



Solving Multistage Graph Using ACS Algorithm

Pankaj D.Khambre

Department of Computer Engineering
GHRCEM, Pune, India

Ms. Aarti Deshpande

Department of Computer Engineering
GHRCEM, Pune, India

Abstract— Ants release a chemical called pheromone while searching for food. Ants are capable of finding shortest path to their target. Observing their behaviour an ACS (Ant Colony System) algorithm was proposed. Earlier ACS has been used to give solution for many problems like travelling salesman problem. We propose to use ACS to solve Multi-stage graphs. Ant colony system can effectively give optimal solution in Multi-stage graphs. This approach can be used in real world applications like Grid computing, Data Mining etc.

Keywords— ACO, Data Mining, Multistage Graph, TSP

I. INTRODUCTION

ACS algorithm is used to find shortest and optimal paths in multistage workflows. In this algorithm as ants set out in search of food they follow the pheromone trail laid by previous ant and find out the shortest path to food. Each ant secretes some amount of pheromone while travelling in search of food. This way chemical gets deposited and the pheromone amount gets updated. The path which has higher accumulation of pheromone indicates that many ants have followed that path and it can be the shortest path. So the probability of other ants following this route increases. Similarly ACS algorithm is used to find shortest execution time of workflow. Workflows are partitioned into stages of multistage graph.

Multistage graph is a directed graph in which vertices are divided in several disjoint partitions. Each partition is called as a stage which contains several vertices. The first and last partition /stage of the graph contains one vertex each, namely, the source (s) and the sink (t).

In this paper we are going to find out shortest path in multistage graph using ACS algorithm. Which in turn will find out Maximum time a workflow can take to complete a task which is minimum time taken in sense.

II. THE BACKGROUND OF ACS

Ants solve complex problems such as searching for food, carrying the food and so on. They can find the shortest possible path between their nests and food. They deposit pheromone on the paths for their companions. When the next ants have to go through the paths, they select the path with high density of pheromone. Ants determine the next node on the route according to the density of pheromone present. Once all ants have terminated their search, the amount of pheromone on the tours will be modified. The ant algorithms have been designed to simulate the above ant behaviour to solve optimization problems. The ant system has usually been adapted according to the problems to be solved.

III. ANT COLONY SYSTEM

Pheromone Initialisation: Initially each ant is placed at source node of Multistage Graph. The rate of pheromone present on all the edges is Zero initially $\tau_0 = 0$. After the ant starts moving on to the route the initial amount of pheromone deposited at each path is 0.5. Every ant has its own pheromone. While an ant goes through the path, it will deposit pheromone on the path between two nodes.

Transition Rule: An ant K currently at node I choose to move to node j by applying the following probabilistic transition rule:

$$P_k(i, j) = \frac{[\tau(i, j)]^\alpha \cdot [\eta(i, j)]^\beta}{\sum_{l \in N_i^k} [\tau(i, l)]^\alpha \cdot [\eta(i, l)]^\beta}$$

Where artificial ant K in node i will choose node j, with probable value $P_k(i, j)$. In addition, $\eta(i, j) = \frac{1}{d_{ij}}$ is a heuristic function where d_{ij} is the distance between two nodes i and j. N_i^k Indicates the neighborhoods of node i where ant K has not yet met it. $\tau(i, j)$ is also indicating the rate of pheromone existing in the route.

Local Update: The rate of pheromone present in a route may not be remained in the same size and may be changed according to the equation as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

In this equation,

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{if } (i,j) \in \text{tour done by ant } k \\ 0, & \text{otherwise} \end{cases}$$

Where ρ the evaporation rate of pheromone and m is the number of ants.

Global Update: Global updating rule is performed after all ants have completed their tours. In order to make the search more directed, global updating is intended to provide a greater amount of pheromone to shorter tours and reinforce them. Therefore, only the globally best ant that found the best solution (i.e., the shortest tour) up to the current iteration of the algorithm is permitted to deposit pheromone.

ACS Algorithm:-

```

Initialize
Loop
/* at this level each loop is called as iteration */
    Each ant starts from the source node
    Loop
    /* at this level each loop is called as step */
    Each ant applies the state probability transition rule to
    Incrementally build a solution and
    A Local pheromone rule updating rule
    Until
    All ants have built a complete solution
    A Global pheromone updating rule is applied
    Until
    End_condition
    
```

IV. MULTISTAGE GRAPH

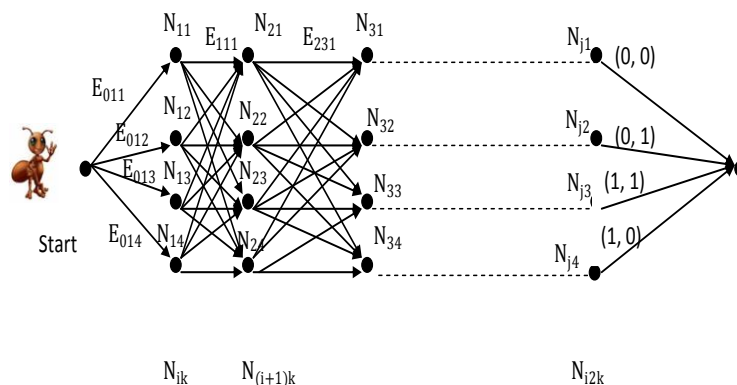


Fig.1 Multistage Graph

Implementation of ACS on MULTISTAGE GRAPH

- Each ants move randomly on paths. Initially Pheromone value on each edge of the weighted graph is 0. $\tau_{ij}(t)=0$. ρ is set between 0 and 1. $\Delta\tau_{ij}(t)$ is calculated which is equal to Q divide by sum of weights of each edge. After every iteration, pheromone value on edges is updated according to the formulae as follows

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{1}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{if } (i,j) \in \text{tour done by ant } k \\ 0, & \text{otherwise} \end{cases}$$

2. Probabilities of all the edges are calculated so that we can evaluate the shortest path to reach to the food. Higher the pheromone value on particular edge higher will be the probability of ants choosing that edge.

$$P_k(i,j) = \frac{[\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta}{\sum_{l \in N_i^k} [\tau(i,l)]^\alpha \cdot [\eta(i,l)]^\beta} \tag{2}$$

3. If the best solution in an iteration is not changed after some iterations. Then the best solution is the result. Otherwise repeat step 1.

VI. APPLICATION OF ACO ON MULTISTAGE GRAPH

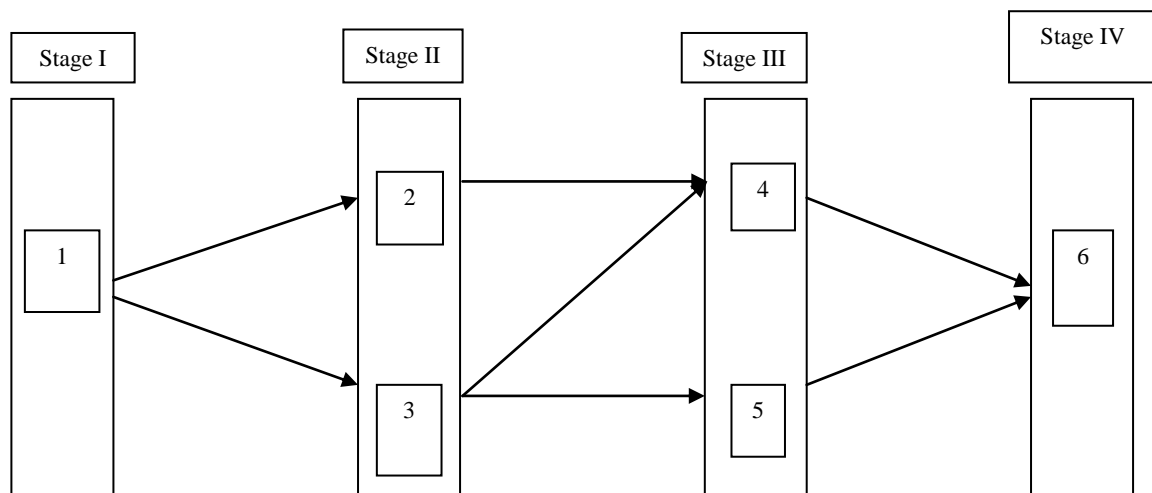
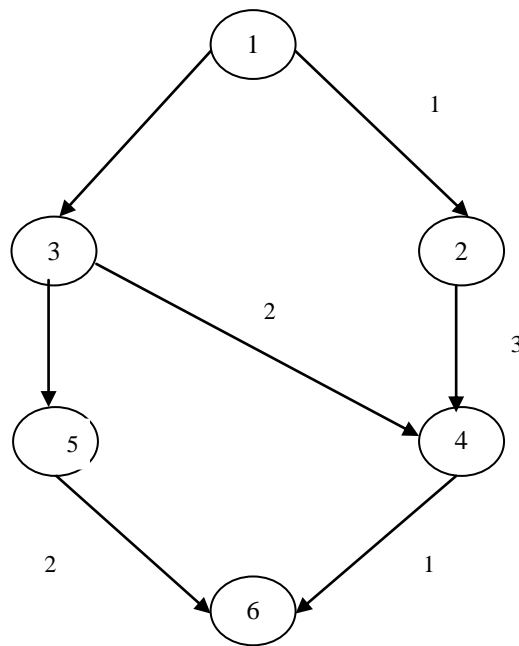


Figure 2. Multistage graph

Table 1 . Edge values of fig 2

	I	II		III		IV
	1	2	3	4	5	6
1	0	2	3	0	0	0
2		0	0	3	0	0
3			0	2	3	0
4				0	0	1
5					0	2
6						0

We employ figure 2 as an example of multistage graph. Node 3 is considered while ant k constructs the path in figure 2. When ant k moves from node 1 to node 3 and stops, it wants to choose the next node to which it moves as shown in figure 3.

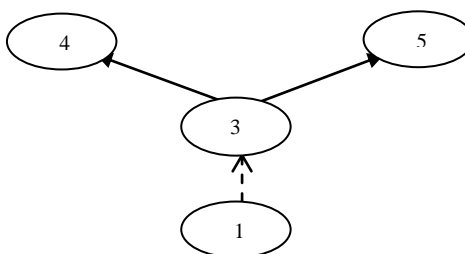


Figure 3. Node Diagram

According to the calculation of the probabilities, ant k selects a feasible edge. Here $\eta(i, j)$ is denoted by $1/d_{ij}$, where d_{ij} denotes the value of edge (i, j). In fig 2 node 1 is not considered because that it has been passed. In iteration t, the amount of pheromone on edge (3, 4) and edge (3, 5) are $\tau_{3,4}(t)$ and $\tau_{3,5}(t)$ respectively. The value of $\eta(3, 4)$ and $\eta(3, 5)$ are $1/2$ and $1/3$ respectively.

The value of α is 1 and so is β . The probabilities $P_k(3, 4)$ and $P_k(3, 5)$ are calculated as follows:

$$P_k(3, 4) = \frac{[\tau(3,4)]^{\alpha} \cdot 1/2}{[\tau(3,4)]^{\alpha} \cdot 1/2 + [\tau(3,5)]^{\alpha} \cdot 1/3}$$

$$P_k(3, 5) = \frac{[\tau(3,5)]^{\alpha} \cdot 1/3}{[\tau(3,4)]^{\alpha} \cdot 1/2 + [\tau(3,5)]^{\alpha} \cdot 1/3}$$

The amount of pheromone on all edges is updated after all ants create their own paths. Equation 3.2 is applied to calculate the amount of pheromone after the update step.

$\Delta\tau_{ij}^k(t)$ for multistage is defined as follows:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{if } (i, j) \in \text{tour done by ant } k \\ 0, & \text{otherwise} \end{cases}$$

Where L_k is the total length of the tour that ant k creates and Q is a constant.

We give an example to show the update of the artificial pheromone on the paths for candidate solutions to the problem. Here we still employ figure 1 as an example of multistage. Three ants are considered 1, 2 and 3. Ant 1 creates a path 1-2-4-6 another ant 2 creates 1-3-5-6 and the ant 3 creates a path 1-3-4-6.

Table 2 shows the total path length.

Table 2. Total Path Length

Path	L_k	$\Delta\tau_{ij}^k(t)$
1-2-4-6	6	1.6
1-3-4-6	4	2.5
1-3-5-6	6	1.6

Considering for multistage, we assume that Q is 10. In equation 2 we assume α is 0.5. Table 3 shows the update of pheromone on the three paths. From result we observe a fact that the amount of pheromone added to the paths with shorter total length is more than that added to the paths with longer total length.

Table 3. Actual Results

		Round1		Round 2	
		$\tau(i,j)$	(i,j)	$\tau(i,j)$	(i,j)
Path 1-2-4-6	(1,2)	1.6	0.21	1.28	0.11
	(2,4)	1.6	1	1.28	1
	(4,6)	1.6	1	3.25	1
Path 1-3-4-6	(1,3)	2.5	0.78	3.9	0.88
	(3,4)	2.5	0.70	3.7	0.60
	(4,6)	3.3	1	4.9	1
Path 1-3-5-6	(1,3)	2.85	0.78	4.8	0.88
	(3,5)	1.6	0.30	2.4	0.28
	(5,6)	1.6	1	2.4	1

After performing iteration a tour with shortest total length is extracted from the candidate solutions. If the shortest tour in the later iterations is not shorter than the tour, it is the heuristic optimal solutions.

According to the calculated values of pheromone density and probability we find the path 1-3-4-6 as the shortest path.

VI. Conclusion

Hence by using the concept of ACS algorithm in which ants try to find out the shortest path to their food, we solved multistage graph. Multistage graph is divided into various stages with one node as source node and one as sink node. Ants start moving from source node and by taking various paths reach to sink node where their food is available.

After an ant completes a path, pheromone is updated and the next ant calculates probability of using that path. Higher the pheromone value on that path higher will be the probability. And ant tends to choose higher probability path.

References

- [1] Zar Chi Su Su Hlaing, May Aye Khine, Member Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm 2011.
- [2] Vahid Modiri, Morteza Analoui, Sam Jabbehdari, Fault tolerance in Grid Using Ant Colony Optimization and Directed Acyclic Graph (2011),
- [3] Wei-Neng Chen, Jun Zhang Ant Colony Optimization Approach to a Grid Scheduling Problem With Various QoS Requirements 2009.
- [4] Tzung-Pei Hong, Ya-Fang Tung, Shyue-Liang Wang, Min-Thai Wu, Yu-Lung Wu An ACS based framework for fuzzy data mining. 2009 .
- [5] Wei-Neng Chen, Jun Zhang an Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem with Various QoS Requirements 2009.
- [6] Hung-Wei Huang, Solving Traveling Salesman Problem by Ant Colony Optimization (2004).