IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 1, January 2021

DOI 10.17148/IJARCCE.2021.10113

Multi-Robot Path Planning using Dijkstra's Algorithm with Multi-layer Dictionaries

Saif Allah M. Abgenah[1], Azrul Amri Jamal[2], Syed Abdullah Fadzli[3]

Faculty of Informatics and Computing, Univerisiti Sultan Zainal Abidin, Malaysia[1]

Faculty of Informatics and Computing, Univerisiti Sultan Zainal Abidin, Malaysia[2]

Faculty of Informatics and Computing, Univerisiti Sultan Zainal Abidin, Malaysia[3]

Abstract: Path planning is the first task for a Robot to autonomously navigate, especially for autonomous Robots. For multi-Robot systems the process is more complex than a single Robot system. The commonly known algorithms for path planning usually finds solutions for single Robot systems and do not propose ideas for multi-Robot systems. In this paper an enhanced Dijkstra's algorithm for multi-Robot systems with a multi-layer dictionary is used to navigate multiple Robots on an indoor map autonomously and simultaneously. Simulation and Experimental results show that the proposed enhanced algorithm was able to generate paths for the multiple Robots that where navigating through the map simultaneously and assigning optimal or feasible paths for the Robots to navigate through.

Keywords: Dijkstra's algorithm; multi-Robot systems; multi-layer dictionaries; Path Planning.

I. INTRODUCTION

For an autonomous Robots, finding the shortest, most optimal and feasible path for the Robot to move is known as Path Planning. It also includes having a trajectory based on a map mentioning clear directions for both source and destination. This map of trajectory can either be indoors as well as outdoors. In the indoor environments, the trajectory is commonly surrounded by objects like furniture, walls, stairs, hallways, door, and even humans as well. However, the environment with no moving objects shall be called a Static Environment. The Robot is expected to move on a given trajectory without being blocked by these surrounding obstacles. For path planning, an algorithm known as Dijkstra's Algorithm is known to be the most famous one. However, this algorithm comes with a shortcoming of having a limited data pertaining to surrounding objects. Therefore, this information has to be given to the algorithm by the user. To overcome this database shortcoming, many researchers have proposed different methodologies. Different researchers have proposed different ideas including object oriented data storage approach, adoption of heap structures for the minimization of storage spaces. These methods have some shortcomings like execution time etc. In an attempt to increase the performance of the algorithm in real time, researchers proposed a reduction in search area by putting in a diagonal between the source and destination nodes. This algorithm however, found the shortest and most optimized path but was considered faulty because of its reduced efficiency [1] - [6].Therefore, we are proposing a Multi layered Dictionary Based Data Storage Algorithm for single and multiple Robots along with collision avoidance.

II. DIJKSTRA'S ALGORITHM.

Dijkstra's algorithm is a graph search algorithm used in computer networking and Robot path planning to find the shortest path between the starting point and the destination when there are multiple paths available for the data packet or the Robot to reach its destination point. It is also used to find the shortest path from one node to all other nodes. The shortest path or distance is known as link distance. In the algorithm a graph is denoted as G while the starting and ending nodes are represented as u and v respectively. The two nodes are together are known as edge which is represented as E(u,v) and the weight of an edge is denoted as W(u,v). Three variables are initialized in the start which are distance dist, a vector Q which stores all nodes to be visited from the starting point and a vector S which indicates about the nodes that have been visited by the algorithm. Distance of the staring node is initialized as dist (snode) = 0 and distance of all other nodes is initialized as dist (v) = ∞. The distance to each node v is updated when the shortest distance is found by the algorithm. The weight of an edge is taken from the graph.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 1, January 2021

DOI 10.17148/IJARCCE.2021.10113

III. DIJKSTRA'S ALGORITHM WITH MULTI –LAYER DICTIONARIES

Dijkstra's Algorithm comprising of multi-layered dictionaries contains a couple of dictionaries and a categorically arranged data structure list where the mapping of each node to its adjacent node is done by first dictionary while the second dictionary keeps track of the information of every adjoining pathway. G(u,v) gives the information of the path between u and v where G is the floor layout or map of an indoor location and u and v are the two adjoining nodes. To calculate the shortest distance between two locations, multi-layered dictionary algorithm is applied to a floor plan as in Figure 1. The information of the target locations labelled from 1 to 79 are considered as nodes and the distance of each two linked nodes is considered as sedges of the graph. The angles for the movement of nodes can be pre-calculated and fed in dictionary as the locations are already known as in figure 2.

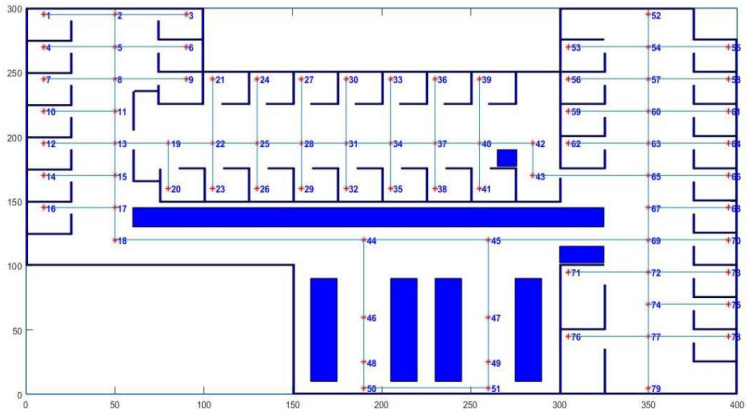

Figure 1: Floor Plan

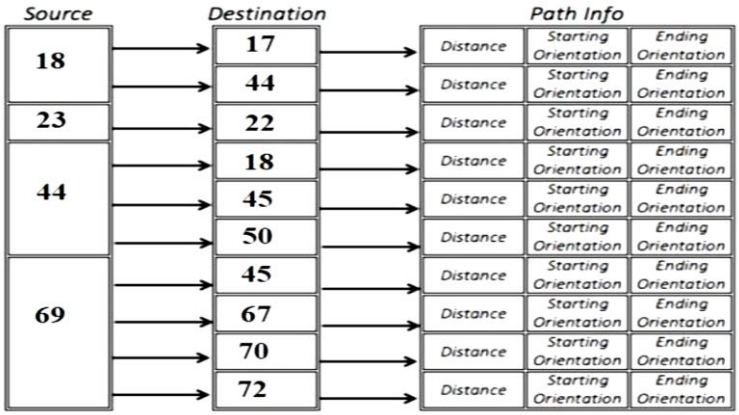

Figure 2: A sample of the Multi-layer dictionary representing the floor plan.

The primary info of the data acquired from the plan is shown in Figure 2. The info consists of three columns where the first one represents the first dictionary and contains all the nodes of the graph, second one serves as second dictionary and comprises of the edges of graphs and the last column is an arranged list of distance between start and end node and starting position and ending position of adjoining nodes. Together the 1st and 2nd column create the 1st dictionary whereas the 2nd and 3rd columns create the 2nd dictionary i.e. also a list [6]. To compute the shortest distance amid two pre-defined positions multi-layered dictionary algorithm was used and the corresponding results were implemented on a Robotic controller to move the Robot from one point to another by moving either straight, right, left or turn using the information based on its results. The commands guided the Robot towards the final position by being implemented at each node. The linking of various edges is labeled as path and given as

$$path(x,y) = min \sum (W(u,v) + W(DoD)) \qquad (1)$$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 1, January 2021

DOI 10.17148/IJARCCE.2021.10113

Where the best route between two points x and y is shown by path (x,y) and degree of rotation of an edge of two adjoining nodes is represented by weight function W(DoD) which represents the difficulty level of the path.

Dijkstra's and multi-layered dictionary algorithm for single Robot is implemented for Robot path planning on a floor plan as shown in the figure 3. There are 79 nodes plotted on the map for the movement of the Robot with several obstacles in between. The Robot can only move in straight paths rather than diagonals. The results of both algorithms are assessed on the total degree of turns and optimal path for the Robot is planned.

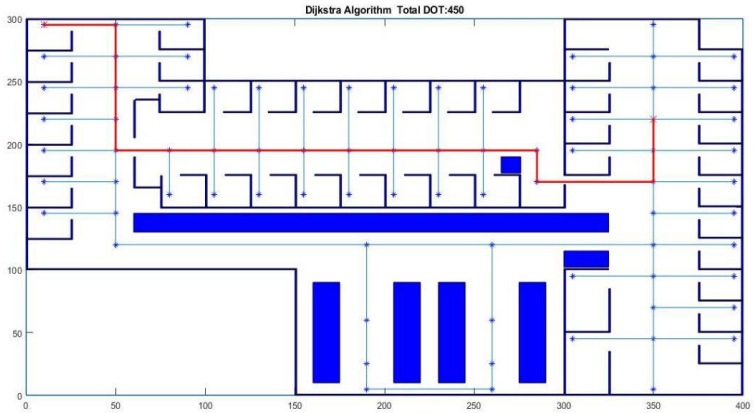

Figure 3: Implementation of Dijkstra's Algorithm on a floor plan

In figure 3, it can be seen when Dijkstra's algorithm is applied the total degree of turn for the Robot is 450 degree. The starting node of the Robot is 1 and the ending node is 60. The Robot moves from node 1 to 2, then takes a 90° turn and moves to node 13 where it takes another 90° turn and follows a straight path till node 42. The Robot makes three more 90° turns at nodes 42, 43 and 65 to reach the ending node. It is the shortest path but not the fastest (optimal) as the Robot makes several turns to reach the destination.

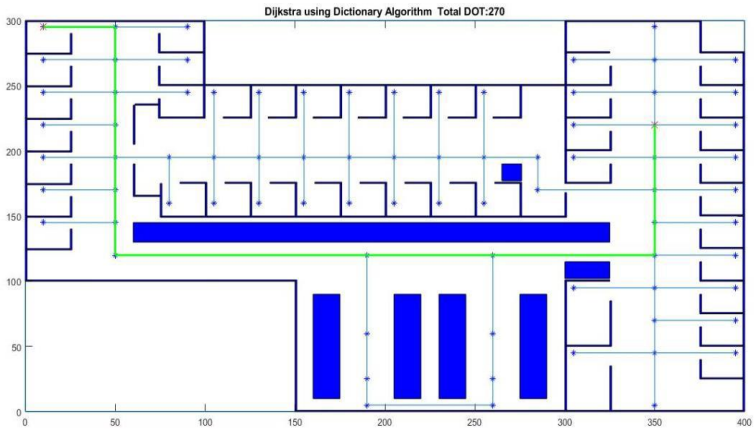

Figure 1: Dijkstra's Algorithm using Multi-layer dictionary

When implementing the multi-layer dictionary to the Dijkstra's Algorithm as done in [6], the path planned for the Robot only makes three 90° turns at node 2, 18 and 69 to reach the end point. The route planned by this algorithm is the fastest and optimal route as the total degree of turns is just 270 °. Comparison between Dijkstra's algorithm and Dijkstra's algorithm with multi-layer dictionary when single Robot is navigating in the environment is shown in table 1. It can be seen from the table that the degree of turns is greater for Dijkstra's algorithm compared with algorithm using dictionary for paths 1-60 and 10-79 but the path is same for node 10-69 for both algorithms.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 1, January 2021

DOI 10.17148/IJARCCE.2021.10113

Table I: Comparison of Dijkstra's and multi-layer dictionary algorithm for a single Robot

Starting and Ending Nodes	Dijkstra's Algorithm	Dijkstra's Algorithm using Multi-layer Dictionary
1 to 60	Shortest Path: 1, 2, 13, 42, 43, 65, 60 Orientations: Straight, Right, Straight, Left, Straight, Right, Straight, Left, Straight, Left, Straight Total degree of turns: 450°	Optimal Path: 1, 2, 18, 69, 60 Orientations: Straight, Right, Straight, Left, Straight, Left, Straight Total degree of turns: 270°
10 to 79	Shortest Path: 10, 11, 13, 42, 43, 65, 79 Orientations: Straight, Right, Straight, Left, Straight, Right, Straight, Left, Straight, Right, Straight Total degree of turns: 450°	Optimal Path: 10, 11, 18, 69, 79 Orientations: Straight, Right, Straight, Left, Straight, Right, Straight Total degree of turns: 270°
10 to 69	Shortest Path: 10, 11, 18, 69 Orientations: Straight, Right, Straight, Left, Straight Total degree of turns: 180°	Optimal Path: 10, 11, 18, 69 Orientations: Straight, Right, Straight, Left, Straight Total degree of turns: 180°

IV. NAVIGATING MULTI-ROBOTS ON THE FLOOR MAP

Dijkstra's Algorithm using Multi-layer Dictionaries are also used in scenarios where several Robots are present and moving simultaneously on the floor map. The algorithm used for single Robot does not have to prevent collision as there are no distractors or other Robots in its path. The problem in multiple Robots is that there are several Robots navigating simultaneously and if one Robot has starting node of 1 and ending node of 71 and other Robot has starting node of 71 and ending node of 1, then according to the algorithm the optimal path is chosen for both Robots will be the same. When both Robots have the same path the problem of collision will arise. To overcome this problem slight changes have been made in the multi-layer dictionary algorithm.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 1, January 2021

DOI 10.17148/IJARCCE.2021.10113

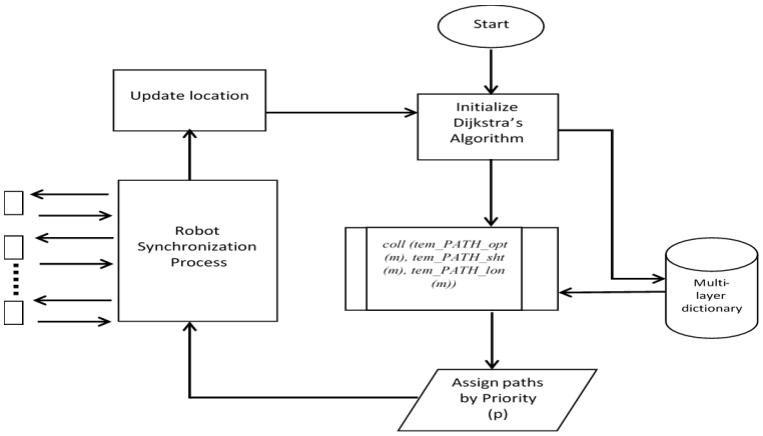

Figure 5: Flowchart for navigating Multi-Robots on the floor map

A function coll has been added in the single Robot algorithm to prevent collision as shown in figure 5. This function stores all the available paths between two nodes for all Robots i.e. (i) Optimal path (ii) Shortest path and (iii) Longer path. Priority P is also assigned to each Robot, to assign optimal paths when there is a case of collision. The function checks whether the optimal paths of all Robots are different from others or not. If the optimal path of each Robot is different from others, then the Robots are assigned there optimal paths. If two Robots have the same optimal path then one Robot is assigned the optimal path while the other Robot is assigned the shortest path according to their priority. If the optimal and shortest path are similar then the third available longer path is assigned a one Robot based on their set priority.

V. IMPLEMENTATION OF DIJKSTRA'S ALGORITHM WITH MULTI- LAYER DICTIONARY FOR MULTIPLE ROBOT.

Results have been computed for two Robots and three Robots navigating simultaneously in the environment. It can be seen from the results that multi-layer dictionary algorithm is able to compute the optimal path for all the Robots and also prevents collision.

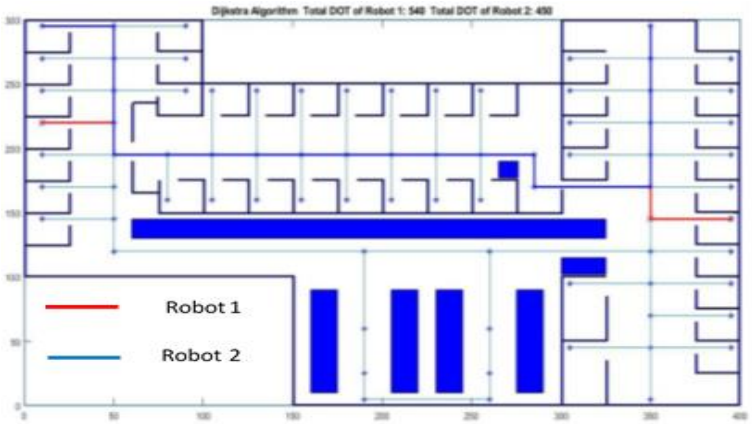

Figure 6: results for two Robots working simultaneously using only Dijkstra's algorithm

Dijkstra's algorithm is applied in figure 6 if there are two Robots present in the environment. It can be seen that Robot 1(red), which's starting node is 10 and ending node is 68, makes six 90° turns at nodes 11, 13, 42, 43, 65 and 67. The total degree of turns for Robot 1 is 540 °. Similarly, Robot 2 (blue) having a start and end node 1 and 52 respectively, makes five 90° turns at nodes 2, 13, 42, 43 and 65 with total degree of turns of 450 °.

To find the optimal paths for the Robots, Dijkstra's algorithm with dictionary is applied as shown in figure 7 for two Robots. The degree of turns for Robot 1 (red) and Robot 2 (blue) is decreased to 360° and 270° respectively.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 1, January 2021

DOI 10.17148/IJARCCE.2021.10113

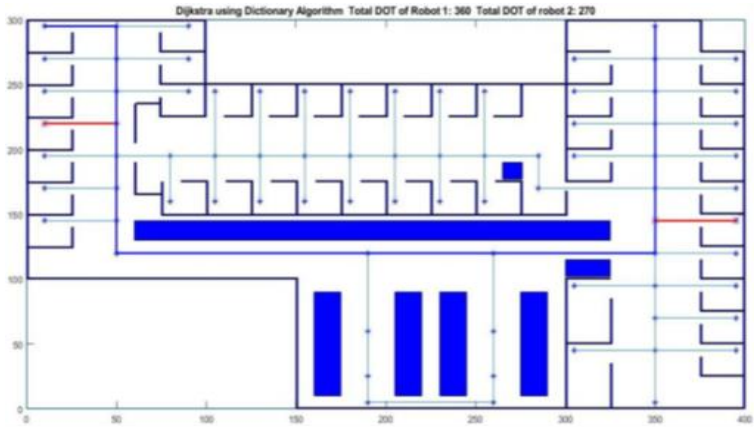

Figure 7: results for two Robots working simultaneously using Dijkstra's algorithm with multi-layer dictionary.

Robot 1 takes 90° turns at nodes 11, 18, 69 and 67 while Robot 2 makes 90° turns at nodes 2, 18 and 69. These paths are the fastest routes to reach the ending node. Table 2 illustrates the Comparison between applying only Dijkstra's algorithm and Dijkstra's algorithm with multi-layer dictionary for two Robots.

Table 2: Comparison of Dijkstra's and multi-layer dictionary algorithm for two Robots

Starting & Ending Nodes	Dijkstra's Algorithm	Dijkstra's Algorithm using multi-layer dictionary
Robot 1: 10 to 68 Robot 2: 1 to 52	Total degree of turns: Robot 1: 540° Robot 2: 450°	Total degree of turns: Robot 1: 360° Robot 2: 270°

VI. CONCLUSION

From the results that were collected during the experimentation and simulation, it was clear that the proposed method had a more sustainable and reliable outcome than the conventional Dijkstra's Algorithm. The proposed method offered a less degree of turns thus saving more time and energy consumption when it comes to navigating a single robot in a known indoor environment. As for multi-Robots, the navigation for each robot was priorities in the proposed method, therefore assigning each robot to path where it is clear to navigate which eliminates the chance of collision between robots while navigating from source to destination.

ACKNOWLEDGMENT

This paper and the research behind it would not have been possible without the exceptional support of CREIM (Centre for Research Excellence and Incubation Management UniSZA).

REFERENCES

[1] B. Madhevan and M. Sreekumar, "Identification of probabilistic approaches and map-based navigation in motion planning for mobile Robots," *Sādhanā*, vol. 43, no. 1, pp. 1–18, 2018.

[2] Kai Sun, Yuanlong Yu, and J. Gu, "Efficient robot navigation for semistructured indoor storehouse," in *proceeding of 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Halifax, NS, 2015, pp. 1313-1317.

[3] E. D. Mahmut DİRİK, Adnan Fatih KOCAMAZ, "Static Path Planning Based on Visual Servoing via Fuzzy Logic," *25th Signal Process. Commun. Appl. Conf.*, pp. 1–4, 2017.

[4] J. Liang and C. Lee, "Efficient collision-free path-planning of multiple mobile Robots system using efficient artificial bee colony algorithm," *Adv. Eng. Softw.*, vol. 79, pp. 47–56, 2015.

[5] H. D. P. Qiao, "(2014) Pigeon-inspired optimization: a newswarm intelligence optimizer for air Robot path planning," *Int. J. Intell. Comput. Cybern.*, vol. 7, no. 1, pp. 24–37, 2014.

[6] Fadzli, S., Abdulkadir, S., Makhtar, M., & Jamal, A. (2015). Robotic Indoor Path Planning Using Dijkstra's Algorithm with Multi-Layer Dictionaries. *2015 2Nd International Conference On Information Science And Security (ICISS)*. doi: 10.1109/icissec.2015.7371031.