ACO Based Shortest Path between Locations within a Campus

Maria Wenisch S [1], Amali Asha A [2]

[1]Loyola Institute of Business Administration, Chennai, India
[2]Loyola College, Chennai, India

Abstract **– Modelling the behaviour of nature is the most inspiring field among researchers. Many algorithms are developed based on the behaviour of nature to solve problems including solution to very complex problems. Ant Colony Optimization - ACO is a best optimization algorithm used almost in all the fields to find solution to the difficult combinatorial problems. In our paper we are using ACO to find out the shortest path between the locations within the campus. This is used to give path direction to the visually challenged students within our campus to reach out the various places. The shortest path information will be used as a basis for the design of a voice monitoring system. This voice monitoring system guides the visually challenged students.**

Keywords **– Optimization algorithm, shortest path, voice monitoring system, ACO, Ant Colonies, Swarm Intelligence, pheromone.**

1. Introduction

1.1 Origin of the problem

Our college has been extending whole hearted support for the education of the differently-abled students. To facilitate the education, various developments have taken place in our campus. A separate center called RCDA – Resource Center for the Differently Abled was inaugurated and is equipped with advanced devices to enable the students to grow intellectually. Around 95 visually challenged students admitted in our campus to various undergraduate and post graduate courses. For their language and allied papers they have to go from one place to another along with other students. Also, if they want to meet the faculties handling all these papers they have to go to the department. To reach the particular place they need help from their friends. This makes us to propose a system which can guide them to reach the particular location without others help. In this paper we have constructed a graph with all locations from which the minimum distance path will be selected to guide the students.

DOI: 10.18421/TEM74-34
https://dx.doi.org/10.18421/TEM74-34

Corresponding author: Amali Asha A,
Loyola College, Chennai, India
Email: aamaliasha@gmail.com

Received: 30 July 2018.
Accepted: 11 October 2018.
Published: 26 November 2018.

1.2 Ant Colony Optimization

Modeling the behavior of nature is the most inspiring field among researchers. Many algorithms are developed based on the behavior of nature to solve problems including solution to very complex problems. Swarm intelligence is one of the best techniques among the nature inspired algorithms. Swarm intelligence techniques are based on the behavior of insects in solving their real life situations. Ant Colony Optimization, wasp nets and particle swarm optimization are some of the methods developed by simulating the behavior of insects. Ant colony optimization is based on the behavior of real ants in searching and finding out their food.

Ant Colony Optimization technique was originated from a concept presented by Marco Dorigo [1] in his Doctoral Dissertation. Ant system was used by Dorigo to solve the traveling salesmen problem. It is completely based on a concept called pheromone deposited by ants. The real ants will deposit a chemical substance called pheromone on their path between the nest and the food. The process of finding out food by an ant contains various activities such as pheromone deposit, moving from one place to another to form a path between the nest and the food and pheromone evaporation to indicate that the path is not an optimal path. The ants travelled by the shortest path lay pheromone faster with high threshold value. This makes other ants likely to follow this shortest path.

1.3 Shortest path

Ant colony optimization [2] technique uses the basic idea of Ant system. This was used by many researchers to find a solution to optimization [3] problems. This algorithm is also used to solve computational problems using graph. Finding out the shortest path is the best example and this technique is used in navigation and in telecommunication field. In telecommunication field, the shortest path is used to transfer information packets.

Varieties of algorithm are available to identify the path between the source and the destination. There may be multiple routes between the particular source and destination. Using shortest path algorithms [8] we will be able to identify only one path and that path is the best path between source and destination.

2. Methodology

We have designed an algorithm to find out the shortest path between the given source and destination. There will be more than one path from source to destination [7]. Among the set of available paths the minimum distance path will be identified. That path is considered as a shortest path. We have considered for each node, the adjacency nodes to which it is connected with their distance to implement the algorithm. The adjacency nodes are called nearest neighbor list. Each node has information about the set of all its neighbors with the distance. The ants start from the source location and select the next location based on the minimum distance.

2.1 Choice for Next Movement

An ant will be placed at the source node. Among the set of its entire neighbor the minimum distance node will be identified and the ant will select that particular node as its next choice in the path. The ant will deposit the pheromone on its way. A threshold value is assigned for the pheromone level and always the pheromone level must be higher than the threshold. Pheromone deposited by the ant will be available only certain period of time and after that time limit the pheromone evaporates. To avoid the evaporation of pheromone on the minimum cost path, the pheromone's strength on that path will be maintained at high level. This makes the successor ants to select the correct path.

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 2.1.

$$\boldsymbol{\rho}\ \mathbf{i,j} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\Sigma\ \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} \qquad 2.1$$

The probability ρij defined here is to move ant from ith node to jth 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 original map with their location name is as shown in Figure 1. This map was taken from our college website which covers the entire campus locations. We have considered all locations to identify the path and tested the results. In this paper we have given only one graph and the corresponding path between the source and the destination.

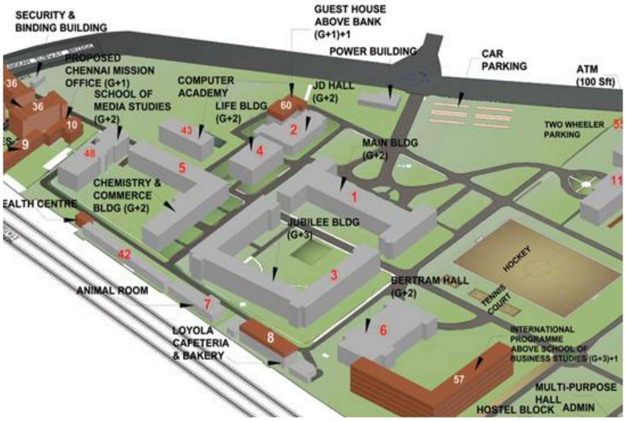

Figure 1. Loyola College map

The locations at the college are assigned with labels such as "C, E, O, M, L, J, N, B and X". The graph is created by connecting the locations and the edges were assigned with the distance value. In our graph we have considered only nine locations and the implementation was done using that graph. The graph which we have used in the algorithm is shown in Figure 2.

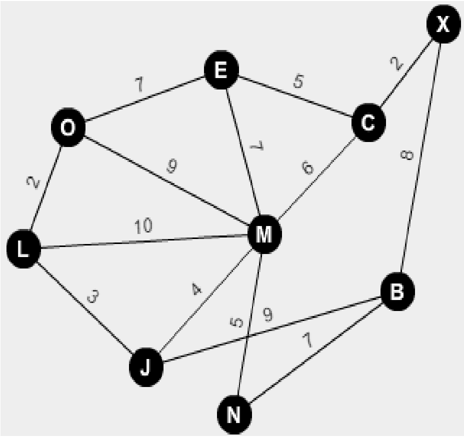

Figure 2. Location Graph

In our paper we have represented the locations as nodes of a graph and the distances between the nodes were represented over the edges. The existence of path between two nodes is denoted by P_{ij}. If there is a path between two locations, the length of that path is denoted as D_{ij}. This D_{ij} is updated whenever a location is added into the path.

Adjacency list is maintained for each and every location in a graph to denote the list of nodes to which the current node is connected and the corresponding distance. Moving to the next location basically depends upon the information presented on Adjacency list and the distance matrix. The location names are listed in Table 1.

Table 1. Locations Label

LABEL	NAME
E	Entry Gate
C	Car Parking
X	Exit Gate
O	Office – I
M	Main Building
L	LIFE Building
J	Jubilee Block
B	Bertram Hall
N	New Com & Eco Block

The nearest neighbor list for each node is given as input for the algorithm to find out the shortest path. The algorithm gets the number of locations, location names and the nearest neighbor list as input. By using all the information, the algorithm identified the shortest path with the path length.

The algorithm accepts the source and the destination location from the user. To find the shortest path, the source node's adjacency list was selected and the ant placed in the source location will decide the next location and moves towards the new location. The path between the source and the current location is added to the path and the current location is marked as visited.

The decision of selecting the next location is based on the neighbors list of the current location and also the decision [4] based on equation 1. Once the edges between locations are identified, then the pheromone level is updated with high value in that path. Always the pheromone level is kept above a value called threshold value.

Whenever a new location is reached the pheromone on the edge is checked. If pheromone level [5] [6] on a particular edge is less than the threshold value, that path was not followed by other ants. The high pheromone level makes the other ants to proceed through that path.

The current location name is checked with the destination location name each time, and if both are same, the algorithm stops by updating the shortest path. Mainly two parameters are used and updated each time. The first parameter is used to make note of vertices visited, and the second parameter is used to store the shortest path length.

Algorithm

INPUT:

No. of locations, Locations and the corresponding neighbors, Source and the destination.

OUTPUT:

Shortest path and the length of that path.

Begin
- *n=no. of locations*
- *Graph (n)*
- *For each vertex*
  - *Get the neighbors information*
- *Sor=source, Dest=destination*
  - *Ant decision to move to the next location*
  - *Threshold checking and Pheromone update process*
  - *Call recursively the method Shortestpath (Graph, Sor, Dest)*
  - *Graph.append (v)*
  - *Print shortest path length and the path*

End

The algorithm was checked by considering 6 locations and the selected locations are C (Car Parking), E (Entry Gate), M (Main Building), O (Office-I), L (LIFE Building) and J (Jubilee Building). The location graph with neighbor information is as shown in Figure 3. The source and destination is C (Car Parking) and L (LIFE Building). Multiple paths between the source and the destination are as shown in Table 2.

```
{'C': {'E': 5, 'M': 6},
 'E': {'C': 5, 'M': 7, 'O': 7},
 'J': {'L': 3, 'M': 4},
 'L': {'J': 3, 'M': 10, 'O': 2},
 'M': {'C': 6, 'E': 7, 'J': 4, 'L': 10, 'O': 9},
 'O': {'E': 7, 'L': 2, 'M': 9}}
```

Figure 3. Neighbor Information

Table 2. All Paths

PATH	LENGTH
C → E → O → L	14
C → M → E → O → L	22
C → M → O → L	17
C → M → L	16
C → M → J → L	13

Among all five paths, the path with length 13 is the shortest and the corresponding path is marked in the graph in Figure 4. The output of the algorithm is as shown in Figure 5.

```
Enter number of vertices: 6
Enter vertices(keys) :
E
C
O
M
L
J
{'C': {'E': 5, 'M': 6},
 'E': {'C': 5, 'M': 7, 'O': 7},
 'J': {'L': 3, 'M': 4},
 'L': {'J': 3, 'M': 10, 'O': 2},
 'M': {'C': 6, 'E': 7, 'J': 4, 'L': 10, 'O': 9},
 'O': {'E': 7, 'L': 2, 'M': 9}}

enter the source node name  C

enter the destination node  L

 The shortest weight and path is

(13, ['C', 'M', 'J', 'L'])
```

Figure 4. Shortest path

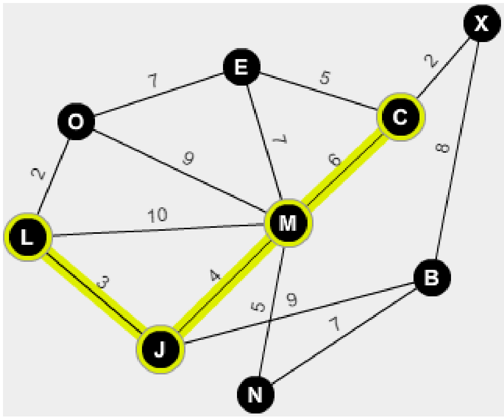

Figure 5. Shortest path Graph

4. Conclusion

The ACO approach identified the shortest path correctly between the locations. The algorithm was tested with more than 5 graphs with different locations within our campus. For all the cases, the algorithm found the best shortest path. Once the shortest path has been identified, a voice monitoring system will be designed to give direction to the visually challenged students to reach various locations within our campus. In our future work, the latitude and longitudinal points are identified for all the locations and the shortest path is stored to give input to the voice monitoring system. An android app will be developed and this would help them to reach their destination without any difficulties with the help of our guidance system.

Acknowledgements

We thank the University Grants commission for the financial support through the grant for the Minor Research Project (F.NO:4-4/2015-16(MRP/UGC-SERO).

References

[1]. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *26*(1), 29-41.

[2]. Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, *344*(2-3), 243-278.

[3]. Asha, A. A., Victor, S. P., & Lourdusamy, A. (2011). Performance of ant system over other convolution masks in extracting edge. *International Journal of Computer Applications (0975–8887)*, *16*.

[4]. Peng, W., Hu, X., Zhao, F., & Su, J. (2012). A Fast algorithm to find all-pairs shortest paths in complex networks. *Procedia Computer Science*, *9*, 557-566.

[5]. Azevedo, C. L., Cardoso, J. L., & Ben-Akiva, M. (2014). Vehicle tracking using the k-shortest paths algorithm and dual graphs. *Transportation Research Procedia*, *1*(1), 3-11.

[6]. Dilpreet Kaur, Dr. P.S Mundra. (2012). Ant Colony Optimization: A Technique Used For Finding Shortest Path, *International Journal of Engineering and Innovative Technology (IJEIT)*, *1*(5).

[7]. Rais, H. M., Othman, Z. A., & Hamdan, A. R. (2007, November). Improved dynamic ant colony system (DACS) on symmetric Traveling Salesman Problem (TSP). In *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on* (pp. 43-48). IEEE.

[8]. Hlaing, Z. C. S. S., & Khine, M. A. (2011). Solving traveling salesman problem by using improved ant colony optimization algorithm. *International Journal of Information and Education Technology*, *1*(5), 404.