Swarm Intelligence Technique ACO and Traveling Salesman Problem

Harsh Bhalani[1], Dr. Seema Mahajan[2], Prof. Zalak Vyas[3]

[1]Student, Dept. of Computer Engineering, Indus University, Gujarat, India.
[2]Head of the Dept. , Dept. of Computer Engineering, Indus University, Gujarat, India.
[3]Assistant Professor , Dept. of Computer Engineering, Indus University, Gujarat, India.

---***---

Abstract – *A* ***swarm*** *is a large number of homogenous, simple agents interacting locally among themselves, and their environment.* ***Swarm Intelligence (SI)*** *can be defined as a relatively new branch of Artificial Intelligence that is used to model the collective behavior of social swarms in nature. The inspiration often comes from nature, especially biological systems. The social interactions among swarm individuals can be either direct or indirect. Examples of direct interaction are through visual or audio contact, such as the waggle dance of honey bees. Indirect interaction occurs when one individual changes the environment and the other individuals respond to the new environment, such as the pheromone trails of ants that they deposit on their way to search for food sources. Examples in natural system SI include bacterial growth, ant colonies, bird flocking, and microbiological intelligence. This paper comprises a snapshot of ant colony optimization algorithm with its application in Traveling Salesman problem (TSP).*

Key Words: Swarm Intelligence (SI), Artificial Intelligence (AI), Ant Colony Optimization (ACO), and Traveling Salesman Problem (TSP).

1. INTRODUCTION

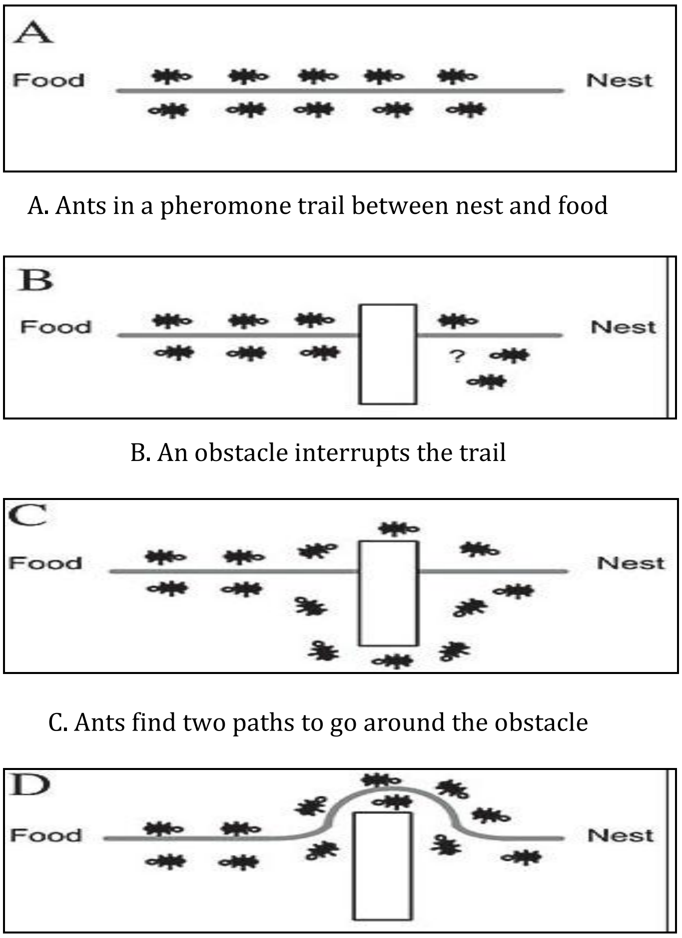
The various techniques of swarm intelligence used by researchers are as follows:

1. Particle Swarm Optimization
2. Ant Colony Optimization
3. Bees Algorithm
4. Artificial Bee Colony Algorithm
5. Differential evolution
6. Artificial Immune System
7. Bat Algorithm
8. Glowworm Swarm Optimization
9. Gravitational Search Algorithm

We mainly discuss the Ant Colony Optimization (A.C.O) algorithm in this paper.

1.1 Ant Colony Optimization

In 1991, Ant Colony Optimization (ACO) was introduced by M. Dorigo and colleagues for the solution of hard combinatorial optimization (CO) problems. ACO draws inspiration from the social behavior of ant colonies. It is a shown in Fig-1.



A. Ants in a pheromone trail between nest and food

B. An obstacle interrupts the trail

C. Ants find two paths to go around the obstacle

D. A new pheromone trail is formed along the shorter path

Fig -1: Pattern of path following by Ants

1.1.1 Ant Colony Optimization Metaheuristic

As shown in the basic flow of ACO in Fig-2, the objective of ACO's third step is to construct ant solutions (i.e., find the quality paths on the problem's construction graph) by stochastically moving through neighbor nodes of the graph.

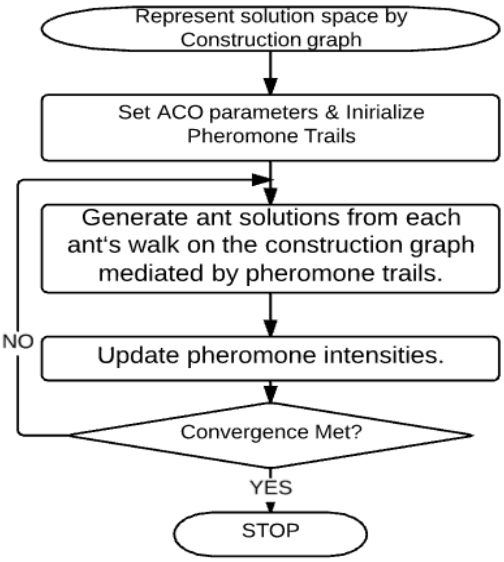


Fig -2: Basic Flow chart of ACO

Ants are driven by a **probability rule** to sequentially choose the solution components that make use of pheromone trail intensities and heuristic information.

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}*[\eta_{ij}]^{\beta}}{\sum_{l\in N_i^k}[\tau_{il}(t)]^{\alpha}*[\eta_{il}]^{\beta}}, & j\in N_i^k\\ 0, & j\notin N_i^k\end{cases}$$

Where:

- $P_{ij}^{k}(t)$ is the probability of the k^th ant to move from node i to node j at the t^th iteration/time step.
- $\mathcal{N}_i^k$ is the set of nodes in the neighborhood of the k^th ant in the t^th node.
- $[\tau_{ij}(t)]^{\alpha}$ is the pheromone amount on the arc connecting node i and node j, weighted by **α**.
- $[\eta_{ij}]^{\beta}$ is the heuristic value of the arc connecting node i and node j, weighted by **β**.
- The heuristic value η_{ij} is usually a non-increasing function in the moving cost from node i to node j.
- **α** and **β** are weight parameters that control the relative importance of the pheromone versus heuristic information, and application depend constant.

The pheromone trail updated by all ants in the iteration.

$$\tau_{ij}\leftarrow(1-\rho)\cdot\tau_{ij}+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

$$\Delta\tau_{ij}^{k}=\begin{cases}Q/L_k & \text{if ant } k \text{ used edge } (i,j) \text{ in its tour,}\\ 0 & \text{otherwise,}\end{cases}$$

Ants select next vertex by a stochastic function which depends on both pheromone and problem-specific heuristic $\eta_{ij}=1/d_{ij}$.

1.1.2. ACO Example: Traveling Salesman Problem

In TSP, the goal is to find the shortest possible tour from the salesman's home city to a finite number of customer cities with only one constraint that each city must be visited just once before finally returning to the starting home city.

There are many different ways to translate the above principles into a computational system apt to solve the TSP. In our *ant colony system* (ACS) an artificial ant *k* in city *r* chooses the city *s* to move to among those which do not belong to its working memory M_k by applying the following probabilistic formula:

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

where τ*(r,u)* is the amount of pheromone trail on edge *(r,u)*, η*(r,u)* is a heuristic function, which was chosen to be the inverse of the distance between cities *r* and *u*, β is a parameter which weighs the relative importance of pheromone trail and of closeness, *q* is a value chosen randomly with uniform probability in [0,1], *q*0 (0≤*q*0≤1) is a parameter, and *S* is a random variable selected according to the following probability distribution, which favors edges which are shorter and have a higher level of pheromone trail:

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

where $p_k(r,s)$ is the probability with which ant *k* chooses to move from city *r* to city *s*.

Local updating is intended to avoid a very strong edge being chosen by all the ants: Every time an edge is chosen by an ant its amount of pheromone is changed by applying the *local trail updating* formula:

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s)+\alpha\cdot\tau_0,$$

where τ_0 is a parameter. Local trail updating is also motivated by trail evaporation in real ants.

Typical parameter settings for TSP are: m = n (i.e., number of ants = number of cities), α = 1, β=2 to 5, ρ=0.5, and $\tau_0 = 10^{-6}$.

1.2. ACS: Ant Colony System

Ant Colony System has been introduced to improve Ant System's performance. ACS differs in 3 main factors from Ant System. ACS uses a more aggressive action choice rule than AS. Second is that, the pheromone is added only to arcs that belong to global best solution. Third, whenever ant uses an arc (i, j) to move from city i to city j, it removes some pheromone from the arc.

In ACS, ants choose the next city by using the *pseudo-random-proportional action choice rule*:

When located at city I, ant k moves, with probability q_0, to city l. It is the best possible move as decided by learned pheromone trails and heuristic information.

In ACS, only global best ant is allowed to add pheromone after every iteration. Thus, the update is modified to

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t)+\rho\cdot\Delta\tau_{ij}^{gb}(t),$$

Here, it is important to note that the trail update only applies to the arcs of the global-best tour.

Aditionally to the global updating rule, the ants use a local update rule that they apply after having crossed an arc during the tour construction:

$$\tau_{ij} = (1-\xi)\cdot\tau_{ij}+\xi\cdot\tau_0$$

where $\xi,\ 0<\xi<1,$ is parameter.The effect of this is to make an already chosen arc less desirable. So that exploration of not yet visited arcs is increased.

2. CONCLUSIONS

This paper presents an approach for solving Traveling Salesman Problem based on ant colony algorithm. The improved version of ACO algorithm based on Ant Colony System is also presented, which gives a better solution of traveling salesman problem compared to ACO algorithm.

REFERENCES

[1] Ant colonies for Traveling Salesman Problem. TR/IRIDIA/1996-3
- Marco Dorigo & Luca Gambardella

[2] M. Dorigo and T. Stützle. Ant Colony Optimization. MIT press, Cambridge, MA, 2004

[3] M. Dorigo, M. Birattari, T. Stuzle, Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique, IEEE Computational Intelligence Magazine, November 2006.

[4] Y. Zhang, Z-l.Pei, J-h.Yang, Y-c. Liang, An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Traveling Salesman Problem, IEEE 2007. 1-42441509-8.

[5] Swarm Intelligence: Concepts, Models and Application
-Hazem Ahmed & Janice I. Glasgow