

Multi-UAV Path Planning using Modified Dijkstra's Algorithm

Dhruv Karve

Computer Engineering Department
Thadomal Shahani Engineering College
Mumbai University, Mumbai, India

Farhan Kapadia

Computer Engineering Department
Thadomal Shahani Engineering College
Mumbai University, Mumbai, India

ABSTRACT

Unmanned Aerial Vehicles (UAVs), informally referred to as drones, are ubiquitous in various fields today. The widespread applications and popularity of drones arise from the fact that these automatic devices can cover large distances efficiently and without the need of an operator on board. However, to cover these large distances in an efficient manner, an efficacious algorithm is also necessary. In order to optimize not just the distance between the source and the destination but also satisfy various other constraints such as covering the maximum area possible using the least number of drones and visiting the closest charging stations to complete the journey within the battery life, a modified version of Dijkstra's Algorithm has been used. There already exist algorithms to optimize the path of a drone but very few algorithms also take constraints such as energy cost and area maximization into account. This was the inspiration to take up this project; to devise an algorithm that satisfies the aforementioned constraints while also remaining pertinent to the main objective of multiple UAVs path optimization algorithms - distance and cost minimization.

General Terms

Algorithms, Routing, Geographic Information Systems, Multi-agent Systems.

Keywords

Path Planning, Multi-UAVs, Dijkstra's Algorithm, Drone Routing, Multi-objective Optimization, Charging Stations.

1. INTRODUCTION

A drone is an unpowered aircraft. It is a component of a system called the Unmanned Aircraft System (UAS). The UAS comprises a drone, an operator on the ground and a network of communications between the two. UAVs can either be partially autonomous or completely autonomous depending on the purpose of the task at hand. A drone embarks on its journey from a starting ground station, visits all sensors it is programmed to visit and returns to its ground station in a set period. This can be likened to how aircrafts operate. They start from the departing airport, visit a set of mission waypoints and return to the home airport to be taxied into their respective hangars [1]. In the problem statement, these waypoints are of two types- visiting nodes and charging stations. Drones are used in a variety of fields such as the military for carrying out aerial operations without risking a pilot on board, lightweight transportation, surveillance, in movies for a bird's eye view, delivery of goods and many more.

The reasons for the drone's popularity and widespread application among an array of heterogeneous fields are as follows:

- i) Aerial: Drones traverse in the air. The air does not suffer from the problems that the ground does such as heavy traffic, congestion, obstacles, etc. This is one significant advantage the drone has over its potential competitors.
- ii) Environment friendly: Drones are not driven by fuel unlike most other vehicles. The fact that it is operated purely on a battery makes it not only energy efficient but also very environment friendly.
- iii) Safety: Drones do not require a pilot or operator on board to function. They can easily be controlled from the ground or be run completely autonomously by a computer program. This reduces the risk of loss of life significantly when compared to other alternatives.

There are several important applications of drones. Some of them are as follows:

- i) Surveillance: The drone's ability to fly at incredible heights and in narrow spaces makes it an indispensable tool for remote surveillance. It can be used for surveillance in multiple areas, like military surveillance in a hilly disputed region, in offshore factories and many more places.
- ii) Lightweight delivery: A drone's capability to fly long distances without the need of a pilot makes it an excellent replacement for door to door deliveries for lightweight packages. Two tech giants in Google and Amazon have also taken initiatives to exploit the capabilities of drones in this field. Google X's Wing initiative has been working since 2012 to create an efficient, safe and precise drone delivery system. They successfully completed their first real-life drone delivery in Queensland, Australia in 2014, successfully transporting first-aid kits, candy bars, dog treats, and water to farmers [2]. Meanwhile, Amazon has also undertaken a project by the name, Prime Air, aiming to create an efficient autonomous drone delivery system as a substitute to the current delivery systems [3]. It is still in works but the initial testing phase has returned positive results which is a cause for optimism. The fact that these gargantuan corporations are so keen on exploring a drone's delivery potential is very promising, especially in uncertain and daunting times like now during this unfortunate pandemic where timely delivery of medicinal goods is vital.
- iii) Filming: In movies and shows where a bird's eye view is required for a scene or a shot is required from a certain height, drones can be used instead of the traditional approach of a cameraperson climbing up a crane. This is not just efficient in terms of time but also a lot safer than deploying cranes or helicopters.
- iv) Perilous missions: Drones can be very useful to collect

data and information in areas that are otherwise dangerous. For example, if data needs to be acquired about the toxic gases being released into the atmosphere by a factory that suffered a hazardous leak, a drone can be sent on that mission instead of a human.

With already so many applications across multifarious domains, the applications of drones are only going to increase with extensive research. With ramped up research and use of state-of-the-art technology, drones are only going to get better. A fully automated drone is the ultimate dream, and this dream seems attainable with the rapid progress being made in this domain.

Even with all their advantages and applications, drones do suffer from some problems of their own and are plagued with constraints. A few problems that a drone suffers from are:

- i) **Battery Life:** A limited battery life limits the distance a drone can travel severely. It will need to visit charging stations constantly, so it does not run out of battery and completes the task. This is not just costly in terms of energy but also time. This is a major constraint of drones that we are yet to overcome.
- ii) **Aerial:** While the air is clear of any traffic and congestion it is still rife with its own problems. Drones are affected by wind, air drift, etc. while travelling through air. This can not only slow the drone down but also misdirect it completely.
- iii) **Fragility:** Perhaps the most debilitating problem with drones is their fragility. They are incapable of carrying heavy loads and if overloaded by chance, they are prone to crashing or falling. This lack of robustness of drones prevents us from using it in areas that involve heavy lifting.

All these constraints and problems that plague drones were the inspiration to take up this project. This project has attempted to extenuate some of these problems significantly. To achieve this, an algorithm that is a modified version of Dijkstra's Shortest Path algorithm was devised. The following constraints have been satisfied with the adoption of this approach:

- i) **Area Maximization:** The algorithm has been integrated with conditions that try not just to minimize the distance between the source and the sink but also do so while attempting to maximize the area between the source and the sink (visit the maximum nodes possible in a particular graph) while minimizing the number of drones required.
- ii) **Battery Life:** One of the problems that the drone suffered from was a limited battery life. While nothing can be done about the battery limit, it is still possible to direct a drone to a charging station that is closest to its current position. That is precisely what has been implemented in our algorithm. Provision has been made for the drone to be directed to its nearest charging station in case of low battery, thus saving a long trip, an incomplete journey or worse, a complete collapse of the drone.

A few details that are pertinent to the problem statement that are necessary to highlight. The project uses offline routing paths for a drone. This means that the user has a well-defined plan for the route of the drone before its flight. The drawback of this is that unexpected drone failures endured during its journey may cause hindrance. Since the problem statement deals with multiple waypoints, it increases computational complexity however this complexity can be compensated for

using multiple drones, a provision that we have made in our algorithm. This decreases operation completion time [4].

The paper is planned as follows: Section 2 contains the background information which gives some insights on the literature survey conducted and description of the problem in detail. In Section 3 the implementation and algorithm used is elucidated. Results and analysis extracted are discussed in Section 4. Finally, the paper is concluded along with the future scope in Section 5.

2. BACKGROUND

2.1 Literature Survey

The problem of multiple UAVs path planning is quite analogous to the multiple traveling salesman problems (mTSPs), which is a conundrum. This problem is much more complex than a single TSP and thus belongs to the NP-hard category. Solving such a problem requires high computational time, power and space [5]. Intensive research has been carried out in the domain of multiple travelling agents path planning. A variety of approaches and algorithms have been tried and tested. A generalized approach has not yet been able to provide optimal solutions for all domains, thus different domains yield optimal results only to a specific approach. The approach of using improved non-dominated sorting genetic algorithm (NSGA-II) on a truck and drone transportation system for delivering parcels [6] might not be as good as using A* (shortest path) algorithm on a problem involving a dynamic environment with pop-up threats in multi-UAV systems [7]. Both the approaches are ideal in their respective field.

Other implemented algorithms include Bellman-Held-Karp dynamic programming for drone delivery systems [8], High Performance Ant Colony Optimizer (HPACO) for TSP [9], Parallel Simulated Annealing in the field of drone route planning for military image acquisition [1] and path planning using Genetic algorithms for multi-UAV system [5].

The problem dealt here involves a multiple objective optimization like covering maximum area with the least possible number of drones, operational energy cost, distance travelled and completion time. A robust method which can be easily implemented within the framework of the aforementioned constraints was the need of the hour. Dijkstra's algorithm seemed a viable option for its simple and uncomplicated approach

Dijkstra's shortest path first (SPF) algorithm is one of the most popular search algorithms used for finding least distances between nodes of a network or a graph. It is also referred to as a single source shortest path algorithm which uses the greedy approach to find shortest paths from source to all other nodes, consequently producing a minimum spanning tree (MST). It is classified as a greedy algorithm because it finds an optimal solution at every step and continues till it gets the best solution. Subsequently, this results in a large time complexity, that is $O|V|^2$ using an array where $|V|$ is the number of nodes/vertices of the graph. A weighted graph $G(V, E)$ is given as the input and the output is the shortest path from source node to all other nodes. The best solution/path is readily available now. Here E are the edges/sides of the graph [10].

Since Dijkstra's algorithm only takes a single source shortest path into account and the requirement for the problem is multiple travelling drones, it cannot be used directly in its native form. If so, then all the drones will be following only a single best/optimum path from source to destination. The

intention is to maximize the area as well, meaning to cover the maximum number of nodes in a graph while being frugal with the number of drones used. Thus, a new path is needed for each drone. The nodes may or may not be repeated but the path must be unique for every drone taking flight. This can be only achieved if the existing algorithm is modified to fulfill these requirements.

Modifications to Dijkstra's algorithm have been a research hotspot. While remaining pertinent to the constraints and desired objectives, various refinements have been made in the past.

Some examples include collaboration between Dijkstra and Floyd-Warshall algorithm in the domain of health services to determine the best route to the nearest hospital for an emergency patient or victim [10]. Another one focuses on road network route planning by improvement of data storage structure and the searching area of restricted algorithms [11]. This improves the algorithm to a great extent. In [12], authors have introduced IDA (improved Dijkstra's algorithm) for 2D eight-neighbor grid map planning. A modified version of the classical Dijkstra has been used in public transport route planning by implementation of a penalty system [13]. Lastly in [14] the authors have amalgamated improved Dijkstra algorithm and the particle swarm optimization for path planning. The proposed work yielded good performance and results.

2.2 Problem Description

2.2.1 Scenario

A routing algorithm for planning the path of multiple UAVs to cover a given area. This problem intends to find a path from a starting point to an end point, while visiting several waypoints or intermediate nodes (terminology in the case of graphs) under some predefined constraints. The plan must minimize the travel time while maximizing the area covered. Charging stations may be present for refuelling drained batteries.

The inputs provided by a user consist of:

- Map of the area to be covered.
- Start (source) and end (sink) points.
- Number of UAVs / drones.
- Depreciation rate (% per unit cost).
- Location of Charging station(s).

Result of the problem includes:

- Minimum number of UAVs required to map the area.
- All unique paths taken by each UAV.
- Remaining battery of each UAV after completing its journey.
- A Map plotted with the traversed path of each UAV, along with the location of charging station(s).

2.2.2 Assumptions

- UAVs are assumed to fly at a constant speed and have a constant depreciation rate irrespective of the terrain.
- Aerodynamic effects such as wind disturbances and turbulence are not considered.

- The area map given as an input will be analogous to a graph data structure (G) consisting of waypoints as nodes (V) and paths connecting them as edges (E). It is well assumed that a UAV can fly only through the given edges (E). The shortest distance will be calculated accordingly.
- All UAVs are completely charged (100%) while starting their journey and even after visiting the charging stations. If it does visit a station, it is recharged at its full limit.
- Time taken to charge is equal for all UAVs irrespective of their residual battery.
- A UAV will traverse a charging station only if the cost of the edge connecting the successive waypoint results in a zero or negative battery percentage. Unless the edge consists of an end point and results in a zero battery on traversal, in such exceptional scenarios the edge will be considered.
- If the UAV does not require charging, even if the shortest path is through a charging station, the path will not be considered.

2.2.3 Constraints

For efficient working of the system, some constraints need to be defined:

- Start and end points must be different.
- Depreciation rate and number of UAVs cannot be negative.
- Start and end points cannot be charging stations. Only intermediate waypoints can be.
- Charging station (if included) must be at least one of the present waypoints in the given map. Any external point will not be accepted.
- If all the waypoints are traversed at least once, the process stops. Remaining UAVs are not routed since the maximum area is covered.

3. IMPLEMENTATION

A map is provided by a user where routing of UAVs must be done. The map as initially stated, will be converted to a graph $G(V, E)$. The graph will be undirected as the flight path can be in both directions. Since, the implementation is based on a real-world scenario, all edges (E) will be positive.

A modified Euclidean formula has been used to calculate the distance between two points on the graph since a native Euclidean formula cannot directly be used because of the points being in (longitude, latitude) form instead of Cartesian coordinates (x, y). The converted distance between two points is nothing but their edge weight.

$$Distance = 6371.01 \times \cos^{-1} (\sin(slat) \times \sin(elat) + \cos(slat) \times \cos(elat) \times \cos(sl原因 - elon)) \quad (1)$$

where,

$slat$ = Starting Latitude coordinate in radians

$slon$ = Starting Longitude coordinate in radians

$elat$ = Ending Latitude coordinate in radians

$elon$ = Ending Longitude coordinate in radians [16].

Once the weights of all edges (E) are obtained, the graph G is

converted to an adjacency matrix M . The rows and columns of the matrix M consist of all the points on the map, which is equivalent to all the nodes (V) of graph G . Every intersection of a row and column is equal to the corresponding edge weight between the row and column in graph G . If they are connected by an edge, it is equal to the value found by the modified Euclidean formula in km. If two nodes are not connected by an edge, the value is taken as 0. Note that the edge connecting the vertex to itself is also valued 0 to avoid self-loops.

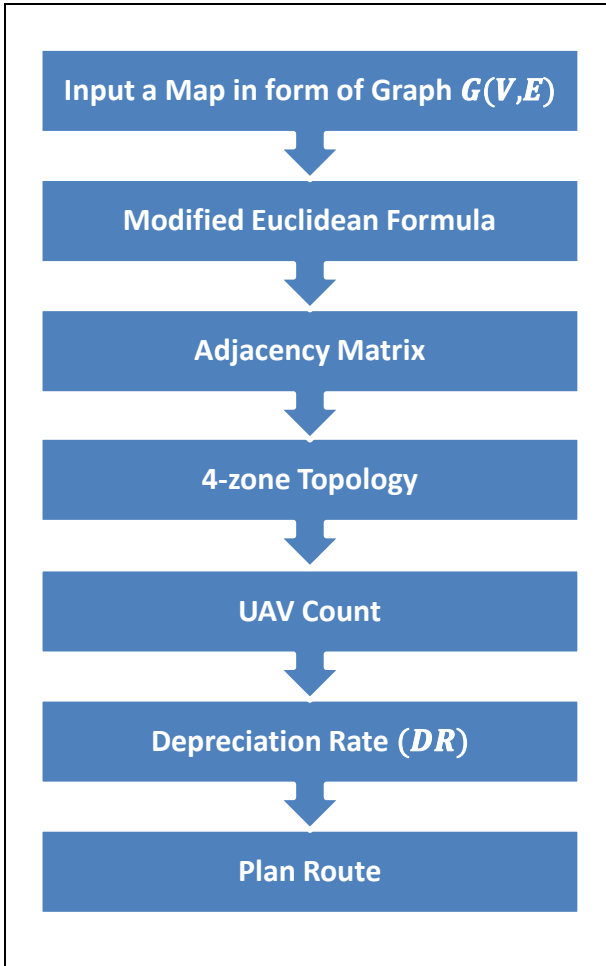


Fig 1: Block diagram of prerequisite steps for route planning

In [15] the authors have implemented a generalised 4-zone topology to the graph G which consists of a set V of nodes classified as:

$$V = \{s, e, Ch, Wp\} \quad (2)$$

Table 1: 4-zone Topology

s	=	Start node (source)
e	=	End node (sink)
Ch	=	Charging station node
Wp	=	Waypoint

Next, the UAV count is defined which must be ideally a positive integer. There is always a minimum requirement to cover all Wp at least once for any graph. If the entered count is surplus, a minimum count is generated at the end.

Last step of information needed is the Depreciation rate (DR). The value is in percentage and per unit distance. This means for every unit traveled along the edge, DR % of battery is lost. If initial battery percentage is considered B , then after traveling a distance d units (km) the equation is:

$$B = B - (DR \times d) \quad (3)$$

The path is then planned in accordance with the Modified Dijkstra's algorithm which is discussed further. Fig 1 summarizes all the prerequisite steps for implementing the route plan.

3.1 Algorithm Description

The approach used in solving such a problem is to first find the shortest path for the first UAV and track its path. This can be found using Dijkstra's single source shortest path algorithm. Now, check if there are more UAVs available. If affirmative, the second shortest path for this new UAV must be found since area maximization also has to be considered now. Remove the initial edge taken by the first UAV when it starts its journey from source node. Now source has $N-1$ outgoing edges if it previously had N . Apply the single source shortest path algorithm to find the path for the second UAV. Continue this procedure till either UAV count is 0 or all the outgoing edges of the source are visited at least once, whichever condition is satisfied first.

Next, if there are more UAVs left and still some intermediate Wp are unvisited, they need to be traversed. Therefore, now the strategy of divide and conquer is used. It is split into 2 halves initially, where $s \rightarrow X$ (unvisited Wp) is planned in the first case. Subsequently, $X \rightarrow e$ is planned in the second case. Both these paths are merged to give the final route. *Note:* While using this divide and conquer technique, the visited nodes list is cleared, and they can be re-visited. Hence, each Wp which was unvisited during the first segment is now targeted on an individual level to plan paths strictly traversing through them. This will ensure that all Wp are covered, which leads to maximum area coverage. If any UAVs are still left to be planned, they are not planned as they are superfluous. This will yield a minimum count of UAVs in the end.

vertices with the edges of their choice. The interface is equivalent to filling an adjacency matrix.

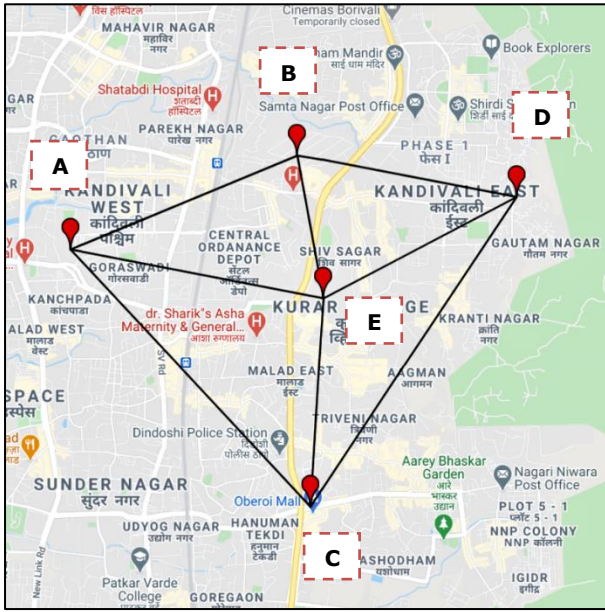


Fig 3: Input Map

The distances between each Wp / node is calculated and converted to its approximate value in kilometers (km) in Table 2 using the modified Euclidean formula.

There is another interface (Fig 4) for the user to choose the source and sink nodes from a combo box, select the charging stations needed from a checkbox and enter the number of drones and loss rate from two numeric input fields.

D-Route

Choose source node: A

Choose sink node: D

Choose Charging Stations if needed (denoted in Blue): A B C D E

Enter No. of drones:

Enter the Loss Rate (% per unit):

Fig 4: Input Window

The performance of the algorithms is analyzed using 3 different cases. The input provided is the same as in Fig 3 and Fig 4. An output window is generated (for each case) at the end which displays various output details. Additionally, a map is plotted with each drone route. Every drone route has a different color. The waypoints (Wp) are denoted in red and charging stations (Ch), if present are denoted in blue. The cases considered for analysis are as follows:

Table 2: Edges and their weights / lengths

Edges	Approximate length in km
B → E	1
A → B, A → E, B → D, E → D, C → E	2
A → C, C → D	3

Case 1: Low rate of loss

In this case, the loss rate (DR) is taken as 20%, 4 drones are chosen, A and D have been chosen as source and sink respectively and E as the sole charging station. The first drone chooses the shortest possible path from A to D using Dijkstra’s algorithm: A → B → D and the total distance covered during this journey is 4 km. $DR = 20\%$ and thus the total battery discharged is 4×20 which is 80 %. Remaining battery stands out to be 20% when is reached D. The second drone’s job now is to cover any remaining intermediate waypoints to maximize area covered while still trying to optimize the distance. The only remaining intermediate node is ‘C’, hence drone 2 goes from A → C first. However, during this journey, due to the relatively large distance between the nodes, the drone cannot go directly to ‘D’ without getting completely discharged. Distance between C → D is 3km. Therefore, total battery required to travel from C → D will be $3 \times 20 = 60\%$. At point C the battery left with drone is 40% as it travelled 3km from A → C. Choosing C → D is not practical. Hence, it goes to the charging station ‘E’, replenishes its battery to full capacity and then goes to the destination ‘D’. E → D distance is 2km. Hence remaining battery of drone 2 is displayed as 60%. Total distance travelled A → C → E → D is equal to 7km ($3 + 2 + 2$). Since all the nodes have been visited, the maximum possible area has been covered and the next two drones become redundant. Fig 5 and Fig 6 show output map and window respectively, for this case.

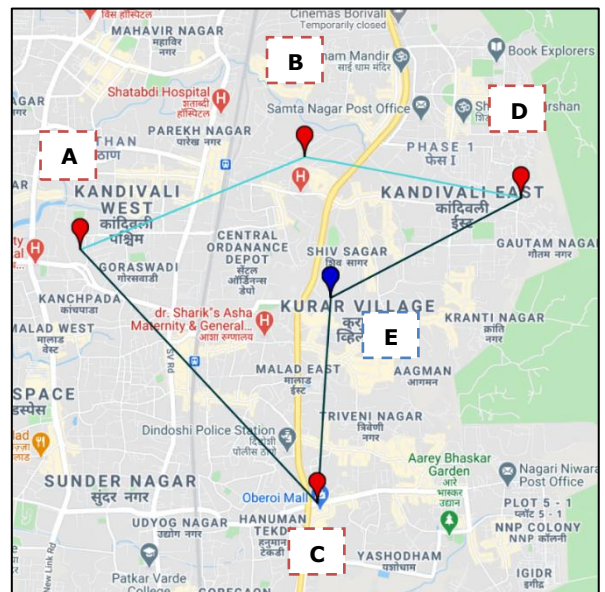


Fig 5: Output Map for Case 1

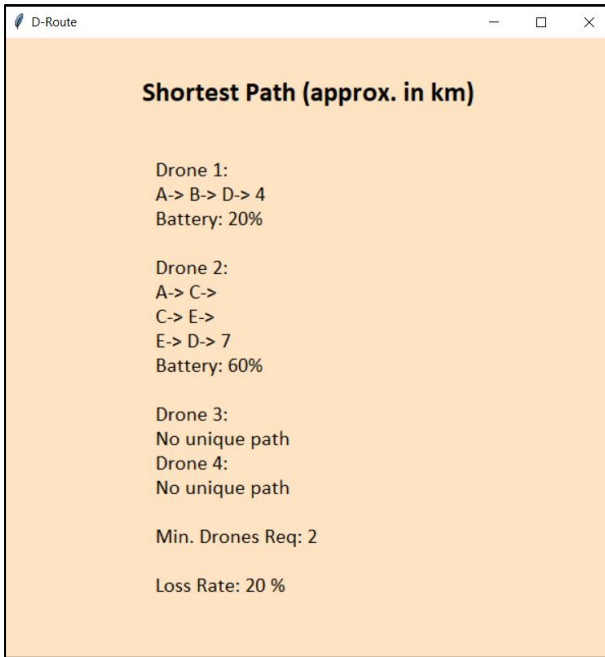


Fig 6: Output Window for Case 1

Case 2: No loss

This is an ideal case where *DR* is assumed to be 0% with all the remaining parameters the same as above. The drone count is taken as 3. The process is implemented, and we get the same result as *Case 1* except for the path chosen by drone 2. Drone 2, now, does not need to visit the charging station ‘E’ since the loss rate is 0 and hence there is no loss of battery from $A \rightarrow C$ and $C \rightarrow D$. Thus, it chooses the direct route from $A \rightarrow C \rightarrow D$ covering 6 km in the process. Both the drones retain 100% battery and drone 3 remains redundant as all *Wp* are covered. Output map and window is shown in Fig 7 and Fig 8, respectively.

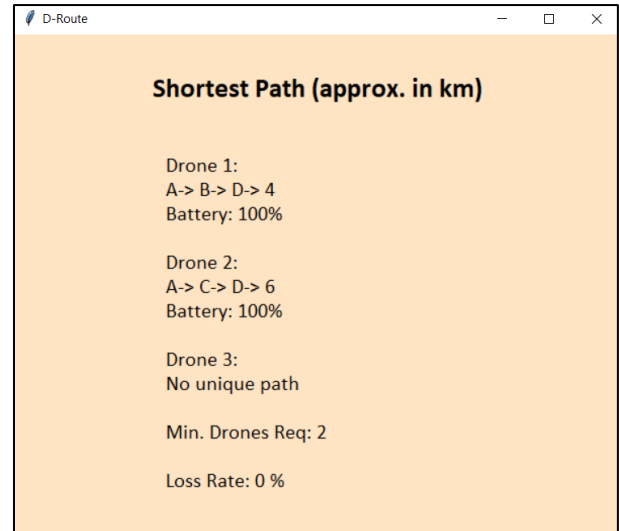


Fig 8: Output Window for Case 2

Case 3: High rate of loss

In this case, *DR* is chosen to be high, i.e., 25% with the remaining parameters remaining the same as *Case 2*. The same process as the above two cases is repeated, however, there is a problem. The first drone completes its journey exactly as it did before with 0% battery left ($4 \times 25 = 100\%$), however, the second drone goes from A to C and then stops suddenly. This is because the loss rate is so high that even though charging station E is available for use, drone 2 has lost so much of its life from $A \rightarrow C$ and the distance between C and E is so huge that it cannot afford to go from $C \rightarrow E$ without being completely drained mid-journey. At point C after traveling 3km from A, battery left is $100 - (3 \times 25) = 25\%$. Cost to reach E is $2 \times 25 = 50\%$. Therefore, this is not feasible. The output map (Fig 9) depicts the same for clearer understanding. The other drones do not embark on a journey since there are no other possible paths left to be traversed.

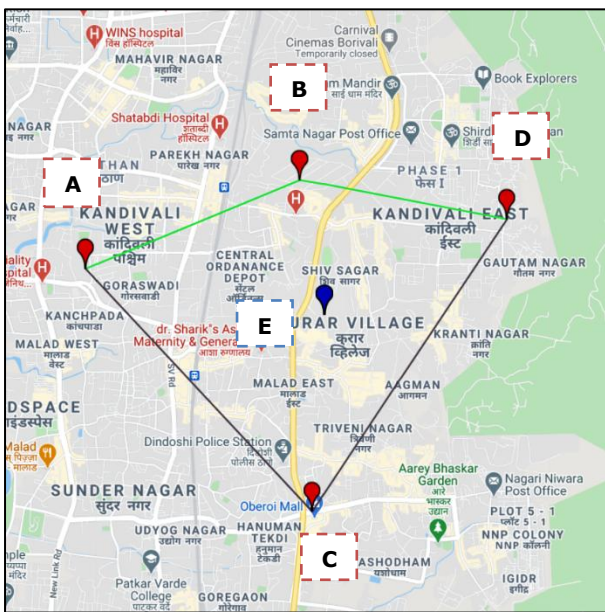


Fig 7: Output Map for Case 2

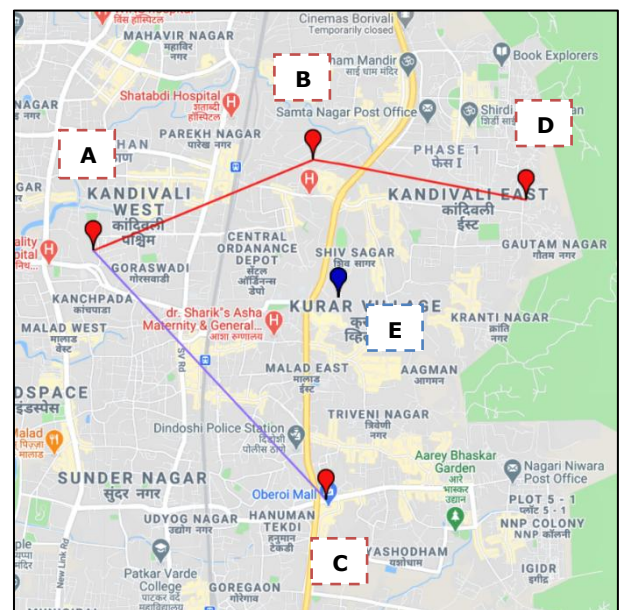


Fig 9: Output Map for Case 3

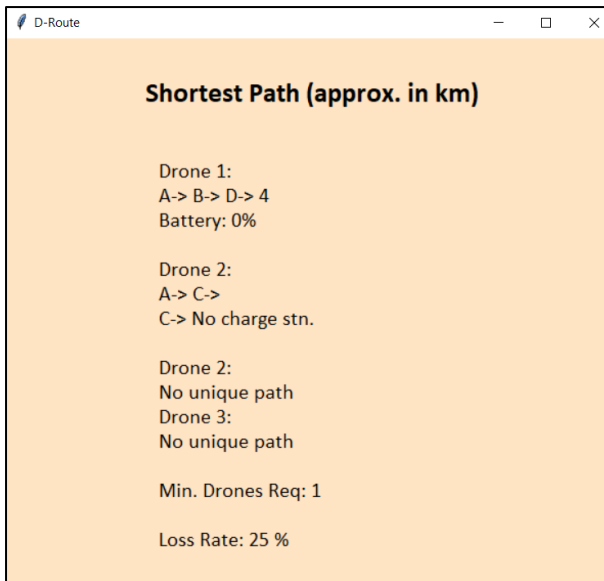


Fig 10: Output Window for Case 3

5. CONCLUSION AND FUTURE SCOPE

The results obtained above bolster the claim for the need of this unprecedented and novel routing algorithm. There have been several modifications to Dijkstra's Shortest Path algorithm for various purposes, however, no algorithms have taken the amount of constraints this algorithm has considered while devising these modifications. The constraints considered have been sufficiently elaborated on above. While the algorithm is still left wanting in terms of efficiency due to the tedious iterative and sub-iterative approach adopted, it still solves a quandary that was not explored in the depth that this project has managed to.

Several improvements can be made to this algorithm by considering more constraints that were initially ignored like variable speeds of different drones, variable depreciation rate, drone failure, different charging rates at charging stations and speed drag due to turbulence. We have devised an algorithm which serves the purpose of offline path planning. There is a scope of improvement in this sector as well. The performance of the algorithm can also be improved upon by adopting a slightly less iterative and more recursive approach.

6. REFERENCES

- [1] Eman Alsafi and Soha S. Zaghoul (2017). Drone Route Planning for Military Image Acquisition Using Parallel Simulated Annealing.
- [2] "Wing - X, the Moonshot Factory" [Online]. Available: <https://x.company/projects/wing/>
- [3] "Amazon.com: Prime Air" [Online]. Available:

<https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>

- [4] Sahingoz, O. K. (2013). Generation of Bezier Curve-Based Flyable Trajectories for Multi-UAV Systems with Parallel Genetic Algorithm. *Journal of Intelligent & Robotic Systems*.
- [5] Sahingoz, O. K. (2013). Flyable path planning for a multi-UAV system with Genetic Algorithms and Bezier curves. *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*.
- [6] Wang, K., Yuan, B., Zhao, M., & Lu, Y. (2019). Cooperative route planning for the drone and truck in delivery services: A bi-objective optimisation approach.
- [7] Moon, S., Oh, E., & Shim, D. H. (2012). An Integral Framework of Task Assignment and Path Planning for Multiple Unmanned Aerial Vehicles in Dynamic Environments.
- [8] Gjorshevski, H., Trivodaliev, K., Kosovic, I. N., Kalajdziski, S., & Stojkoska, B. R. (2018). Dynamic Programming Approach for Drone Routes Planning.
- [9] Sahana, S. K., & Jain, A. (2014). High Performance Ant Colony Optimizer (HPACO) for Travelling Salesman Problem (TSP).
- [10] Risald, Mirino, A. E., & Suyoto. (2017). Best routes selection using Dijkstra and Floyd-Warshall algorithm. *2017 11th International Conference on Information & Communication Technology and System (ICTS)*.
- [11] Fan, D., & Shi, P. (2010). Improvement of Dijkstra's algorithm and its application in route planning.
- [12] Li Wenzheng, Liu Junjun, Yao Shunli (2019). An Improved Dijkstra's Algorithm for Shortest Path Planning on 2D Grid Maps.
- [13] Bozyigit, A., Alankus, G., & Nasiboglu, E. (2017). Public transport route planning: Modified dijkstra's algorithm.
- [14] Kang, H. I., Lee, B., & Kim, K. (2008). Path Planning Algorithm Using the Particle Swarm Optimization and the Improved Dijkstra Algorithm.
- [15] Economou, J. T., Kladis, G., Tsourdos, A., & White, B. A. (2007). UAV optimum energy assignment using Dijkstra's Algorithm. *2007 European Control Conference (ECC)*.
- [16] "Python Math: Distance between two points using latitude and longitude" [Online]. Available: <https://www.w3resource.com/python-exercises/math/python-math-exercise-27.php?passed=passed>