ISSN (online): 2320-9836
ISSN (Print): 2320-9828

Path identification between locations within a campus using ACO

A. Amali Asha *[1], T. Pramananda Perumal[2]

[1]Assistant Professor, Loyola College, Chennai, India
[2] Principal, Presidency College, Chennai, India
aamaliasha@gmail.com[1], pramanandaperumal@yahoo.com[2]

Abstract

Objectives: To identify the paths between locations within the college campus. The paths were stored to create a voice guidance system for the visually challenged students studying in our institution.
Methods: We have allotted number for locations, and each location has its neighbor's detail. A graph was generated by this information which gives a complete outline of connection among the locations. We have generated an algorithm based on Ant Colony. The algorithm was tested first with 9 locations and it was able to exactly list out all possible paths between sources and the destination.
Findings: Once the edge between vertices has been identified by an ant, then the pheromone level is maintained in that edge should be high. The pheromone level is kept above a value called threshold value. If pheromone level on a particular edge is below the threshold value then that path was omitted by other ants. The high pheromone level makes the other ants to proceed through that path. The current vertex is checked with the destination vertex to check whether the algorithm process has identified a path. Tests were conducted by considering all the locations within our campus, where our visually challenged students will go for their classes.
Application: All paths between the source and destinations are identified correctly and recorded. The voice guidance system is its incubation stage and surely this would help the visually challenged students to reach their destinations without others help.
Keywords: ACO, Path identification, Ant Colonies, Ant system, Swarm Intelligence, All possible paths.

1. Introduction

1.1. Ant Colony Optimization – ACO

During 1992, Marco Dorigo proposed the algorithm called AS – Ant System [1]. He presented this algorithm in his Doctoral Dissertation and described how to use this algorithm to find the solution to the classical traveling salesman problem. The basic idea of Ant System is that when an ant searches for food, it deposits a chemical substance called Pheromone on its path.

This chemical trail is used by other ants to reach the food location. This behavior is called Swarm Intelligence, where simple agents can solve complex problems without any centralized control. Depositing pheromone on their path is called stigmergy, which is an indirect co-ordination mechanism of exchanging information. This mechanism creates a favorable path which can be followed by all other ants who are the members of the colony. Ant Colony Optimization [2] was developed from Ant system, the popular optimization algorithm used to find out solutions to optimization problems [3] which are complicated. It is also known as a heuristic algorithm to solve variety of computational problems using graphs. Finding out all possible paths and a shortest path are best examples.

The shortest path technique is one of the great aspects used in navigation and also in the telecommunication field. The path between two locations is identified and the shortest path among all is used to transfer information packets.

1.2. Route information

For a specified source and destination the set of all possible routes are identified. Each node will have zero or more number of adjacent nodes. This information will be used as a base for finding out all possible routes. For each node the adjacent information is updated [4] by equation 1.

 ISSN (online): 2320-9836
ISSN (Print): 2320-9828

$$Aij = \begin{Bmatrix} 1 \ if \ i \text{ is adjacent to } j \\ 0 \qquad\qquad \text{Otherwise} \end{Bmatrix} \qquad 1$$

The above equation is used for updating adjacent information especially for graph with no edge weight. If the edges of a graph have some positive values and the information is represented by a data structure called as weight matrix.

In this paper we have considered only the adjacency information for finding out all possible routes from source to destination. The route is closely depending on the adjacency nodes of each and every node. For the implementation I have selected only certain locations. The graph includes only one section with limited locations of the campus. The coding has been done using python. The following are the location name used as node of a graph.

2. Methodology

The algorithm will be able to identify all paths between a selected source and destination. Usually there will be more than one path from a source to the destination. Variety of techniques is available to find a shortest path between the source and the destination. We have considered the adjacency nodes of a source and ants are assigned on these nodes. If there are 'n' adjacent nodes then 'n' ants are assigned to find the path. Because of the pheromone each ant colony will be moving to the route of its predecessor.

Each ant colony will be assigned with different pheromone value. Similarly a threshold value will also be assigned. The pheromone value will be strengthened if its value is below the threshold value so that the successor will be able to select the correct path.

2.1. Ant movement

Ants will move from one node of a graph to another based on the deposited pheromone level. The ant will decide which node has to be selected next to proceed to complete the path is based on the following equation 2.

$$\mathcal{C}_{ij} = \frac{\tilde{\iota}_{ij}{}^{\alpha} . \eta_{ij}{}^{\beta}}{\sum \tilde{\iota}_{i,l}{}^{\alpha} . \eta_{i,l}{}^{\beta}} \qquad 2$$

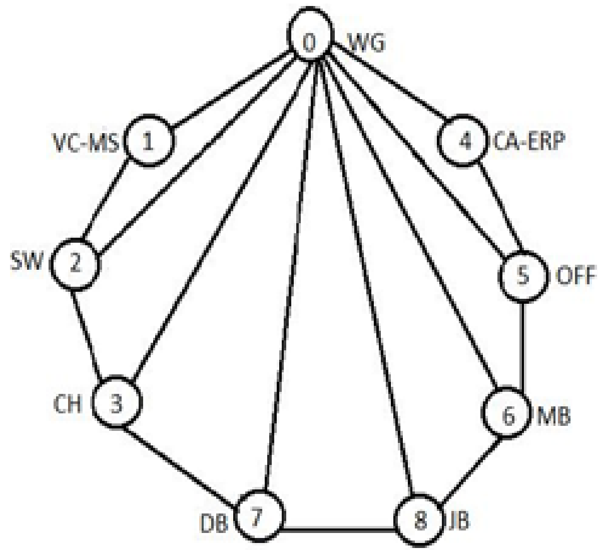
The probability ρij defined here is to move ant from i^{th} node to j^{th} node. The pheromone on edge from i to j is denoted by a term τij. The parameter α is used to control the influence of the amount of pheromone τij. The desirability is denoted by ηij and to control this, a parameter β is used.

3. Implementation

The implementation is done using Python. The locations in the college are given a number starting from 0. The graph is created by considering the locations and the location i and the location j by an edge. The graph which I want to apply the algorithm is shown in Figure 1.The location names are listed in the Table 1. The adjacency list is used in the algorithm to decide and find out a path. The number of nodes of a graph is received as input and for each node the adjacency list is also received as input. For a given source and a destination the set of all paths [5] are identified using the algorithm. To identify the paths, the source node's adjacency list is considered and an ant is placed in one of the adjacent node in the list. The edge between the source and the current vertex is added to the path and the current vertex is marked as visited.

The decision of selecting next node is based on the adjacency list of the current node and also the decision [6] based on equation 2. Once the edge between one vertex to an another is identified then the pheromone level is maintained in that edge should be high. The pheromone level is kept above a value called threshold. Whenever a new location is reached the pheromone on the edge is checked. If pheromone level [7] [8] on a particular edge is below the threshold value that path was omitted by other ants. The high pheromone level makes the other ants to proceed through that path.

Figure 1. Location graph



The current vertex is checked with the destination vertex to check whether the algorithm process has identified the path. This process will be continued for all other vertices until the current vertex and the destination are same [9]. Each time the path is updated. There are two parameters used in the algorithm, one to store the visited vertices (vervisited) and the other to store the path (pathgraph). This process will be repeated for all the vertices in the adjacency list of a source and all the paths are printed.

Table 1. Location names

Node	Name
WG	Wicket Gate
VC-MS	Vis. Com, Dept. of Media Studies
CA-ERP	Computer Academy, ERP
SW	Dept. of Social Work
CH	Chemistry Block
DB	Display Board
JB	Jubilee Building
MB	Main Building
OFF	Office–Principal, Secretary, Shift-I

3.1. Algorithm

Input:

No. of locations.
Locations and the corresponding adjacency list.
Source location and the destination location.

Output:

All possible paths from source to destination

Begin
n=no. of vertices
Graph(n)

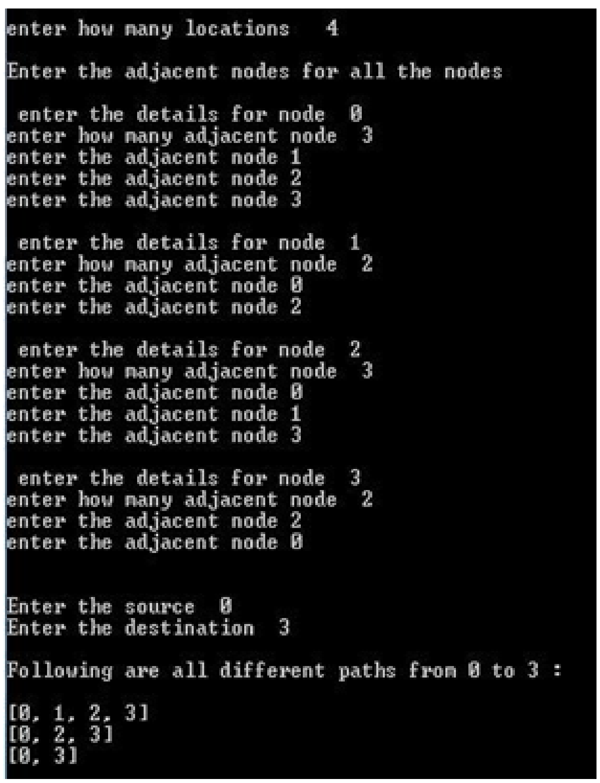
 ISSN (online): 2320-9836
ISSN (Print): 2320-9828

For each vertex
Get the adjacent nodes S or=source, Dest=destination
Ant decision to move to the next location
Pheromone update process
Threshold checking
Graph Append (v)
Call recursively the method Pathsallfind (Graph, sor, Dest, Vervisited, Paths graph)
Print path graph
End

The graph we have considered is as shown in Figure 1. Nine locations in the campus were considered and the locations were connected with undirected edges. The edges are undirected to indicate the two way path between nodes.

The selected locations are 0-WG, 1-VS-MS, 2-SW, 3-CH. The source location was 0-WG and the destination location was 3-CH. The adjacent nodes for all the vertices were given as for 0 (1, 2, 3), 1 (0, 2) , 2 (0, 1, 3) , and for 3 (0, 2).The set of all possible paths from the source vertex 0 and the destination vertex 3 had been listed in the Figure 2.The table shows the nine locations chosen with in a campus their detailed name. These locations are the vertices of a graph and the presence of edge shows the connectivity between the locations. The edge connecting the two locations has no direction. This is to indicate that the path between the two vertices is bidirectional.

Figure 2. Output paths



```
enter how many locations   4

Enter the adjacent nodes for all the nodes

 enter the details for node  0
enter how many adjacent node  3
enter the adjacent node 1
enter the adjacent node 2
enter the adjacent node 3

 enter the details for node  1
enter how many adjacent node  2
enter the adjacent node 0
enter the adjacent node 2

 enter the details for node  2
enter how many adjacent node  3
enter the adjacent node 0
enter the adjacent node 1
enter the adjacent node 3

 enter the details for node  3
enter how many adjacent node  2
enter the adjacent node 2
enter the adjacent node 0


Enter the source  0
Enter the destination  3

Following are all different paths from 0 to 3 :

[0, 1, 2, 3]
[0, 2, 3]
[0, 3]
```

Output - Paths:

The set of all possible routes identified by the algorithm is as shown below. The source considered for testing the algorithm is wicket gate and the destination is chemistry block. The paths from the source, wicket gate to the destination – chemistry block are

WG→VC-MS→SW→CH-WG→SW→CH-G→CH.

4. Conclusion

In this paper, the algorithm was tested by considering only limited locations and found it works correctly. The algorithm also tested by changing the source and destination. For the given source and destination the algorithm produces all possible paths. So this algorithm can be used for the graph with any number of vertices. For different source and destination pattern this method listed the all possible paths correctly. The same algorithm is used in this paper to find out all possible routes in our college campus.

The outcome of this method will be used as an input to the system which gives voice guidance to the visually challenged students. This voice guidance system is in its incubation stage and surely would help the visually challenged students to reach out various locations within our college campus.

5. References

1. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man*, *and Cybernetics*. 1996; 26(1), 1-10.
2. M. Dorigo, C. Blumb. Ant colony optimization theory: A survey. *Theoretical computer Science*. 2005; 344 (2-3), 243-278.
3. Marco Dorigo, Thomas Stutzle. Ant Colony Optimization. The MIT Press. 2004.
4. Amali Asha, S.P. Victor, A. Lourdusamy. Performance of ant system over other convolution masks in extracting edge. *International Journal of Computer Applications* (0975 – 8887). 2011; 16(4), 1-6.
5. W. Peng, X. Hu. A fast algorithm to find all-pairs shortest paths in complex networks. *Procedia Computer Science*. 2012; 9, 557-566.
6. C.L.Azevedo, J.L. Cardoso. Vehicle tracking using the k-shortest paths algorithm and dual graphs. *Transportation Research Procedia*. 2014; 1(1), 3-11.
7. AAljanaby, K.R. Ku-Mahamud, N.M Norwawi. Optimizing large scale problems using multiple ant colonies algorithm based on pheromone evaluation technique. *International Journal of Computer Science and Network Security (IJCSNS)*. 2008; 8(10), 54-58.
8. N.Aljanaby, K.R. Ku-Mahamud, N.M. Norwawi. A new multiple ant colonies optimization algorithm utilizing average pheromone evaluation mechanism. *Proceedings of the Knowledge Management International Conference*, Langkawi, Malaysia. 2008; 531-534.
9. Dilpreetkaur, P.S. Mundra. Ant colony optimization: a technique used for finding shortest path. *International Journal of Engineering and Innovative Technology (IJEIT)*. 2012; 1(5), 1-3.

The Publication fee is defrayed by Indian Society for Education and Environment (www.iseeadyar.org)
***Cite this article as**:*
A. Amali Asha, T. Pramananda Perumal. Path identification between locations within a campus using ACO. *Indian Journal of Economics and Development*. Vol 6 (5), May 2018.