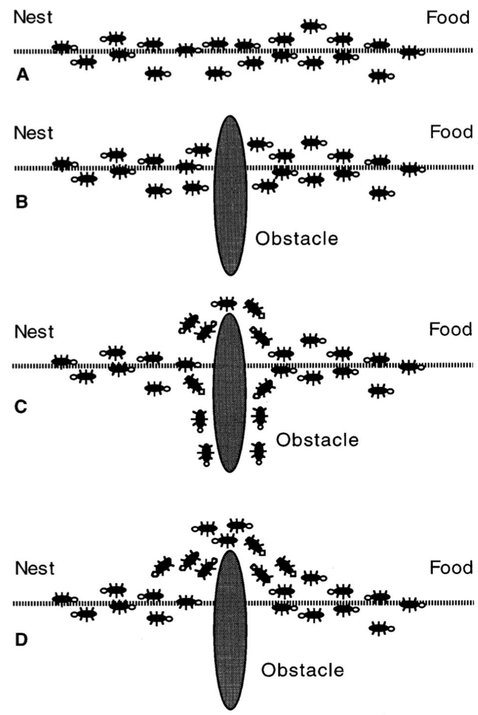
Artificial ant was taken as a mobile agent from another city on the TSP graph. The city to move is selected using a probability function. This function uses both accumulated TRAIL and heuristic values at the edge. We chose here this heuristic value to be a function of the edge length. These artificial ants are programmed to prefer those cities that are connected by high amounts of pheromone trail and edges, depending on the probability function, and by proximity. At first, m artificial ants are considered and placed above randomly selected cities. At each time step, it moves to a new city. Change the pheromone trail to the edge where they are used. This is called a local trail update. Once all the ants have completed the tour, the calculations will be made about the ants that were on the shortest tour. This ants correct the side that belongs to that tour, known as the global trail update. Here, the amount of pheromone trail, which is inversely proportional to the length of the tour, is added to the edge.

We used some of the behaviors by any natural ant and transferred them to our colony of ants:

(i) the paths with a high pheromone level should be preferred

(ii) the paths should be shorter if the rate of growth of pheromone amount is high

(iii) the trail mediated communication among ants



(A) Real ants always like in nature, following the path for the food from their nest.

(B) If an obstacle hinders their path: the ants select either to turn left or right with equal probability. Pheromones are deposited more quickly in short paths.

(C) Given that the turnings are of equal probability, we can see that half ant choose to turn right, and the other half turn left. A very similar situation is beyond the obstacles.

(D) All ants are obliged to choose the shorter path.

In our Ant Colony System (ACS), the city being r, has k artificial ants to choose from to move among those that do not belong to its working memory *Mk* by applying the following probability formula:

$$s = \begin{cases} \arg\max_{u \notin M_k} \{[\tau(r,u)] \cdot [\eta(r,u)]^{\beta}\} & \text{if } q \le q_0 \\ S & \text{otherwise} \end{cases}$$

Where:

τ (r, u) : The amount of pheromone trail on the edge of (r,u)

η (r, u) : A heuristic function chosen to be the inverse of the distance between cities r and u

β : A parameter which weighing the relative importance of pheromone trail and of closeness

q : A value chosen randomly with uniform probability in [0, 1]

$q0$ ($0 \le q0 \le 1$) : A parameter

S : A random variable chosen in accordance with a distribution of probability below the short, which is in favor of the edges with higher levels of pheromone trails.

$$p_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)] \cdot [\eta(r,s)]^{\beta}}{\sum_{u \in M_k} [\tau(r,u)] \cdot [\eta(r,u)]^{\beta}} & \text{if } s \notin M_k \\ 0 & \text{otherwise} \end{cases}$$

where

$pk(r,s)$: This depicts the probability with which ant k has to chooses to move from city r to city s.

It is only at the edge that belongs to that tour, when artificial ants complete their tour, the best ants deposit pheromone at the visited edge. By this, we are following the ant colony system.

The pheromone amount τ (r, s) deposited on each of the visited edge (r, s) with the best ant is short, i.e., the amount of deposition of pheromone on the edge is inversely proportional to the length of the tour. This method of depositing pheromone is intended to emulate the characteristics of differential pheromone trail accumulation that was due to the interaction between time path length and continuity in the case of real ants .

The global trail updation takes place with the help of following the formula:

$\boldsymbol{\tau}(r, s)$<--(1−α) ·$\boldsymbol{\tau}(r, s)$+ α ·$\Delta\boldsymbol{\tau}(r, s)$

Where,

$\Delta\boldsymbol{\tau}(r, s)$ = (shortest tour) − 1.

Global trail updates are similar to reinforcement learning methods where better solutions have gained higher enhancements.

Algorithm of ACS:

Initialize the pheromone trail;

Don't end while construction:

Construction is done by ant passing through the graph;

Daemon activity: any use of local search heuristics; local or global; update pheromone trails;

End while

TSP USING GA:

The GA for TSP problem will follow the following steps:

1. Create the population
2. Determine fitness
3. Select the mating pool
4. Breed
5. Mutate
6. Repeat

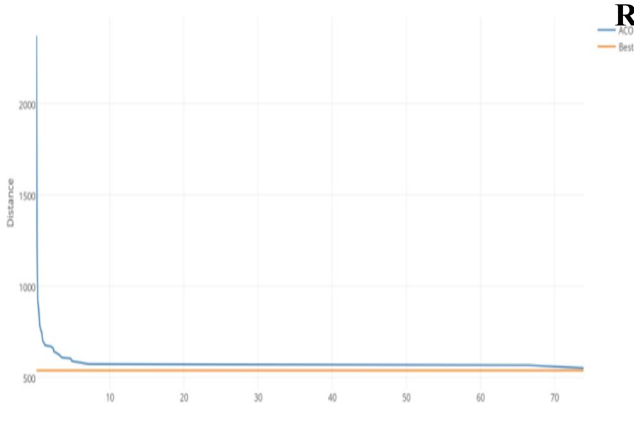
These steps are as normal as any GA would follow.

Given this, there are two important rules to keep in mind:

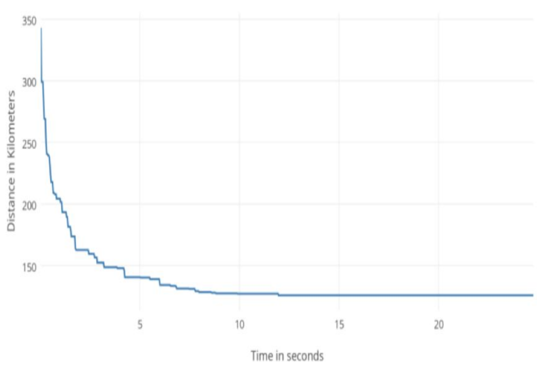
1. Each city needs to be visited exactly one time
2. We must return to the starting city, so our total distance needs to be calculated accordingly

RESULTs:

- Result for ACO:



- Result for GA:



COMPARISON:

- From the outputs of both the algorithms for traditional TSP, we observed that the best path in Ant Colony Optimization is achieved just only 823 iterations, while Genetic Algorithm achieves its best path after 50,000 iterations which is still far lower in efficiency when compared to ACO.
- Also, the ACO gives the best path to be of 3687 while GA gives that of 4734.
- Hence, ACO provides OPTIMAL as well as BEST solution for the addressed TSP.

REFERENCES:

1. https://pdfs.semanticscholar.org/b547/a2a20b88d1f01a10d0a8e67edd1132540e5d.pdf
2. http://www.scholarpedia.org/article/Ant_colony_optimization
3. http://ijarcsse.com/Before_August_2017/docs/papers/Volume_6/3_March2016/V6I3-0115.pdf
4. http://gist.github.com/turbofart/3428880
5. https://gist.github.com/NicolleLouis/d4f88d5bd566298d4279bcb69934f51d
6. https://en.wikipedia.org/wiki/Travelling_salesman_problem