



DIJKSTRA ALGORITHM USING UAV PATH PLANNING

¹Elaf Jirjees DHULKEFL , ²Akif DURDU , ³Hakan TERZİOĞLU

^{1,2,3}Konya Technical University, Faculty of Engineering and Natural Sciences, Department of Electrical-Electronics Engineering, Konya, TURKEY

¹elif.cercis94@gmail.com, ²adurdu@ktun.edu.tr, ³hterzioglu@ktun.edu.tr

(Geliş/Received: 05.11.2020; Kabul/Accepted in Revised Form: 24.12.2020)

ABSTRACT: The use of unmanned aerial vehicles (UAV) is increasing today. UAVs can be divided into two parts, which are remote controlled and can travel automatically due to a certain battery problem. Recent research has also focused on the development and application of new algorithms to autonomously control these vehicles and determine the shortest flight paths. Together with these researches, UAVs are used in many civil activities such as weather forecasts, environmental studies and traffic control. Three-dimensional (3D) path planning is an important issue for autonomously moving UAVs. The shortest path for Unmanned Aerial Vehicles (UAV) is determined by using two-dimensional (2D) path planning algorithms using the obstacles in the environment, and allows UAVs to perform their environmental tasks as soon as possible. The purpose of this study is to determine the shortest path to the target point and avoiding obstacles for UAVs using the Dijkstra algorithm. It was developed to evaluate the arrival time of the UAVs in the path planning algorithm with the simulation performed in the MATLAB program. In this study, the obstacles were defined for the purpose of the building with different heights and different widths and 2D and 3D models were carried out, assuming that the UAV flies at certain heights. In addition, the flight of the UAVs in the route planning determined in the real applications was carried out and the data such as battery consumption, amount of battery spent, speed, amount of travel were examined.

Key Words: UAV, 2D and 3D, path planning, Dijkstra algorithm.

DIJKSTRA ALGORİTMASI KULLANILARAK İHA YOL PLANLAMASI

ÖZ: İnsansız hava araçlarının (İHA) kullanımı günümüzde giderek artmaktadır. İHA’lar uzaktan kumandalı ve belirli bir batarya probleminden dolayı otomatik olarak seyahat edebilen olmak üzere iki kısma ayrılabilirler. Son dönemde gerçekleştirilen araştırmalar, bu araçları otonom bir şekilde kontrol etmek ve en kısa uçuş yollarını belirlemek için yeni algoritmaların geliştirilmesi ve uygulanması konularına da odaklanılmışlardır. Bu araştırmalarla birelere kullanım alanı olarak İHA’lar hava tahminleri, çevre çalışmaları ve trafik kontrolü gibi birçok sivil faaliyette kullanılmaktadır. Otonom hareket eden İHA’lar için üç boyutlu (3D) yol planlaması önemli bir konudur. İnsansız Hava Araçları (İHA) için en kısa yol, çevredeki engelleri kullanarak iki boyutlu (2D) yol planlama algoritmaları kullanılarak belirlenir ve İHA’ların çevre görevlerini mümkün olan en kısa sürede yerine getirmelerine olanak tanır. Bu çalışmanın amacı, Dijkstra algoritması kullanılarak İHA’lar için engellerden kaçınarak ve hedef noktasına giden en kısa yolu belirlemektir. MATLAB programında gerçekleştirilen simülasyon ile yol planlama algoritmasında İHA’ların hedefe varış zamanını değerlendirmek için geliştirilmiştir. Bu çalışma farklı yükseklik ve farklı enerji sahip olan bina amacıyla engeller tanımlanmış ve İHA’nın belirli yüksekliklerde uçtuğu kabul edilerek 2D ve 3D modellemeleri gerçekleştirılmıştır. Ayrıca İHA’ların gerçek

uygulamalarda belirlenen yol planlamalarında uçuşu gerçekleştirilerek pil tüketimi, harcanan pil miktarı, hızı, alınan yol miktarı gibi verilerde incelenmiştir.

Anahtar Kelimeler: İnsansız hava araçları (İHA), 2D ve 3D, yol planlama, Dijkstra algoritması.

1. INTRODUCTION

UAV is an aircraft that is not human, remote and self-controlled. Basic UAV components; the primary body is called the shell, propeller, wing, motor drive, engine, battery, control panel. In addition to these basic components, the UAV also contains electronic sensors, communication electronics, numerous detectors and cameras. The historical development of the UAV began in the mid-19th century, but during the periods of the First and Second World War it was noticeable and continued to develop through increased pilot losses as well as exploration and intelligence during the Cold War era.

The design and quality of drones used for both civil and military purposes has received great attention over the past 10 years. Use UAVs that can operate independently in vibrant and complicated operational environments has become increasingly common. UAVs need to be ready to manage emergencies during the process. Under certain constraints, the usual problem of path planning is to determine between the appropriate locations on an ideal or time-based path. After preparation of the flight plan (unmanned aerial vehicle

Capable of entering enemy threats, performing certain duties in the enemy's air defense zone and ensuring their safety). UAVs have the benefits of low cost, flexibility, high reliability and you don't have to worry about private losses (Yong-Fei *et al.*, 2013). In a complicated environment, UAVs perform their tasks, avoiding obstacles that are necessary. UAV shows two classification methods to overcome obstacles: a traditional algorithm and a smart algorithm. The main element of these autonomous flights is the path preparation. path planning is the main component of these autonomous flights. Preparation of the path shows a variety of factors, such as how to get to a destination to avoid obstacles and shorten the path (Lei *et al.*, 2017).

For UAVs path planning to date, suggested a technique of UAV path planning in complex environments based on genetic algorithms. The showed that the suggested algorithm succeeded in finding a non-optimal, unhindered path in a rapidly changing environment (Gao *et al.*, 2005). An autonomous system for planning the UAV route. Achieved by reviewing the development process and eliminating the caps and restoring old plans that could be missed by contacting a PRM (probabilistic roadmaps) or RRT (rapidly discovered random trees) planner to create new plans right now (Wzorek and Doherty, 2006). This article discusses Portsmouth Harbor, a realistic marine environment to search for the optimal computational time path for a USV between start and end points in a real-time map. In this study, a grid map created from the original was used and the Dijkstra algorithm was used to find the shortest path for a single USV (Singh *et al.*, 2018). Recommended rate if path optimization strategies for unmanned aerial vehicles (UAVs) are based on intelligence. They showed that in case of map coverage, the Ant System algorithm can be used to optimize the UAV system. When comparing another approach, NNS (nearest neighbor search) means that the algorithm is more successful in finding a better route than the nearest neighbor search. (Jevtić *et al.*, 2010). The proposed area height-based Dijkstra algorithm MDA (Modified Dijkstra Algorithm) approach for UAV fixed wing trajectory planning shows that EDA (Elevation Based Dijkstra Algorithm) significantly reduces the computation time (Medeiros and Da Silva, 2010). Approach to FVF based UAV route planning. FVF (Fuzzy Virtual Force) approach for UAV route planning is easy and fast. By comparing the FVF method with the A * search algorithm, the FVF method is superior to online UAV route planning in a complex environment (Zhuoning *et al.*, 2010). An autonomous approach to improving UAV flight plan readiness. We started Q-learning, a kind of reinforcement learning algorithm, to remove obstacles until the UAV reaches the target point. The proposed approach suggests a shorter arrival time (Kim *et al.*, 2017). PSO offered a different approach to formulate and solve problems in 3D path planning. The aim is to reduce fuel consumption and minimize the risk of enemy threats. (Foo *et al.*, 2006). They proposed 3D path algorithms based on 2D frames. They also proposed the Line of

Visibility (VL) method, which can create the shortest path from a starting point to a target point in an area of polygonal barriers. This indicates that it is effective in measurement and suitable for real time applications. (Omar and Gu, 2010). A solution approach has been presented to ensure collaborative scheduling between tool teams. A common control approach was created based on the Voronoi diagram and DPSO. Findings show that the proposed genetic algorithm can be used in time-limited complex group assignments (Tong, 2012).

These feature 3D path planning for low-flying unmanned aerial vehicles (UAVs) in difficult terrain conditions. They proposed an adaptive fluid system (IFDS) with intervention to obtain the optimal 3D orientation for runway length and flight altitude. The proposed solution showed better results (Wang *et al.*, 2015). The mobile robot design was created using the open source simulation software V-REP, and the Dijkstra algorithm was applied to determine the six-optimal and collision-free path. The simulated results showed that the mitigation approach in the created robot path planning environments is effective in terms of time and speed (Fusic *et al.*, 2018). Defined a multi-criteria path planning model for unmanned medium altitude and long aircrafts. Taking environmental dynamics into account, they proposed the A * algorithm for unmanned aerial vehicles in terms of range, time and fuel usage (Arica *et al.*, 2012). The techniques and algorithms described in this article have been extensively studied, simulated and proposed in low-level applications, as well as high-level decision algorithms and flight optimization processes, as well as UAV motion strategies in unmanned aviation projects. Intelligent techniques used in this study to optimize a UAV mission plan with sensory input (Vladareanu *et al.*, 2016). Quadrotor-type unmanned aerial vehicles (UAVs) also provide path planning to reduce the power consumption of the vehicle, which is required over time to prevent energy waste. It uses the Genetic Algorithm (GA) to determine the shortest way to fly without the need for power and time savings (Galvez *et al.*, 2014). The algorithm used by the modified Dijkstra was used to find an obstacle avoidance path that connects the starting point, the obstacle turning point and the end point. The algorithm used by the modified Dijkstra was used to find the way to avoid obstacles connecting the starting point, the obstacle turning point and the end point. The results show that not only the proposed method is applicable in road planning, but also good results can be achieved (Wu *et al.*, 2017). In this report, for data transmission, civilian UAVs are using existing 3G communication networks. The approach proposed shows that MOGA (Multi-Objective Genetic Algorithm) provides 1.32 and 3.22 times better signal performance than path-greedy and signal-greedy algorithms (Tseng, 2011).

This study is to determine the shortest path to the target point by avoiding obstacles for UAVs using the Dijkstra algorithm. It has been developed to evaluate the arrival time of the UAVs in the path planning algorithm with the simulation performed in the MATLAB program. In this study, obstacles for the purpose of the building with different dimensions were defined and 2D and 3D models were carried out, assuming that the UAV flies at certain heights. When the obtained results were evaluated, it was seen that the Dijkstra algorithm gave good results in the simulation environment. Simulation results of the area used in the actual application were created and flights were carried out with UAV. The main purpose of the route planning in UAVs is to use the battery at the optimum level and reach the target in the shortest way, the electrical parameter values of these flights have been examined and evaluated.

2. PATH PLANNING

Path planning is a strategy by measuring the distance in space between two points to find the best path between them. Research on path planning usually requires multiple input values or parameters: starting position, goal position, and obstacles. In algorithms, some steps are used to discover a 2D path. Algorithms for path planning are used to enter this path at the highest possible price, both from a starting point to a destination point. Some alternative path planning focused on algorithms for road planning. Different techniques assess the most suitable path planning (Cabreira *et al.*, 2019). The aim of path planning, unlike movement planning, which must be taken into account in terms of dynamics, is to find a cinematically optimal route with the least amount of time and to fully model the environment (Dhulkefl and Durdu, 2019; Yang *et al.*, 2016).

3. DIJKSTRA ALGORITHM

A weighted graph is an algorithm that provides the shortest distance between any two nodes with values on a weighted graph. This algorithm was invented by Dutch mathematician and computer scientist Edsger Wybe Dijkstra. Algorithm is used in many areas, especially routing. Dijkstra algorithm Greedy is an algorithm that is greedy.

That is, the dijkstra algorithm chooses the best solution of the current situation when moving from one node to another. The dijkstra algorithm, which is used in computer science and named after the person who introduced the algorithm to the literature, is used to find the shortest path in the graph below. In a series that shows the distance between two nodes, the problem is to find the route with a minimum edge length from the starting node (source node) to a specific end node (target node) (Mozaffari *et al.*, 2016).

This algorithm was developed in 1956 by Edsger W. Dijkstra. The main purpose of this algorithm is to develop a coveted method for implementation, such as shortest path planning. Heavy modification has reduced the complexity of the algorithm. The basis of this algorithm is the relaxation principle. More accurate values gradually change the correct distance approach until finally the optimal solution is reached. The Dijkstra algorithm gets its name from its constructor. The main purpose of the algorithm is to find the shortest path on Graf. Graf can confuse those who first hear the name. We have to keep the data in the computer environment with a certain logic. One of the methods we use most is to keep the data in graph format. Graphs are highly preferred in terms of showing the relations of the nodes on each other (Mozaffari *et al.*, 2016). Sample Dijkstra algorithm nodes as shown in Figure 1.

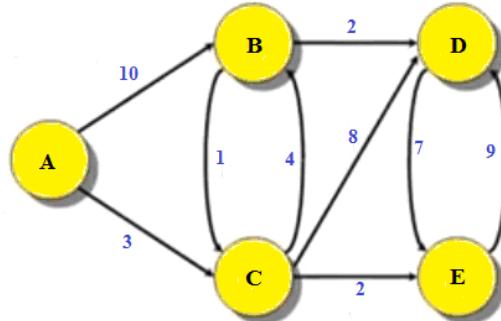


Figure 1. Example nodes for the Dijkstra algorithm

The calculation function used in this algorithm is:

$$f(n)=g(n) \quad (1)$$

As it appears in the equation,

$f(n)$: heuristic function,

$g(n)$: The cost of moving from the starting node to the current node. Flow diagram for Dijkstra algorithm is shown in Figure 2.

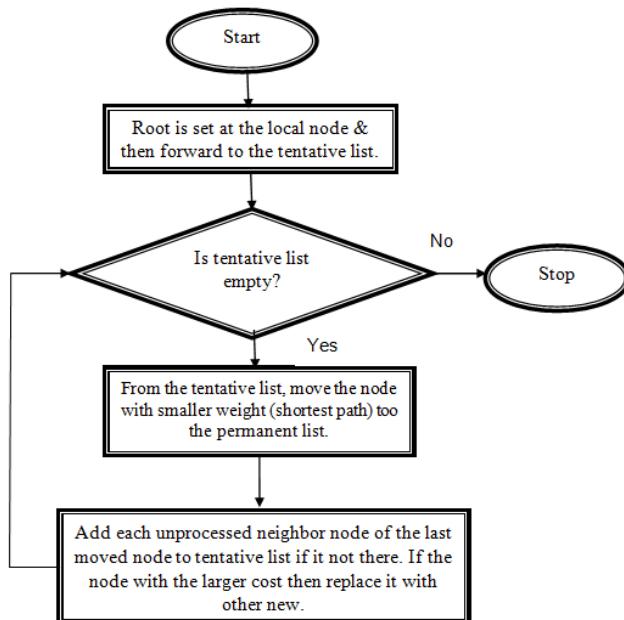


Figure 2. Flow chart for Dijkstra algorithm

3.1. Dijkstra Algorithm Operation

In the Dijkstra algorithm, a starting point is chosen. All remaining nodes should be considered as infinitely far away. We update the distances as we reach the nodes. At each step, Dijkstra looks at neighbors leaving a node, and if a shorter path is found, the distances are updated. And again to the node closest to him. In this way, all nodes on the chart are circulated and the shortest path is found. Dijkstra's Algorithm works by visiting the corners of the graph, starting from the starting point of the object. Then, he repeatedly examines the closest corner that has not been examined and adds the corners to the vertex set to be examined. It expands outward from the starting point until reaching the target. Dijkstra's Algorithm is guaranteed to find the shortest route from the starting point to the destination (Fusic *et al.*, 2018).

The Dijkstra algorithm calculates the shortest paths from one node to all other nodes. The weight value of the edges developed for weighted and directional charts must be zero or greater than zero. If the values of the edges are less than zero, Bellman-Ford can be used in more general algorithms. Because the negative edge consistently produces a better result than the current state, and the algorithm is never stable. In Dijkstra's algorithm, the shortest path from the initially selected node to all other nodes is calculated. Dijkstra's algorithm does not have a target node (Fusic *et al.*, 2018).

If a target node is identified as a problem, the Dijkstra algorithm finds the shortest path to all nodes and then selects one of them. So if you use the Dijkstra algorithm, the shortest distance to all nodes is calculated, regardless of whether you have a target or not. If you have a destination, the shortest route to that destination has already been calculated. In Dijkstra, each step is evaluated for a node's neighbors and the distances are updated if a shorter path is found (Fusic *et al.*, 2018).

To use the Dijkstra algorithm:

- Developed for weighted and directional graphs,
- The weight value of the edges must be zero or greater than zero, if the edge values are less than zero, the Bellamn-Ford algorithm can be used,
- Does not work for negative weights.

4. SIMULATION AND APPLICATION

In this article, Matlab is used to create this map. Designing 3D path planning techniques for advanced 2D path planning algorithms for autonomous mobile robots and continuing fast in practice is desirable. Unmanned aerial vehicles ensure, among the barriers created by the simulation environment, that the shortest path is acquired by avoiding obstacles between the starting point and the target and using suggested technique. The method proposed in this article is the Dijkstra algorithm.

A few steps are used in algorithms to discover a 3D path. Next, data is collected from a regional starting point to an endpoint on the heights of obstacles. The maximum height the UAV can fly to is then determined by the height of these obstacles and the shortest path planning. Many alternative path planning focused on 2D path scheduling algorithms based on obstacle heights. the proposed technique finds the shortest path between the starting point and the target point. redline is the outcome of Dijkstra algorithm path planning.

4.1. First 2D Test Medium

As shown in Figures 3, path planning algorithm are used. The proposed algorithm is the Dijkstra algorithm. As shown in the figures, there are three different obstacles, each with a different Length and width. The first and third obstacles are 20 m wide and 60 m long and the second obstacle is 40 m wide and 40 m long.

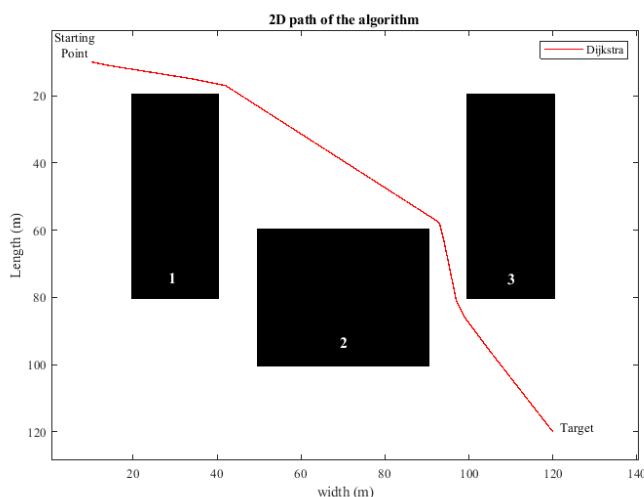


Figure 3. 2D path of the algorithm for the first application

The results of the path in the above figure are as shown in table 1. The path from the starting point to the destination is calculated in 2D. The 154m path is shown to be moving in 1.0410 seconds, according to the Dijkstra algorithm.

4.2. First 3D Test Medium

As shown in Figure 4, the first obstacle is 60 m high after assessing the drone's height and the obstacle, the second obstacle is 40 m high and the third obstacle is 60 m high. The algorithm calculates the path length when the drone's height is 45m, indicating that the second obstacle is higher than the height, thus ignoring the second obstacle and reaching the destination point in a shorter time.

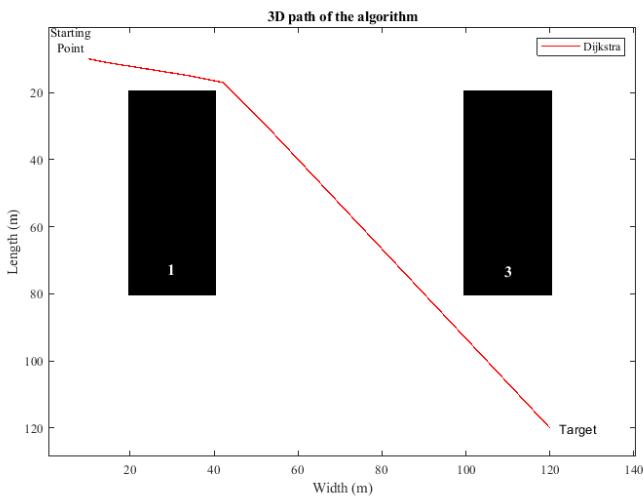


Figure 4. 3D path of the algorithm for the first application

The results of the path in the above figure are as shown in Table 1. The path from the starting point to the destination is calculated in 3D. The 22m path is shown to be moving in 1.2762 seconds, according to the Dijkstra algorithm.

4.3. Second 2D Test Medium

As shown in Figures 5, path planning algorithm are used. The proposed algorithm is the Dijkstra algorithm. As shown in the figures, there are four different obstacles, each with a different Length and width. the first obstacle are 20m wide and 50m long and the second obstacle is 20 m wide, 20m long, the third obstacle is 20m wide and 30m long and the fourth obstacle is 20m wide and 60m long.

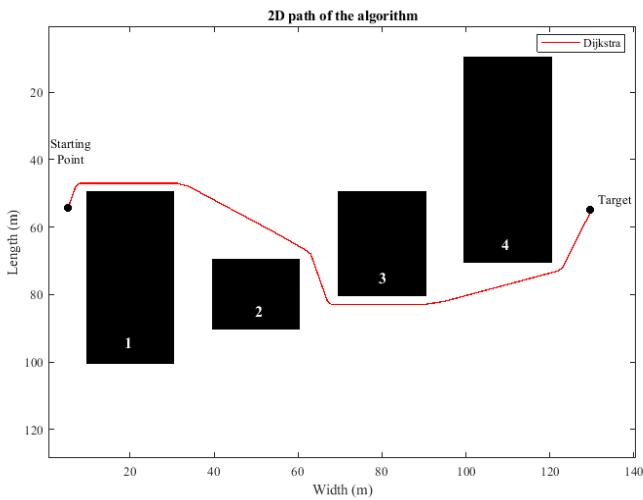


Figure 5. 2D path of the algorithm for the second application

The results of the path in the above figure are as shown in Table 1. The path from the starting point to the destination is calculated in 2D. The 73m path is shown to be moving in 1.1049 seconds, according to the Dijkstra algorithm.

4.4. Second 3D Test Medium

The algorithm calculates the path length when the drone's height is 55 m, indicating that the first, second and third obstacles is higher than the height, thus ignoring the first, second and third obstacle and reaching the destination point in a shorter time.

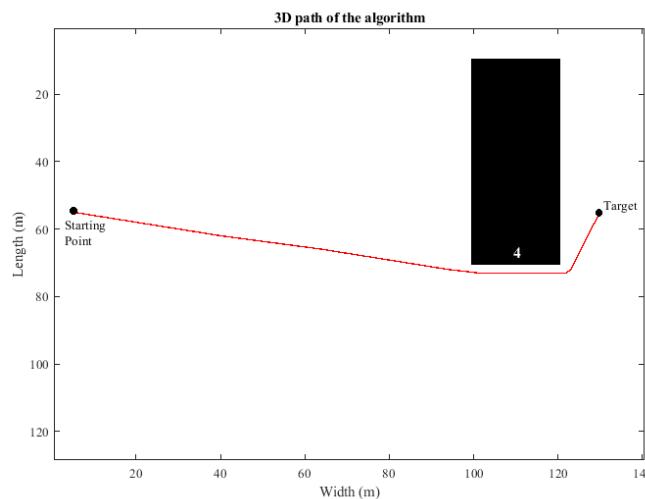


Figure 6. 3D path of the algorithm for the second application

The results of the path in the above figure are as shown in table 1. The path from the starting point to the destination is calculated in 3D. The 43m path is shown to be moving in 1.4835 seconds, according to the Dijkstra algorithm.

Table 1. 2D & 3D path planning algorithm performance per second and path planning line length performance

Figure	Dijkstra Algorithm Time (sec)	Dijkstra Algorithm Path Length (m)
First Environment	3	1.0410
	4	1.2762
Second Environment	5	1.1049
	6	1.4835

The example of Unmanned aerial vehicle in the real-time test environment is as shown in Figure 7 There are three obstacles, each obstacle has different heights. The first obstacle height is 60m, the second obstacle is 70m and the third obstacle is 80m, paths from the starting point to the destination point are calculated.

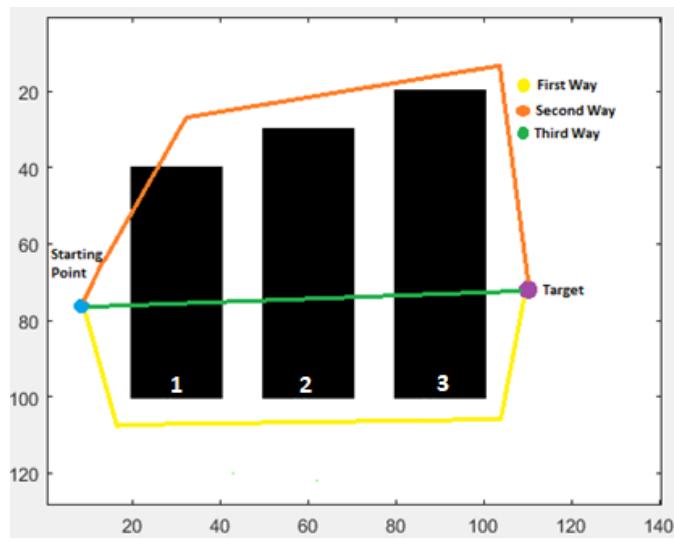


Figure 7. UAV test environment

The results of the UAV test environment are as shown in Table 2. The images on the flight path of unmanned aerial vehicles are as shown in Figure 8, Figure 10 and Figure 12.

Table 2. Energy consumption at different heights

Way	Drone Height (m)	Distance (m)	Drone Speed (m / s)	Battery Expenditure (v)	Battery consumption (%)
First Way	50	385	5	3400	23.43756
Second Way	65	387	5	3400	25.252822
Third Way	85	272	5	3400	22.265682

4.5. Planning of UAV path at different heights



Figure 8. Drone height 50m for first road

As seen in the image below, the drone has a height of 50m. When we try the first path in the simulation

environment, drone height is lower than the height of the obstacles. Therefore, the algorithm passes around obstacles when executed. It shows a path as shown in figure 9.

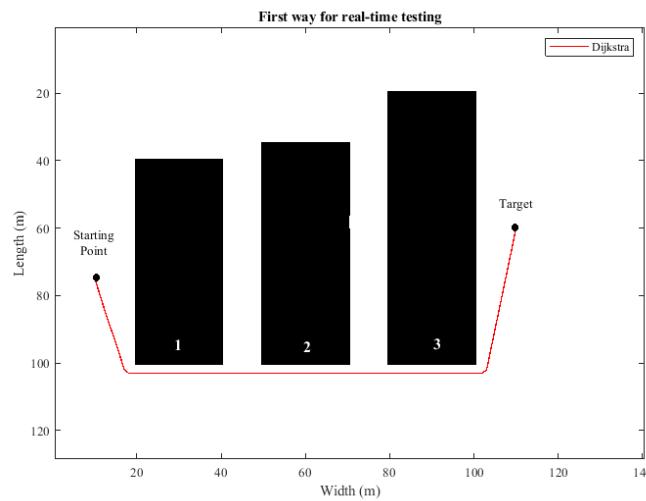


Figure 9. First way for real-time testing

The path from the starting point to the destination is calculated in 3D. The 97m path is shown to be moving in 1.0138 seconds, according to the Dijkstra algorithm.



Figure 10. Drone Height 65m for second way

When the drone height is 65m, we tried the second path in the simulation environment, when the algorithm runs, the first and second obstacles ignores the 60 m, 70 m height because the drone height is higher than the first and second obstacles. The plotted path is as shown in Figure 11 below.

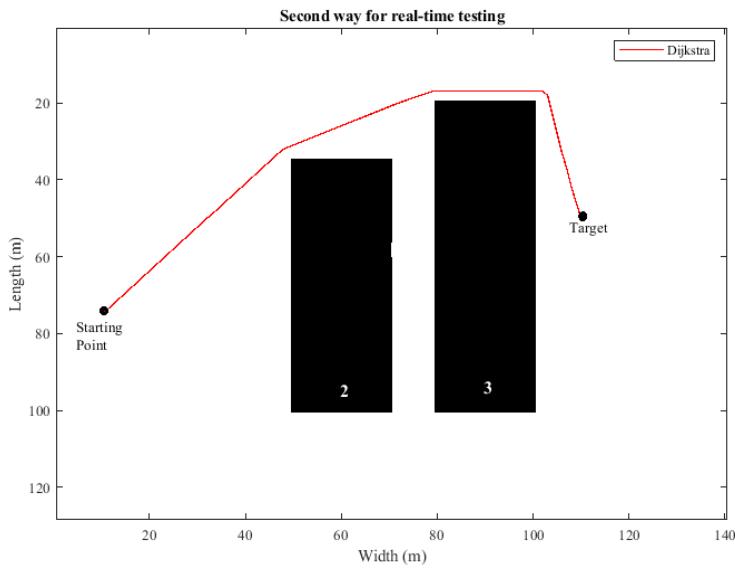


Figure 11. Second way for real-time testing

The path from the starting point to the destination is calculated in 3D. The 61m path is shown to be moving in 1.2142 seconds, according to the Dijkstra algorithm.



Figure 12. Third way for drone height 85m

When the drone height is 85m, we tried the third way in the simulation environment, since the drone height is more than the height given for three obstacles, it ignores the three obstacles when the algorithm is executed. As a result, it is calculated without any obstacles from the starting point to the destination. The plotted path is as shown in Figure 13 below.

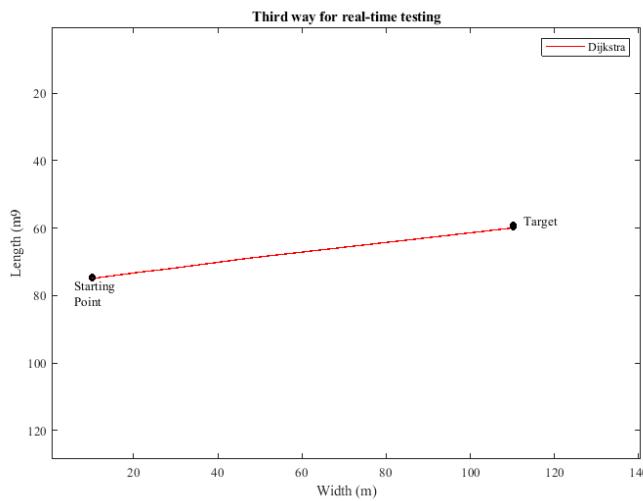


Figure 13. Third way for real-time testing

It is equal to 16.8 V when the battery is full and equal to 13.2 V when the battery is empty. According to the results, the higher the UAV, the more batteries are consumed. The results of the simulation environment are shown in Table 3 below.

Table 3. Three-dimensional simulation result for real-time testing

Three-dimensional simulation result	Dijkstra Algorithm	
	Time (sec)	Optimal path (m)
First way	1.0138	97
Second way	1.2142	61
Third way	1.4389	80

5. RESULTS AND DISCUSSIONS

When we compare the results of Dijkstra algorithm, it gives different results in both experimental environments. By comparing 2D and 3D environments, our algorithm offers better results in 3D environments. As a result, Dijkstra algorithm shows shorter path lengths in three-dimensional environments as a result, unmanned aerial vehicles are used for this purpose comparing simulation and real-time test results as shown in Table 4.

Table 4. Comparison of 2D and 3D real-time test and simulation results per second

Figure	Dijkstra Algorithm (sec)		
	Simulation	Real-Time Testing	
First Environment	3	1.0410	0.10
	4	1.2762	0.06
Second Environment	5	1.1049	0.11
	6	1.4835	0.07

REFERENCES

- Arıca, N., Cicibaş, H., & Demir, K. A. (2012). İnsansız Hava Araçları için Çok Kriterli Güzergâh Planlama Modeli. *Journal of Defense Sciences/Savunma Bilmleri Dergisi*, 11(1).
- Cabreira, T. M., Brisolara, L. B., & Ferreira Jr, P. R. (2019). Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1), 4.
- Dhulkefl, E. J., & Durdu, A. (2019). Path planning algorithms for unmanned aerial vehicles. *International Journal of Trend in Scientific Research and Development (ijtsrd)*, 359-362.
- Foo, J. L., Knutzon, J., Oliver, J., & Winer, E. (2006). Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization. Paper presented at the 11th AIAA/ISSMO multidisciplinary analysis and optimization conference.
- Fusic, S. J., Ramkumar, P., & Hariharan, K. (2018). Path planning of robot using modified dijkstra Algorithm. Paper presented at the 2018 National Power Engineering Conference (NPEC).
- Galvez, R. L., Dadios, E. P., & Bandala, A. A. (2014). Path planning for quadrotor UAV using genetic algorithm. Paper presented at the 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM).
- Gao, X.-G., Fu, X.-W., & Chen, D.-Q. (2005). A genetic-algorithm-based approach to UAV path planning problem. Paper presented at the Proceedings of the WSEAS International Conference on Simulation, Modeling, and Optimization.
- Jevtić, A., Andina, D., Jaimes, A., Gomez, J., & Jamshidi, M. (2010). Unmanned aerial vehicle route optimization using ant system algorithm. Paper presented at the 2010 5th International Conference on System of Systems Engineering.
- Kim, I., Shin, S., Wu, J., Kim, S.-D., & Kim, C.-G. (2017). Obstacle avoidance path planning for UAV using reinforcement learning under simulated environment. Paper presented at the IASER 3rd International Conference on Electronics, Electrical Engineering, Computer Science, Okinawa.
- Lei, W., LI, B.-j., YIN, Z.-h., Cheng, Z., Xin, Z., & CHU, Y.-n. (2017). An Improved Artificial Potential Field for Unmanned Aerial Vehicles Path Planning. *DEStech Transactions on Computer Science and Engineering(cst)*.
- Medeiros, F. L. L., & Da Silva, J. D. S. (2010). A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation. Paper presented at the Brazilian Symposium on Artificial Intelligence.
- Mozaffari, M., Saad, W., Bennis, M., & Debbah, M. (2016). Optimal transport theory for power-efficient deployment of unmanned aerial vehicles. Paper presented at the 2016 IEEE international conference on communications (ICC).
- Omar, R., & Gu, D. (2010). 3D Path Planning for Unmanned Aerial Vehicles using Visibility Line based Method. Paper presented at the ICINCO (1).
- Singh, Y., Sharma, S., Sutton, R., & Hatton, D. (2018). Towards use of dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field. *TransNav, International Journal on Marine Navigation and Safety od Sea Transportation*, 12(1).
- Tong, H. (2012). Path planning of UAV based on Voronoi diagram and DPSO. *Procedia Engineering*, 29, 4198-4203.
- Tseng, F.-H. (2011). Li-Der Chou, and Han-Chieh Chao.". A survey of black hole attacks in wireless mobile ad hoc networks, 1-16.
- Vladareanu, V., Boscoianu, E.-C., Sandru, O.-I., & Boscoianu, M. (2016). Development Of Intelligent Algorithms For Uav Planning And Control. *Scientific Research & Education in the Air Force-AFASES*, 1, 221-226.
- Wang, H., Lyu, W., Yao, P., Liang, X., & Liu, C. (2015). Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system. *Chinese Journal of Aeronautics*, 28(1), 229-239.

- Wu, T.-F., Tsai, P.-S., Hu, N.-T., & Chen, J.-Y. (2017). Combining turning point detection and Dijkstra's algorithm to search the shortest path. *Advances in Mechanical Engineering*, 9(2), 1687814016683353.
- Wzorek, M., & Doherty, P. (2006). Reconfigurable path planning for an autonomous unmanned aerial vehicle. Paper presented at the 2006 International Conference on Hybrid Information Technology.
- Yang, L., Qi, J., Song, D., Xiao, J., Han, J., & Xia, Y. (2016). Survey of robot 3D path planning algorithms. *Journal of Control Science and Engineering*, 2016.
- Yong-Fei, M., Luo, Z., & Luo-Sheng, X. (2013). Application of improved sparse A* algorithm in UAV path planning. *Information Technology Journal*, 12(17), 4058.
- Zhuoning, D., Rulin, Z., Zongji, C., & Rui, Z. (2010). Study on UAV path planning approach based on fuzzy virtual force. *Chinese Journal of Aeronautics*, 23(3), 341-350.