International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Artificial Intelligence Approach for Optimal Path Planning

V Vishnu Vardhan

Student, Department of Information Technology, GMR Institute of Technology

ABSTRACT:

Building intelligent machines that can carry out tasks that traditionally require human intelligence is the focus of the broad field of artificial intelligence in computer science. Intelligence means here, they solve tasks that normally require intelligence, such as decision-making, Searching, and problem-solving. Artificial intelligence depends heavily on search, and the complexity of the search algorithm used in the inner loops of most AI systems affects how well they work. In this study, an expanded Dijkstra algorithm is discussed. The Dijkstra algorithm has been the subject of in-depth research due to its promising future. Few of them, nevertheless, have looked into how mobile robots plan their surface paths. We simulate the surface environment using the Delaunay triangulation. The triangular mesh on the surface is equivalently transformed into a triangle on the two-dimensional plane based on keeping the triangle side length constant. We can determine the shortest way by traversing every two-dimensional, developable, passable path on the surface. As a result of this study, a detailed comparison and visualization of the search techniques will be showcased.

Keywords: **Artificial Intelligence, Search Algorithm, optimal path, Dijkstra algorithm, mobile robots, Delaunay triangulation**.

I. INTRODUCTION

The Dijkstra algorithm was first introduced by Dutch computer scientist Edsger Wybe Dijkstra in 1959. In studies on the path planning of 2D mobile robots, the Dijkstra algorithm is a traditional and well-known shortest-path routing algorithm. The single-source shortest path problem has a straightforward approach that can efficiently determine the shortest route to every destination. This has been successfully applied in fields like mobile robot 2D path planning, geographic information science, computer science, transportation, etc.

In order to take into account many targets, the Dijkstra algorithm was expanded, and the weights between neighboring nodes were set as a linear combination of several weights. Each weight has a corresponding aim. These techniques allowed the Dijkstra algorithm to be expanded to multi-objective challenges and were used to calculate a three-dimensional optimal path. There are three key methods for enhancing the conventional Dijkstra algorithm. One is to examine and reduce the algorithms' space complexity in order to increase storage effectiveness and conserve space. The second is to evaluate and reduce the algorithm's time complexity. The third component is expanding the algorithm's application area and enhancing its application field by using the method in various fields. Numerous real-world issues, including field rescue and material transportation route planning, planetary ground exploration and development path planning, 3D gaming ground travel and war path planning, etc., can be abstracted and transformed into the optimal surface path problem of mobile robots. The drawback of using the conventional Dijkstra algorithm directly in surface path planning is that each intermediate path weight is calculated based on the Euclidean distance between neighboring nodes, which introduces mistakes in the surface path planning that cannot be disregarded. One of the most important causes of the surface's optimum path mistake is the possibility that the path that results from computing the Euclidean distance between nodes is not optimal. All of these indicate that the surface optimum path planning procedure based on the conventional Dijkstra algorithm needs additional tools and techniques to be improved.

In contrast to the traditional Dijkstra model, this study aims goal of this study is to build a new general solution approach with higher precision that is more suited for finding the surface's optimal path.

II. LITERATURE SURVEY

In Paper[1] min Luo, Xiaodong hou, and jing yang proposed an extended Dijkstra algorithm approach to solve optimal path planning. The Traditional Dijkstra algorithm uses the Euclidean distance algorithm to calculate path length between the nodes which produces errors for the optimal path of a surface. When researching the optimal path surface the inaccuracy of the traditional Dijkstra's algorithm can be reduced by extended Dijkstra's algorithm. The Dijkstra algorithm follows a straightforward process. Finding the shortest path between two nodes on a digraph D = (N, W), where N is the set of all nodes and W is the set of weighted edges of connected nodes, is the core task of the traditional Dijkstra method. Based on the invariance of the triangle side length for the surface, the extended Dijkstra algorithm is an algorithm to convert a surface terrain map into a 2D map along the paths that can be traversed grid map using Delaunay triangulation. The shorter, smoother method discovered in this work comes at a higher time cost. In the future, we will integrate the work of decreasing time cost in the literature to further investigate the method of reducing the time cost of this algorithm, making the algorithm more rapidly and conveniently usable in actual research.

In Paper[2] Jiankun Wang and Max Q.-H. Meng proposed a unique nonuniform sampling approach for the GVG and multiple potential functions(MPF) to plan the best possible path. In order to speed up the path planning process even further, the nonuniform sampling method concentrates on the region where the ideal path might exist.

Firstly, the environment gap is initialized with the generalized Voronoi graph and a heuristic path is calculated. Secondly, the heuristic path is discretized to create multiple potential functions(MPF). The MPF then gives the path planner a non-uniform sample distribution. The time cost and path length are selected to demonstrate the effectiveness of the algorithm. This proposed algorithm achieves convincing performance when compared with the state-of-the-art path planning algorithms.

In Paper[3] Jiankun Wang , Wenzheng Chi, Chenming Li, Chaoqun Wang , and Max Q.-H. Meng proposed a novel optimal path planning algorithm known as neural RRT*(NRRT*).RRT consumes a lot of memory and time to find the optimal path. To overcome these limitations neural RRT*(NRRT*) was proposed. The NRRT* algorithm is used to achieve non-uniform sampling in the path-planning process by learning a large number of successful planning cases from the A* algorithm. The presented method is a novel sampling method that is easily adaptable to other sampling-based algorithms for better outcomes.

In Paper[4] Guo, D., Li, Cc., Yan, W. et al proposed a method for travel path planning considering EV power supply was developed. Firstly, based on real-time road conditions, a dynamic energy model of Ev was developed considering the driving energy and accessory energy. Secondly, a multi-objective travel path planning model of EVs was constructed based on power supply, taking the distance, time, energy, and charging cost as the optimization goals. Finally, the model was simulated and tested in MATLAB based on the Dijkstra shortest path algorithm using the real traffic network of a city that spans a 15 km×15 km region. The total distance in the proposed optimal route planning method increased by 1.18 percent, but energy use, charging costs, and driving time all decreased, respectively, by 11.62 percent, 41.26 percent, and 11.00 percent. This effectively lowers the cost of EV travel while also enhancing the driving experience of EVs.

In Paper[5] W. Zhou, J. Fu, H. Yan, X. Du, Y. Wang, and H. Zhou proposed a useful control algorithm is proposed by combining the backstepping technique, adaptive dynamic programming(ADP), and the event-triggered mechanism. Compared with traditional time-triggered control methods this approach can greatly reduce the communication and computational burdens. ADP estimates the utility of a state as the sum of the reward for being in that state plus the projected discounted reward of being in the next state, making it a wiser method than Direct Utility Estimation for learning the model of the environment.

III.TRADITIONAL DIJKSTRA ALGORITHM

The traditional Dijkstra algorithm is a common method for determining the single source's shortest path. Its main principle is to expand each outer layer node sequentially from the beginning point in order to get the shortest path from each vertex to each subsequent node.

The traditional Dijkstra algorithm's main goal is to identify the shortest route connecting any two nodes on a digraph D = (N, W), where N is the set of all nodes and W is the set of weighted edges of connected nodes. N is divided into the sets N_e and N_u by the Dijkstra algorithm. The shortest path to the source node, N_s, as defined by N_e, is a collection of all the end nodes. N_e in the initial step only contains N_s. The group of nodes to N_s with the unidentified shortest path is called N_u. Up until there are no more nodes in the set Nu, the nodes in N_u will be transferred to N_e in ascending order of the source node's shortest path length, N_s.

IV. THE EXTENDED DIJKSTRA ALGORITHM

The extended Dijkstra algorithm uses the invariance of the triangle side length for the surface Delaunay Triangulation grid map to convert the surface terrain map into a 2D map along the passable paths. This algorithm was used to overcome the errors caused by the traditional Dijkstra algorithm. The first step in this process is that model the surface environment by using Delaunay triangulation. A Delaunay triangulation of a vertex set is a triangulation of the vertex set that has the property that no vertex in the vertex set lies in the interior of any triangle's circumcircle (the circle that travels through all three vertices). The main aim of the extended Dijkstra's algorithm is to find the shortest path on a surface Diagraph.

$D_{sg} = (N_{sg}, W_{sg})$

Where N_{sg} is the set of all Nodes and W_{sg} is the set of weighted edges of the connected nodes.

After that, we'll explain how to transform a triangle on a surface into a triangle on a two-dimensional plane by using the invariant principle of triangular side length. The triangular meshes of the curved surface can be sequentially transformed into triangles on the two-dimensional plane one at a time, according to Extended Dijkstra's algorithm, while maintaining the length of each side of the triangle unchanged, to achieve an equivalent passable channel on the two-dimensional plane.

V. VERIFICATION OF A MATLAB SIMULATION

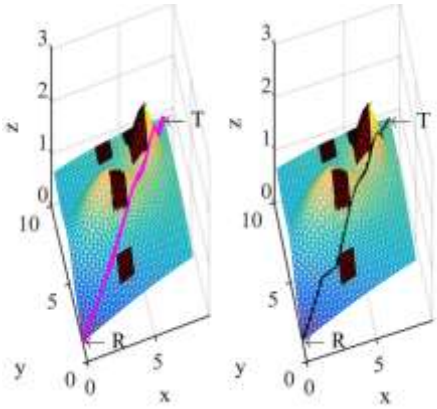
The MATLAB examples in this section will further demonstrate the effectiveness of this extended Dijkstra algorithm. First, surface path planning simulation tests with a single robot and a single target are performed using the extended Dijkstra algorithm and the results are compared to those obtained using the traditional Dijkstra algorithm. Additionally, simulation tests with several robots and many targets are run. In Matlab Simulation the red polygons

indicate the obstacles. The robot is indicated with R and the Target is indicated with T. The magenta color indicates the extended Dijkstra's algorithm surface optimal path curve and the black color indicates the traditional Dijkstra's algorithm surface optimal path curve.

A. 1-ROBOT 1-TARGET SURFACE PATH PLANNING

In this, we perform 1-robot 1-target surface path planning Matlab simulation experiments with a large number of nodes. To build the surface environment we use the Delaunay triangulation algorithm with the required resolution. We do a 1089 nodes surface simulation experiment. The initial position of the robot is [0,1,0.120] and the coordinate position of the target is [8,9,1.90].

The magenta color indicates the optimal path calculated by the extended Dijkstra's algorithm and this curve is smoother when compared with the traditional Dijkstra's algorithm. The optimal path length calculated by the magenta color is about 11.182 which is shorter than the black color with 12.582. And the maximum difference rate between the traditional Dijkstra algorithm and the extended Dijkstra algorithm is about 22.96%.



B. MULTI-ROBOT MULTI-TARGET SURFACE PATH PLANNING

In this, we perform multi-robot multi-target surface path planning Matlab simulation experiments. We do a 625 nodes 2-robot 1-target surface simulation experiment. The initial positions of Robot's R1 and R2 are [0,1,0.569] and [2,5.25,0.789] and the coordinate position of the target is [6,7,0.742]. The magenta color from R1 to T is about 8.956 and the magenta color from R2 to T is about 4.398 and the black color curve length from R1 to T is about 11.853 and the black color curve length from R2 to T is about 4.732. Therefore from the observed values, we can say that optimal paths obtained by the extended Dijkstra's algorithm are shorter than the traditional Dijkstra's algorithm.

VI. METHODOLOGY

The extended Dijkstra algorithm is presented in this section as a solution to the surface optimal path planning problem. Based on the invariance of the triangle side length for the surface Delaunay Triangulation grid map, the extended Dijkstra method is an approach to transforming the surface terrain map into a 2D map along the passable paths. The first thing we'll do is use the Delaunay triangulation method to create a surface map. One of the most popular techniques for modeling triangular meshes is Delaunay triangulation. The Delaunay triangulation algorithm can produce a smoother path and more accurate surface information when compared to a square grid map. It is easier and more practical than the digital point cloud approach. Each triangle unit has 12 adjacent units according to the Delaunay triangle mesh subdivision algorithm's equation. Therefore, this method may provide a smooth motion planning of the path because one non-boundary node can supply 12 viable motion directions. The Delaunay triangle grid map method is a form of map representation that uses a triangular mesh as the cartographic unit and effectively expresses the surface fluctuation properties. The size of the triangle mesh affects the surface map's precision and resolution. The accuracy increases with mesh size.

Finding the shortest path on a surface digraph Dsg = (Nsg, Wsg), where Nsg is the set of all nodes and Wsg is the set of weighted edges of connected nodes, is the main goal of the extended Dijkstra algorithm. Then, we'll explain how to transform a triangle on a surface into a triangle in a two-dimensional plane by using the invariant principle of triangular side length.

VII. RESULTS AND DISCUSSIONS

The MATLAB examples in this section will further demonstrate the effectiveness of this extended Dijkstra algorithm. First, surface path planning simulation tests with a single robot and a single target are performed using the extended Dijkstra algorithm and the results are compared to those obtained using the traditional Dijkstra algorithm. Additionally, simulation tests with several robots and many targets are run. In Matlab Simulation the red polygons indicate the obstacles. The robot is indicated with R and the Target is indicated with T. The magenta color indicates the extended Dijkstra's algorithm surface optimal path curve and the black color indicates the traditional Dijkstra's algorithm surface optimal path curve.

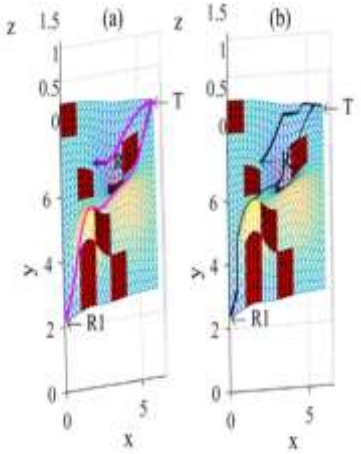


FIGURE 7. 625 nodes 2-robot 1-target example. (a) The extended dijkstra algorithm method. (b) The traditional Dijkstra algorithm method.

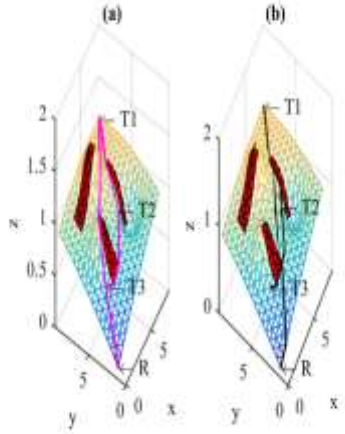


FIGURE 8. 289 nodes 1-robot 3-target example. (a) The extended Dijkstra algorithm method. (b) The traditional Dijkstra algorithm method.

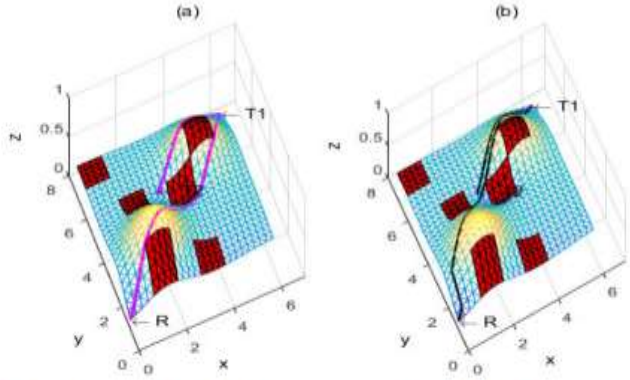


FIGURE 9. 625 nodes 1-robot 2-target example. (a) The extended Dijkstra algorithm method. (b) The traditional Dijkstra algorithm method.

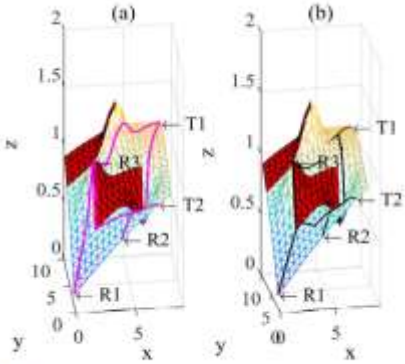


FIGURE 10. 289 nodes 3-robot 2-target example. (a) The extended Dijkstra algorithm method. (b) The traditional Dijkstra algorithm method.

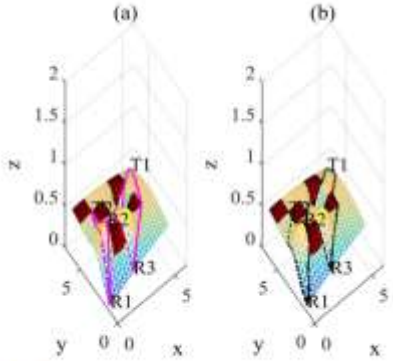


FIGURE 11. Comparion graph of 4225 nodes example. (a) The extended Dijkstra algorithm method. (b) The traditional Dijkstra algorithm method.

Table. Multi-Robot Multi-target surface optimal path comparison

Figure number	Node number	Robot number &Target number	Robot initial position	Target position	Extended Dijkstra	Traditional Dijkstra
Fig. 7	625	$R=2, T=1$	$p_{r1}=[0,1,0.569]$ $p_{r2}=[2,5.25,0.789]$	$p_{t1}=[6,7,0.742]$	$op11=8.956, op21=4.398$	$op11=11.853, op21=4.732$
Fig. 8	289	$R=1, T=3$	$p_{r1}=[0,1,0.120]$	$p_{t1}=[8,9,1.190]$ $p_{t2}=[5,4.5,0.897]$ $p_{t3}=[1.5,3.5,0.589]$	$op11=11.858, op_{t1t2}=5.615$ $op_{t2t3}=4.224$	$op11=14.626, op_{t1t2}=6.398$ $op_{t2t3}=4.572$
Fig. 9	625	$R=1, T=2$	$p_{r1}=[0,1,0.118]$	$p_{t1}=[6,7,0.263]$ $p_{t2}=[2.5,4.75,0.335]$	$op11=8.952, op_{t1t2}=4.266$	$op11=11.256, op_{t1t2}=5.445$
Fig. 10	289	$R=3, T=2$	$p_{r1}=[0,1,0.120]$ $p_{r2}=[4,1,0.575]$ $p_{r3}=[2,5,1.073]$	$p_{t1}=[8,9,1.190]$ $p_{t2}=[7,1,0.860]$	$op11=11.858, op21=9.255$ $op32=7.500$	$op11=14.448, op21=11.554$ $op32=7.906$
Fig. 11	441	$R=3, T=2$	$p_{r1}=[0,1,0.092]$ $p_{r2}=[1,3,0.703]$ $p_{r3}=[2,1,0.269]$	$p_{t1}=[5,6,0.502]$ $p_{t2}=[1,5,0.579]$	$op11=7.536, op21=5.203$ $op31=6.024, op12=4.211$ $op22=2.010, op32=3.455$	$op11=8.512, op21=6.011$ $op31=7.723, op12=5.298$ $op22=2.010, op32=4.455$

In this, we perform multi-robot multi-target surface path planning Matlab simulation experiments. We do a 625 nodes 2-robot 1-target surface simulation experiment. The initial positions of Robot's R1 and R2 are [0,1,0.569] and [2,5.25,0.789] and the coordinate position of the target is [6,7,0.742]. The magenta color from R1 to T is about 8.956 and the magenta color from R2 to T is about 4.398 and the black color curve length from R1 to T is about 11.853 and the black color curve length from R2 to T is about 4.732. Therefore from the observed values, we can say that optimal paths obtained by the extended Dijkstra's algorithm are shorter than the traditional Dijkstra's algorithm.

VIII CONCLUSION

We propose the extended Dijkstra algorithm to reduce the inaccuracy of the traditional Dijkstra algorithm when investigating the surface optimal path task. The fact that the classic Dijkstra algorithm utilizes the Euclidean distance technique to determine the path length between neighboring nodes is the cause of its mistake in studying the optimal path task of the surface. This is not an issue for the optimal path of a two-dimensional plane, but it will result in inaccuracies for the optimal path of a surface. Nowadays, the ideal surface path is sought after in an increasing number of application cases. To adapt the classic optimal path algorithm to the demands of the new era, more study is required. This method increased the Dijkstra algorithm's field of study to the surface path planning area in comparison to the standard Dijkstra algorithm. Both the single-robot, single-target project and the multi-robot, multi-target project can benefit from the improved algorithm. Additionally, as shown by the several MATLAB simulation examples in complicated surface settings, the extended Dijkstra method can gain more precise and briefer surface optimal pathways than the classic Dijkstra algorithm.

IX REFERENCES.

[1] M. Luo, X. Hou, and J. Yang, "Surface Optimal Path Planning Using an Extended Dijkstra Algorithm," in IEEE Access, vol. 8, pp. 147827-147838, 2020, doi:10.1109/ACCESS.2020.3015976.

[2] J. Wang and M. Q. . -H. Meng, "Optimal Path Planning Using Generalized Voronoi Graph and Multiple Potential Functions," in IEEE Transactions on Industrial Electronics, vol. 67, no. 12, pp. 10621-10630, Dec. 2020, doi: 10.1109/TIE.2019.2962425

[3]J. Wang, W. Chi, C. Li, C. Wang and M. Q. . -H. Meng, "Neural RRT*: Learning-Based Optimal Path Planning," in IEEE Transactions on Automation Science and Engineering, vol. 17, no. 4, pp. 1748-1758, Oct. 2020, DOI: 10.1109/TASE.2020.2976560

[4]Guo, D., Li, Cc., Yan, W. et al. "Optimal path planning method of electric vehicles considering power supply". J. Cent.South Univ. 29, 331–345 (2022). https://doi.org/10.1007/s11771- 022-4924-x.

[5]. W. Zhou, J. Fu, H. Yan, X. Du, Y. Wang, and H. Zhou, "Event-Triggered Approximate Optimal Path-Following Control for Unmanned Surface Vehicles With State Constraints," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2021.3090054.

[6] A. Alyasin, E. I. Abbas, and S. D. Hasan, ''An efficient optimal path finding for mobile robot based on dijkstra method,'' in Proc. 4th Scientific Int. Conf. Najaf (SICN), Al-Najef, Iraq, Apr. 2019, pp. 11–14, doi: 10.1109/SICN47020.2019.9019345.

[7] F. R. Souza, T. R. Cámara, V. F. N. Torres, B. Nader, and R. Galery, ''Mine fleet cost evaluation Dijkstra's optimized path,'' REM—Int. Eng. J., vol. 72, no. 2, pp. 321–328, Jun. 2019, doi: 10.1590/ 0370-44672018720124.

[8] Yujin and G. Xiaoxue, ''Optimal route planning of parking lot based on dijkstra algorithm,'' in Proc. Int. Conf. Robots Intell. Syst. (ICRIS), Huai'an, China, Oct. 2017, pp. 221–224, doi: 10.1109/ICRIS. 2017.62.

[9] W. Fink, V. R. Baker, A. J.-W. Brooks, M. Flammia, J. M. Dohm, andM. A. Tarbell, ‘‘Globally optimal rover traverse planning in 3D using Dijkstra’s algorithm for multi-objective deployment scenarios,’’ Planet. Space Sci., vol. 179, pp. 1–9, Dec. 2019, doi: 10.1016/j.pss.2019.104707.

[10] D. Guo, J. Wang, J. B. Zhao, F. Sun, S. Gao, C. D. Li, M. H. Li, and C. C. Li, ‘‘A vehicle path planning method based on a dynamic traffic network that considers fuel consumption and emissions,’’ Sci. Total Environ., vol. 663, pp. 935–943, May 2019, doi: 10.1016/j.scitotenv.2019.01.222.