ENERGY EFFICIENT PATH-PLANNING FOR UNMANNED AERIAL VEHICLE

SANJOY KUMAR DEBNATH

A thesis submitted in
fulfillment of the requirement for the award of the
Doctor of Philosophy in Electrical Engineering

Faculty of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia

APRIL 2022

To my beloved parents, friends, and family

ACKNOWLEDGEMENT

In this thesis, I would like to take this opportunity to express my deepest gratitude to almighty and to everyone who supported me throughout the journey of PhD study.

Foremost, I would like to express my sincere gratitude to my supervisor, and dean Associate Prof. Dr. Rosli Omar, who has supported me with his patience and, inspired me with his valuable guidance’s as well as friendly advice during my PhD study. His guidance and encouragement assisted me to overcome many issues and hurdles related to the research activities.

My sincere thanks also go to Ministry of Education, Universiti Tun Hussein Onn Malaysia, and Research Management Centre for providing partial financial support to undertake my doctorate program.

Last but not the least; I would like to express my greatest appreciation to my mother and wife for their sacrifices, and unconditional love for me. Finally, with great sadness and sorrow, I would like to dedicate my greatest acknowledge to my late father who is always an inspiration to go for higher study.

ABSTRACT

This project develops an efficient path-planning algorithm for an unmanned aerial vehicle (UAV) in obstacle-rich environments considering minimum energy consumption. UAV is increasingly being used to replace humans in performing risky missions in adverse environments. UAV normally gets its energy from solar, hydrogen cell or li-ion batteries. However, these energy sources have limitations; for example, in a cloudy day, solar power might not be fully generated. This may result in the UAV to fail in accomplishing a given mission if its path is longer than necessary. Therefore, it is vital for the UAV to have a minimal path length which leads to the least energy consumption. The proposed path planning algorithm is called Iterative Elliptical-Convex Visibility Graph (IECoVG) which is based on visibility graph (VG) and Dijkstra's algorithm. IECoVG limits the size of the search space which will in turn reduce the number of obstacles for path planning. Performance comparison through simulation in terms of computational time and path length between IECoVG and conventional VG as well as the Iterative Equilateral Space Oriented VG (IESOVG) has been executed. Identical scenarios have been applied in order to have a fair and conclusive result. The simulation shows that IECoVG improves the computation time up to 86 % due to its efficiency in selecting the search space. To further enhance IECoVG, flight cost, segment length, heading angle change and the UAV's speed have also been considered as they proportionally affect the energy consumption of the UAV. The enhanced IECoVG named IECoVG+ can improve the energy consumption of the UAV by 10.42 %.

ABSTRAK

Projek ini membangunkan algoritma perancangan laluan yang efisyen bagi pesawat udara tanpa pemandu (UAV) dalam persekitaran yang banyak halangan dengan mengambilkira kadar penggunaan tenaga minimum. UAV semakin banyak digunakan untuk menggantikan manusia dalam melaksanakan kerja-kerja berisiko dalam persekitaran berbahaya. UAV biasanya memperolehi tenaga daripada solar, sel hidrogen atau bateri li-ion. Walaubagaimanapun, sumber tenaga ini mempunyai hadnya; sebagai contoh, dalam keadaan berawan, tenaga solar mungkin tidak dapat dijana sepenuhnya. Ini mengakibatkan UAV gagal untuk menyempurnakan tugas yang diberi jika laluannya adalah lebih jauh daripada yang diperlukan. Oleh itu adalah penting bagi UAV untuk mempunyai jarak laluan yang minimum bagi membolehkan penggunaan tenaga yang paling sedikit. Algoritma perancangan laluan yang dicadangkan dipanggil *Iterative Elliptical-Convex Visibility Graph* (IECoVG) yang berasaskan kepada graf keterlihatan (VG) dan algoritma Dijkstra. IECoVG menghadkan saiz ruang pencarian supaya bilangan halangan dapat dikurangkan ketika perancangan laluan dilakukan. Perbandingan prestasi melalui simulasi dalam bentuk masa pengiraan dan panjang laluan di antara IECoVG dan VG dan juga *Iterative Equilateral Space Oriented VG* (IESOVG) telah dibuat. Senario yang serupa telah digunakan bagi memastikan keputusan yang adil dan tuntas. Hasil simulasi menunjukkan bahawa IECoVG telah mempercepatkan masa pengiraan sehingga 86% hasil daripada keberkesanan menentukan ruang pencarian. Untuk menambah baik IECoVG, panjang segmen, perubahan sudut tuju dan kelajuan UAV telah diambilkira kerana ia memberi kesan secara langsung kepada penggunaan tenaga oleh UAV. IECoVG yang ditambahbaik digelar sebagai IECoVG+ boleh memperbaiki jumlah penggunaan tenaga sebanyak 10.42 %.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	**Description**
%	Percentage
a_n	Minor Axis
a_j	Major Axis
D_θ	Cost along with heading angle
$f(n)$	Forward Cost Function
$g(n)$	Backward Cost Function
$h(n)$	Backward Cost Function
L	Length of Minor Axis
ρ	Opening Angle
Q_{free}	Free Space
Q_{init}	Initial Configuration
Q_{goal}	Goal Configuration
Q_{obs}	Obstacles' Region
S_p	Starting Point
T_p	Target Point
W_s	Way Output
W, W_p	Way Point
Θ	Heading angle for a discrete scenario
φ	Heading Angle Range
ψ	Yaw Angle

LIST OF ABBREVIATION

Abbreviation	**Description**
ACO	Ant Colony optimization
A	Total Search Area
ACL	Autonomous Control Level
ACD	Adaptive Cell Decomposition
AUV	Autonomous Underwater Vehicles
APF	Artificial Potential Field
BLOVL	Base Line Oriented Visibility Graph
BFS	Breadth-First Search
CD	Cell Decomposition
D	Distance
C-Space	Configuration Space
CM	Cost Metrix
DFS	Depth-First Search
ECoVG	Elliptical Convex Visibility Graph
ESOVG	Equilateral Spaces Oriented Visibility Graph
EDFS	Extended Depth First Search
ECD	Exact Cell Decomposition
GUI	Graphical User Interface
GNSS	Sensing Global Navigation Satellite System
GA	Genetic Algorithm
IGSA	Improved Gravitational Search Algorithm
IPSO	Improved Particle Swarm Optimization
ICAO	International Civil Aviation Organization
IESOVG	Iterative Equilateral Spaces Oriented Visibility Graph
IECoVG	Iterative Elliptical Convex Visibility Graph
IGSA	Improved Gravitational Search Algorithm
LTL	Linear Temporal Logic

ʹOVG	Oriented Visibility Graph
$\acute{O}$, O	Obstacles
PRM	Probabilistic Roadmap
PSO	Particle Swarm Optimization
PF	Potential field
PSFP	Processing Sequence of Features of a Part
PQ	Priority Queue
RM	Road Map
RAM	Radio-frequency Absorbing Material
RRT	Rapidly Random-exploring Tree
RG	Regular Grid
SA	Simulated Annealing
UAV	Unmanned Aerial Vehicle
VG	Visibility Graph
VD	Voronoi Diagram
VANET	Vehicular Ad-hoc Network
2-D	Two Dimensional
3-D	Three Dimensional

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

The Unmanned Aerial Vehicle (UAV) is an aircraft without any pilot on board. It is a complex dynamic automated aircraft controlled remotely by a pilot at a ground control station or it flies autonomously based on pre-programmed flight plans [1]. UAVs can be used in many applications [2], like coastal surveillance, hurricane watch, traffic control [3] and still this area of research is backed by academic and industrial researchers since it has the potential in some other indispensable applications. UAV, which is now an essential companion for armed forces [4], [5] replaces humans and rescuers in performing risky mission as the first source in order to observe the affected area at adverse environments such as natural calamity.

All these applications rely on a stable aerodynamic platform and reliable energy source. UAV must navigate a pre-planned path autonomously within s the permitted duration to minimise the consumption of energy from the battery. UAVs suffer from limited energy capacity because of only on-board energy storage. The ability to fly for extended period with less energy at very high altitudes has been an on-going goal for UAVs. For a successful mission, alternative energy sources [6] such as solar power [7], hydrogen cell [8], and li-ion batteries can be incorporated.
Figure 1.1 shows PUMA UAV powered by hydrogen cell which has a power duration of 9 hours. Figure 1.2 displays a UAV for agriculture which capable of vertical take-off and landing. Figure 1.3 presents "Helios" solar powered UAV developed by NASA.

Figure 1.1: PUMA UAV powerd by hydrogen cell [8]

Figure 1.2: Vertical take-off and landing (VTOL) precision agriculture drone [9]

Figure 1.3: "Helios" solar powered UAV, developed by NASA [7]

Although various types of power systems are used in UAVs [10], [11], these sources have limitations. For instance, in a cloudy day, the amount of power generated from solar panel may not fulfil the requirement. This will lead to a crash, shorter time surveillance or failure in the mission. On the other hand, UAV has a limited flying range and hence, the time spent over the surveying territory should also be minimized.

In a mission, a UAV has to traverse a pre-planned path from a starting point to target point. To complete the mission, the UAV must have either sufficient on-board energy, or an energy efficient path planning algorithm to overcome the above-mentioned constraints. Therefore, energy efficient path planning is one of the vital enablers for the development of autonomous systems such as UAV.Besides producing an optimal path that will minimize the energy consumption, a path planning method should also hold the completeness criterion. This means that it is guaranteed to find a path if one exists.

1.2 Problem Statements

There are several existing path planning methods such as combinatorial, sampling-based, and biologically inspired. Among them, visibility graph (VG) coupled with Dijkstra's algorithm, which are under combinatorial method, are capable of finding an optimal path and hold completeness criterion. In path planning, path optimality is an important criterion that makes a UAV to find the shortest path between two points. On the other hand, an optimal path can be a path that minimizes the number of heading changes, the amount of braking or whatever a specific application requires. Completeness is also the property of a path planning method that its guarantees to find a path if one exists. However, VG has a major problem, i.e., it is relatively slow in finding a collision-free path in obstacle-rich environments. This is because the entire obstacles in C-space are considered for path planning for which the computational complexity increases. This directly affects the computational time of VG. There are numerous methods that have been developed to address this issue such as Dynamic Visibility Graph (DVG) and Iterative Equilateral Space Visibility Graph (IESOVG). However, both methods cannot adequately determine the search space area for path planning which results in a high computation time. Therefore, it is crucial to design an algorithm that can minimize the computational complexity and consequently, the computation time. Another drawback of VG-based method is that the planned path consists of many sharp turns because of the resulting piece-wise linear path. This will lead to an increased energy consumption due to the acceleration and braking near waypoints. Thus, it is assumed that by limiting the number of sharp turns within the allowable range of heading angle change, the energy consumption can be reduced throughout the UAV's operation.

1.3 Aim and Objectives

The aim of this research is to develop an energy efficient path planning algorithm for an unmanned aerial vehicle (UAV) in an obstacle-rich environment so that the energy consumption could be minimised. The objectives of these projects are:

1. To develop an energy efficient path planning algorithm for a UAV based on visibility graph method.
2. To develop a formulation for flight cost calculation based on path length, heading angle change and speed of UAV.
3. To validate the performance of the proposed algorithm through simulation.

1.4 Research Scopes and Limitations

The scopes of the study are as follows:

1. A simulation-based project is considered to develop the energy efficient path planning algorithms for UAV in an obstacle rich environment. It is executed on a 64-bit computational PC with Intel(R) Core (TM) i7-4500U CPU @ 2.40GHz and 4 GB RAM.
2. The efficiency of the developed algorithm is simulated using the MATLAB R2020b software.
3. The developed algorithm will be applied in C-space which contains rectangular shaped obstacles only. However, the planner will work with other shapes as well. In such a case in real time application, the images of high-altitude buildings, trees, transmission line etc, are considered as obstacles.
4. The proposed algorithm is designed in two-dimensional (2-D) spaces.
5. No hardware will be developed in this project and no hardware-based experiment will be conducted. It is a software-based simulation.

The limitations of the study are as follows:

1. Static rectangular obstacles will be created randomly in a particular scenario.
2. Some environmental conditions, such as rain, heat and aerodynamic issues are not measured to develop the algorithm.
3. Energy loss during operation of a UAV due to altitude is not considered during the development of the algorithm.
4. Cost calculation due to heading angle calculated is limited to 90^{o} only.
5. Safety issues are not considered for developing this algorithm.

1.5 Outline of the Thesis

This thesis is structured as follows:

Chapter 1 provides a thorough description of the general concepts required for the research, such as the overall goals to be achieved, current issues related to the research, scopes and limitations of this work and the requirement of further investigation for different techniques and instruments to be implemented.

Prior to the algorithm development, the literature review in **Chapter 2** provides a brief understanding of the theoretical knowledge and the basis of different energy efficient path planning algorithms along with their pros and cons. In addition, it is justified why particular algorithm such as visibility graph and Dijkstra's algorithm are used for energy efficient path planning. We gain ideas about the typical practices and methodologies that can be used for guiding, managing, and organizing the investigations accordingly by reviewing the works conducted by earlier research.

Chapter 3 explains the reason why the particular methodology was proposed for investigation. The proposed IECoVG is in 2D environments based on the VG method and Dijkstra's algorithm. A new method is introduced to calculate the flight cost of an unmanned aerial vehicle (UAV) considering its changing heading angle. Then various measurement taken into account such as heading angle changes, the speed and segment length to advance methodological development of IECoVG+.

In **Chapter 4**, methodological development proved by step by result and analysis.

The thesis is finally ended in **Chapter 5**, by summarizing, and concluding the research works that have been discussed previously. In addition, significance of work, recommendations and suggestions for future works are also addressed.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Unmanned Aerial Vehicle (UAV) is a type of autonomous vehicle for which energy efficient path planning is a crucial issue. The uses of UAVs have been increased for weather forecasting, image processing, traffic controlling, rescuing people [7] and replacing humans in performing risky missions at adversarial environments [8] such as 2001 World Trade Centre collapse, Hurricane Katrina in 2005 [10], and the 2011 Tohoku tsunami and earthquake. Therefore, path planning is necessary to aid human operators in dangerous circumstances [11]. This Chapter analyses all the available path-planning algorithms in terms of energy efficiency for a UAV. At the same time, the consideration is also given to the computation time, path length and completeness because UAV must compute a stealthy and minimal path length to save energy. Its (energy) range is limited and hence, time spent over a surveyed territory should be minimal, which in turn makes path length always a factor in any algorithm. In addition, the path must have a realistic trajectory and should be feasible for the UAV. The mission may be in a messy and obstacle-rich environment, e.g., in an urban area and hence, it is important for a UAV to adopt a path planning algorithm to ensure that the traversed path is collision-free and optimal in terms of path length. However, only the optimal path is not enough as it may cause the UAV to consume more energy than a sub-optimal one. Most common problem of a UAV path planning is to fly from a given starting point to a target point through a set of obstacles [12]. These obstacles may not be fixed at one location and can pop up anywhere within the workspace during the fly. An energy efficient path planning must ensure that the method/algorithm can create a safe and optimal path along with the simultaneous reduction in the travel duration to save energy/fuel.

2.2 Energy Efficient Path Planning Issues

Energy efficient path planning is a complex subject. Many entities need to be measured for calculating the energy efficient path planning. Some of them stated below are usually considered in the energy cost calculation for optimal energy efficient path planning concern[12]:

i. Path Distance
ii. Path Travel Time
iii. Path Heading Change
iv. Path Safety
v. Obstacle Hostility
vi. Weather (wind, Temperature)
vii. UAV Flying Speed and Payload
viii. Computational Complexity
ix. Scalability
x. Best Path
xi. Local navigation

i. Path Distance: Completion of a mission depends on the arrival of a UAV to a target point after going through the planned waypoints. Thus, the core importance is given to path distance that is the travelled distance between the starting and the target point[12].

ii. Path Travel Time: Travel time is also another measure of optimal energy efficient path planning algorithm. For this instance, it considers that the quickest and the shortest paths are different. The quickest/fastest means the vehicle can reach its target within the least travel period. The quickest path may not be the same as the shortest path because of the traffic and haphazard incidences just as the flying or driving guidelines like limited range of speed. Additionally, the fastest path must be updated frequently throughout the vehicle trip as traffic circumstances change quickly, particularly in huge urban areas[12].

iii. Path Heading Change: If UAV travels in a straight path without any obstacle, its speed may be constant. However, if there is obstacle on the flight path, the UAV needs to take a suitable alternative route avoiding the obstacle by changing its direction starting from the nearest waypoint. Therefore, its speed must be slowed down while passing the waypoint to ensure that it is collision-free. The UAV may need to accelerate again after avoiding the obstacle to ensure that the mission is performed within the given time. Every time the UAV changes its speed, it will cause energy loss and hence, with the increasing number of obstacles, the loss will increase proportionally[12].

iv. Safety : Most optimal energy efficient path planning algorithms emphasis on the shortest path finding when other qualities of service also deserve attention such as distance, obstacles, physical limitations of the vehicle, algorithm plan, environment, optimality, completeness, space and time complexity, dynamics etc.[12]-[13]. Among them, safety is always the first priority in a UAV mission.

v. Obstacle Hostility: Obstacles' hostility is the ratio between the area blocked by obstacles in a given free space and the size of the total free space area. To find the available path that the UAV can follow, the description of the configuration space is required. Configuration space (C-space) is the common idea behind almost all path planning approaches and it mainly consists of three elements namely workspace, free space and obstacle [13]-[14]. The UAV must have prior knowledge about the obstacle area and the free space so that it can find its path from starting to target point ensuring no collision. Considering this matter, configuration space is the area where the UAV can fly by avoiding collision and find the shortest path and, as a result, it can save energy. Hence, the configuration space indicates the actual obstacle zone and free space region for the travers of UAV. If the total search area is A and the total area blocked by obstacles is Ό then the obstacles hostility = Ό/A.

vi. Weather : In open-air flight, UAVs distinctly need to deal with the stochastic of weather circumstances which can impact the energy feeding of UAVs [15], [16]. These ensure several characteristics that can potentially and powerfully influence the solution approach routing problem for a UAV. There are two key issues of weather that influence the UAV movement and they are described below.

Wind: Wind is the foremost environmental influencer that disturbs the UAV because of its direction of flow and speed. Wind may give benefit to the energy consumptions or give bigger resistance to the movement in other scenarios [17].

Temperature: The backdrops of temperature is able to disturb the UAV's battery provision as it is interrelated to drain battery and its capability [18].

vii. UAV Flying Speed and Payload: The relative and rational flying speediness of the UAV is a precarious issue to determine the fuel consumption. Direction of the Wind speed is related with the flying speed because wind direction disturbs the flying standard of the UAV, either positively or negatively. The flying position of a UAV can situate at any subsequent: a. hovering and b. level flight, cruising or horizontal moving also c. vertical moving-vertical take-off/ altitude adjustment /landing alteration. Therefore, the flying condition of the UAV must be measured along with the flying speediness in computing the energy feeding[17].

Normally, UAVs carry specific forms of payloads, for instance, camera kit or parcels. Effect of the dissimilar masses of payloads might be significant when deriving the model of energy consumption [17], [18]. In aircraft engineering, it is recognized that the energy/fuel consumption is subject to certain factors. Perhaps, maximum flight time or flight distance of UAV may be constrained by takeoff total weight, overweight, empty weight and thrust to the weight ratio [19], payload, and fuel weight [20]. Since the UAV's engineering/manufacturing, individual can get equivalent prototypes intended for flight for example, existing/obtainable fuel replicas for multi-rotor helicopters [21] which demonstrate the linear estimation of the energy ingesting is not appropriate aimed at huge deviations of the payload conveyed [18].

viii. Computational complexity: This is a metric associated with computational performance of an algorithm. It is important that the computational complexity of every algorithm needs to be considered [12].

ix. Scalability: The assessment of a path planning algorithm for an autonomous vehicle or UAV is considered as scalability. Scalability is a state when with a larger network, the performance of an algorithm declines. Hence, a well performed algorithm which is designed for trivial path network probably will not be appropriate for bigger path networks [12].

x. Quality of the best path: This metric is utilized to compare the multiple finest paths that are planned and computed by altered heuristics and supportive to similar metrics (i.e. Travel time, travel distance etc.) with the aim of deciding which algorithm is manipulative in obtaining the nearest answer/clarification to the ideal or optimal path [12].

xi. **Local navigation**: It is a process of avoiding obstacles by using only acquired data of the current surrounding environment. It is also a process of ensuring the vehicle's stability and safety and runs in real time using a reactive path planning approach[12].

2.3 Autonomy in UAV

Autonomy means the capability of a UAV to make its own decision based on the information presently available captured by sensors, and potentially covers the whole range of the vehicle's operations with minimal human intervention [22]. International Civil Aviation Organization (ICAO) classifies unmanned air vehicle in two ways, either remotely piloted aircraft or fully autonomous. Actual UAVs may offer transitional degrees of autonomy. E.g., a vehicle that is remotely piloted in most contexts may have an autonomous return-to-base operation. Basic autonomy comes from proprioceptive sensors. Advanced autonomy calls for situational consciousness, knowledge about the environment surrounding the aircraft from exteroceptive which an integrators information from multiple sensors [23]. Figure 2.1 illustrates the autonomous control basics for an autonomous vehicle.

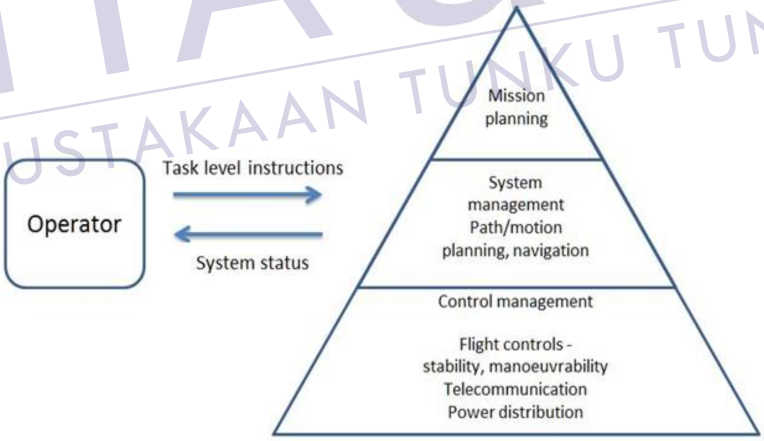


Figure 2.1: Autonomous control basics [14]

Autonomy increases system efficiency because all decisions are executed on board except for critical decisions such as launching a missile that has to be made by humans[24]. A UAV with autonomy would be able to execute a mission in environments with uncertainties. Furthermore, with autonomy, the UAV can perform a long-duration mission, which is beyond the capability of humans (operators).

UAV manufacturers often build in specific autonomous operations [25], such as:

- Self-level: The aircraft stabilizes its altitude.
- Hover: attitude stabilization on the pitch, roll, and yaw axes. The latter can be achieved by sensing global navigation satellite system (GNSS) coordinates, called alone position hold.
- Care-free: Automatic roll and yaw control while moving horizontally.
- Take-off and landing; automatically landing upon loss of control signal.
- Return-to-home and Follow-me.
- GPS waypoint navigation, communication, Path planning.
- Pre-programmed tricks such as rolls and loops.
- Sensor fusion and trajectory generation.
- Task allocation, scheduling, and cooperative tactics.

Additionally, there are ten UAV autonomy levels known as Autonomous Control Level (ACL) also introduced [25], [10]. The concepts of ACL as a metric to describe the autonomy in UAVs are widely accepted. Figure 2.2 illustrates the Autonomous Control Level and trends in UAV autonomy.

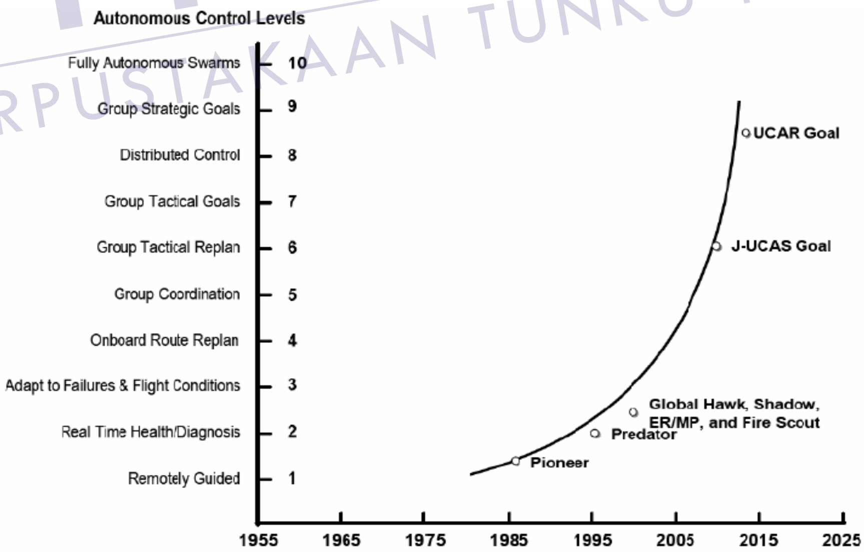


Figure 2.2: Cutting-edge autonomous levels for existing systems [7-8],[25]

However, autonomy technology is now in its moderate stage and still needs a huge development in the future [22].

2.4 Path Planning Approaches

Path planning algorithm is important to produce an optimal path that enables the shortest distance movement of a vehicle or robot from a starting point to a target point with minimum computational time. The path planning algorithm should also hold complete criterion so that it is able to find a path if one exists. Besides that, the vehicle's safety, memory usage for computing and the real-time applicable algorithm are also significant [12],[26],[29]. Path planning approaches, in general, can be classifies in three ways, such as combinatorial method, sampling-based method, and bio-inspired method. Figure 2.3 illustrates the classification of path planning approaches.

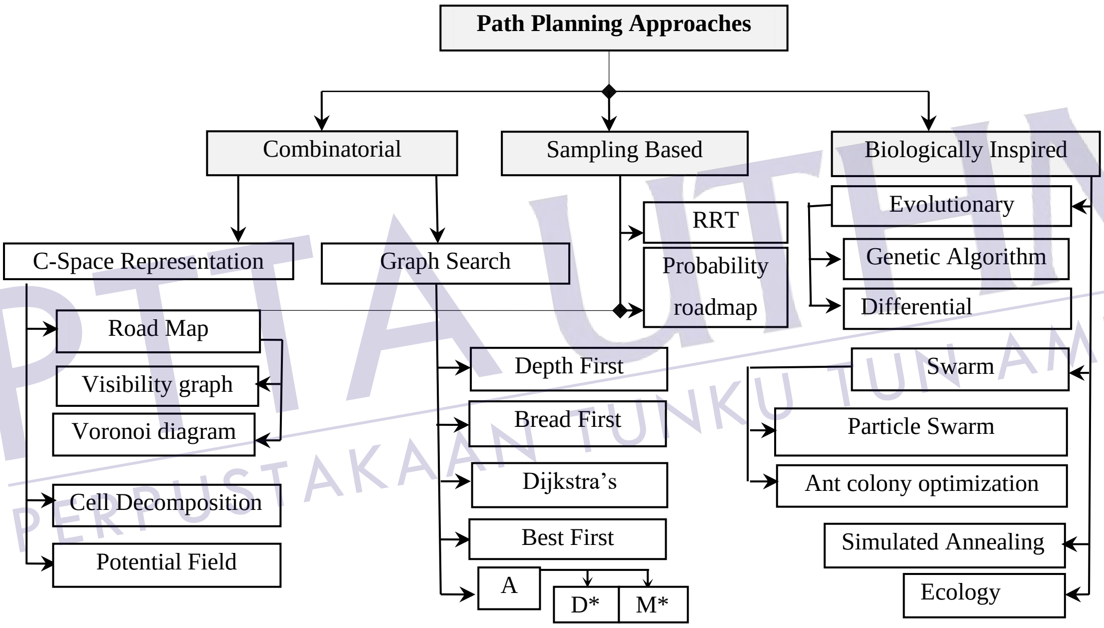


Figure 2.3: Classification of path planning approaches[12]

2.4.1 Combinatorial Path Planning

The combinatorial path planning creates a route by resolving queries along the way. Combinatorial path planning is already proposed in several techniques and classified by researcher, generally two types, they are (a) configuration space representation technique and (b) graph search algorithm. Combinatorial method applies C-space concept to the workspace representation methods such as cell decomposition (CD), potential field (PF) also visibility graph (VG), Voronoi diagram (VD) under road map technique (RM). Again, with Graph Search algorithms like Dijkstra's, A-star, Breadth First Search (BFS) and Depth First Search (DFS)[27] is associated.

2.4.1.1 Configuration Space Representation Technique

The configuration space (C-space), which is a most commonly used technique for path planning, provides detailed position information of all points in a system and this is the space for all configurations. It assumes that a UAV as a point and adds the area of the obstacles so that the path planning can be done more efficiently. C-space is obtained by adding the UAV radius while sliding it along the edge of the obstacles and the border of the search space. An illustration of a C-space for a circular UAV is shown in Figure 2.4. In Figure 2.4(a), the obstacle-free area is represented by the white background while the solid dark area represents the obstacles' region. The UAV is denoted by a black dot circled with Gray color and three pre-planned paths are represented by dotted, semi-dotted and solid lines to reach the target/goal configuration Q_{goal} from start/initial configuration Q_{init} considering that the C-space is not created. Conversely, when the workspace is considered as C-space, as shown in Figure 2.4 (b), the UAV has only one feasible path. This also reveals that the free space Q_{free} has been reduced while the obstacles' region Q_{obs} has been increased. Therefore, C-space denotes the real free space area for the movement of UAV and ensures that the vehicle or UAV must not collide with the obstacle [28].

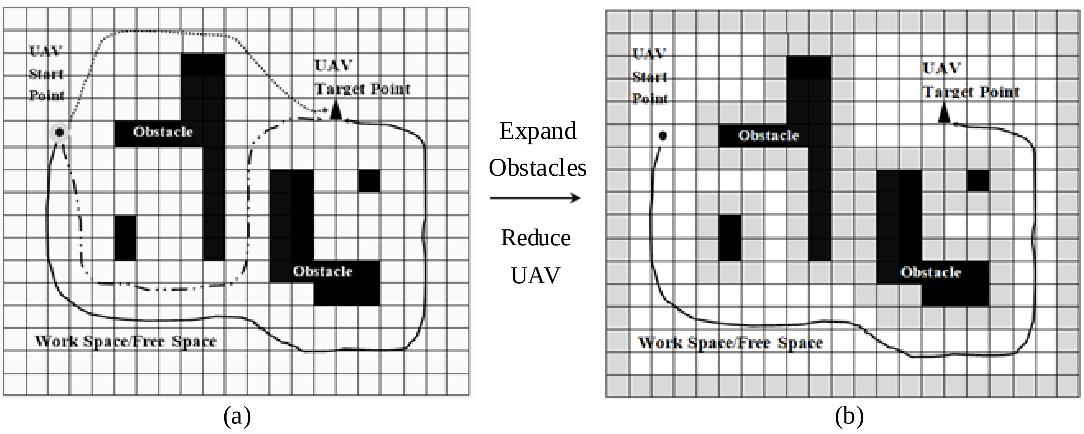


Figure 2.4: Configuration space for a UAV[26]

The popularity of C-space method in path planning is due to its use of uniform framework to compare and evaluate various algorithms.

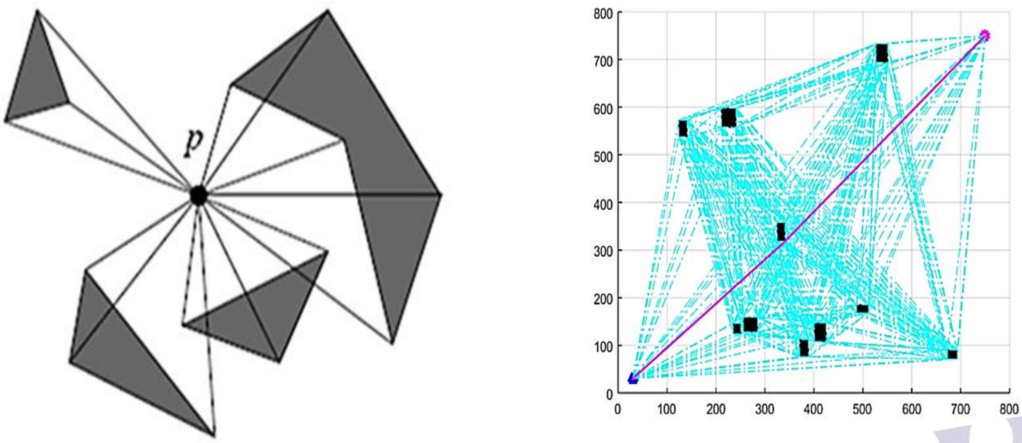
a) Road Map Technique

The C-space representation allows efficient path planning techniques based on roadmap (RM) and cell decomposition (CD) to obtain a solution. The roadmap captures the connectivity within Q_{free} using a graph or network of paths. In a roadmap, nodes are considered as points in Q_{free} and two nodes are adjoined by an edge that must be within Q_{free}. A set of collision-free paths from an initial configuration Q_{init} to a goal configuration Q_{goal} builds the roadmap that uses several steps for path planning. Firstly, it connects the nodes with edges in free C-space area to build a network or graph. After that Q_{init} and Q_{goal} are associated with the network to conclude the roadmap. A series of line segments constructs a collision-free optimal path that can be explored within $Q_{free.}$ Visibility graph and Voronoi diagram is the general classification of roadmap technique[12], [26].

i. Visibility Graph:

Visibility graph (VG) is a path planning method based on combinatorial system [29] that finds out a path by solving queries along the way. VG is used in many applications such as graphics and robotics[12], [26]. It is a set of polygonal configurations in a plane (either two or three dimensional) at an undirected graph where vertices are the obstacles' vertices, and the edges are the pairs of vertices. In VG, the vertices include starting point and target point [30]. An open line segment between two vertices does not intersect any obstacle [30], [31]. To proceed with the visibility graph in search space, the sets of vertices that are mutually visible need to be discovered. This implies that for each pair of vertices, it needs to be tested whether the connecting line segment hits any obstacle. Figure 2.5(a) shows visibility graph and connection of vertices.

VG is a path planning method that produces optimal path, i.e., planned path has the least possible length. A shortest path is vital in energy saving for a UAV to accomplish its mission successfully. However, VG is slow in obstacle-rich environments as its computation time increases rapidly with the increment in obstacle numbers. VG usually takes $O(N^3)$ computation time [29] where N is the total number of vertices. Figure 2.5 (b) shows the VG method that is used for a path planning.As all the vertices are used for path planning, a considerably long time is required to find a path in the C-space. This becomes more apparent in obstacle-rich environments where the computation time will be exponentially related to the number of vertices.



(a) (b)

Figure 2.5: (a) Vertices connected with canter *p*[13][26],
(b)VG callculate all the obstacles inside C-space

Extensive investigations have been done on VG to reduce the computational time. For instance, researchers applied VG in [32] based on the polygon aggregation. The main idea of this technique is to cluster small obstacles and merge the polygons after clustering. The shortest path was determined by integrating each partial minimum distance path. Besides that, a study in [33] enhanced the VG algorithm by sharing local information between multiple autonomous vehicles. The modified VG, named virtual rubber band visibility graph (VRBVG) method was developed to generate a VG under the assumption that C-spaces were unknown and located outside the vehicle of sonar coverage. They used torpedo-type under actuated vehicles to travel in an unknown underwater condition**.** Dynamic visibility graph (DVG) for path planning was introduced to get an efficient path planning where a rectangle shape was used to limit the c-space[34]. Again, Optimal path planning using equilateral spaces-oriented visibility graph (ESOVG) method was developed to get a computationally efficient path for autonomous vehicle [35]. Maini and Sujit designed two step algorithm to plan obstacle-free paths for a UAV where visibility graph was used for graphical environment representation [36].

ii. Voronoi Diagram

The aim of Voronoi diagram (VD) is to find a path far from the obstacles [37]. The idea behind the VD is to generate a line segment called Voronoi edge which is equidistant from all the points of the obstacle area in C-space [38], [39]. The point, where the Voronoi edge joins with each other, is called Voronoi Vertex. Figure 2.6 is an example of VD representation that is used for path planning where resulting path is shown in solid black line. As per the illustration, VD has edges to give a maximum clearance path among set of obstacles in the C-space. If a vehicle traverses the planned path, it is guaranteed that the vehicle must not intersect any obstacle. However, the VD generated paths are not optimal in terms of length. Figure 2.6 the dashed lines in Voronoi diagram are the set of points equidistant to obstacles. The path is shown in solid darker lines [13].

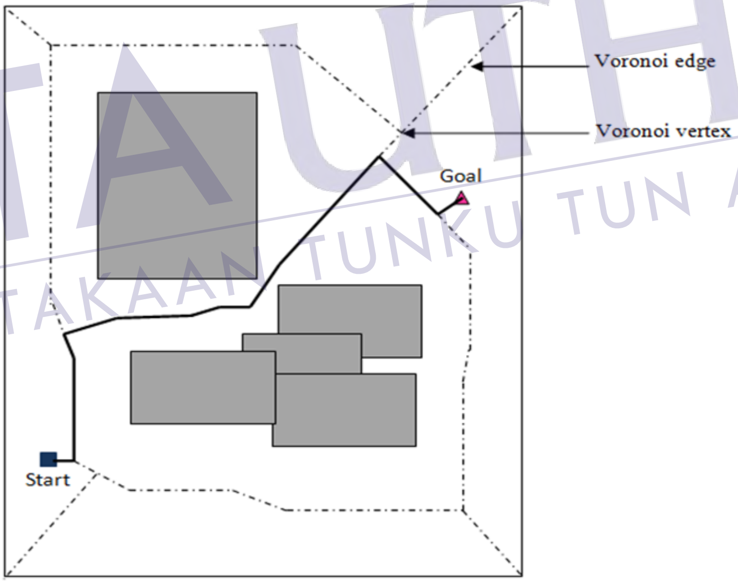


Figure 2.6: Voronoi diagram(edited) [12]

In [38], improvements were done in VD to follow the kinematic constraint of an aircraft in three steps. Firstly, an initial diagram was generated by the fundamental VD. Secondly, the initial diagram was enhanced by smoothing the impractical corner of all paths from starting point to target point. Then, the cost of the edge of improved VD was modeled and weighted. Finally, the optimal path was selected using the Dijkstra's algorithm from the smooth path. Improved VD was much lower weighted compared to the fundamental one. Another enhancement was done on the fundamental

VD in [37] with Delaunay triangulation. The algorithm was tested with 25 different settings. The outcomes revealed that the improved VD was computationally less expensive, and it responded in shorter time. But it produced a path that might not be the shortest one and this was the drawback of this algorithm. The modified VD was applied on unmanned air vehicle in a dynamic environment. Here, the path was created by using the radar threat field-based VD. To reduce the computational time, in [39] the images were captured and then clustered into a smaller group. A smooth path for a robot was also created from a path planning algorithm based on the fuzzy interference mechanism.

b) Cell Decomposition Method

Cell decomposition (CD) method mainly finds an obstacle-free cell and builds a finite graph for these cells. It breaks the environment into cells and ensures that each cell is discrete, non-overlapping and not occupied by any obstacle. A finite graph is built by relegating every cell as a hub. In cell decomposition method, the first step is to decompose the configuration space into cells. After that, the connectivity graph is built. Each node of the generated graph represents a cell, and they are between two nodes representing two corresponding adjoined cells. Then the connectivity graph from initial to end point is determined. Figure 2.7 illustrates the Cell decomposition form a starting point to goal or target point. There are several types of cell decomposition methods such as regular grid decomposition (RG), adaptive cell decomposition (ACD) and exact cell decomposition (ECD)[12] .

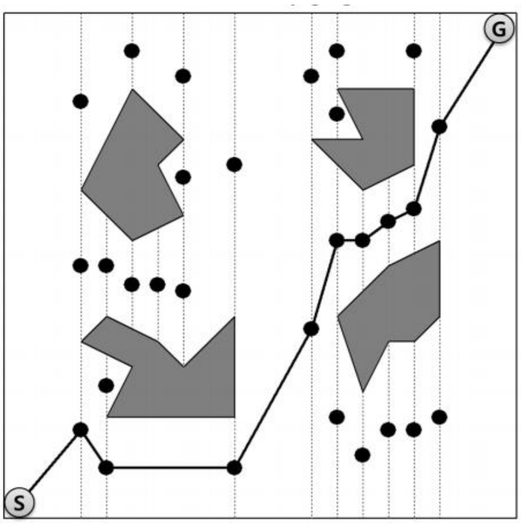


Figure 2.7: Showing path by Cell Decomposition (CD)(edited) [12]

To improve and to increase the efficiency of CD, a cell was divided quarterly. Then, the cell was checked to find out the presence of any obstacle. After that, the cell was divided again in quarter. This method was used to achieve the optimum path. Exact cell decomposition method and exhaustive path planning were used in [40]. The step for this algorithm began with the mineable area that was divided into an exact cell. Then, each cell coverage path was generated with covering direction and the generated graph paths were based on the adjacency graph. By considering all graph paths with all covering directions, a moving path was generated. Lastly, a shortest moving path for all graph paths was determined and the exhaustive path was generated. The non-optimal path is one of the drawbacks of CD. A research in [41] as an alternative to use the cell's midpoints in the fundamental CD, this technique used formulas to define the metrics. A study in [42] generated CD directly into workspaces. They applied the path planning for palletizing and common handling jobs. The algorithm produced cylindrical cell decomposition in the workspace of a six degrees of freedom robot to speed up the time without any requirement of an obstacle's transformation in a workspace into configuration spaces.

c) Potential Field Method

The potential field method (PF) was first suggested by Khatib [43]. This path planning algorithm is based on the attractive potential and repulsive potential in the configuration space consisting of a starting point, a target point, and obstacles. The vehicle is represented as a point that moves under the potential field. The target point acts as an attractive potential while the obstacles in configuration spaces simulate repulsive potential. Repulsive potential tasks in path planning are to prevent the vehicles that may collide with any existing obstacle in configuration spaces, moving under the influences of attractive forces [44], [45]. Figure 2.8 illustrates the potential field method [46].

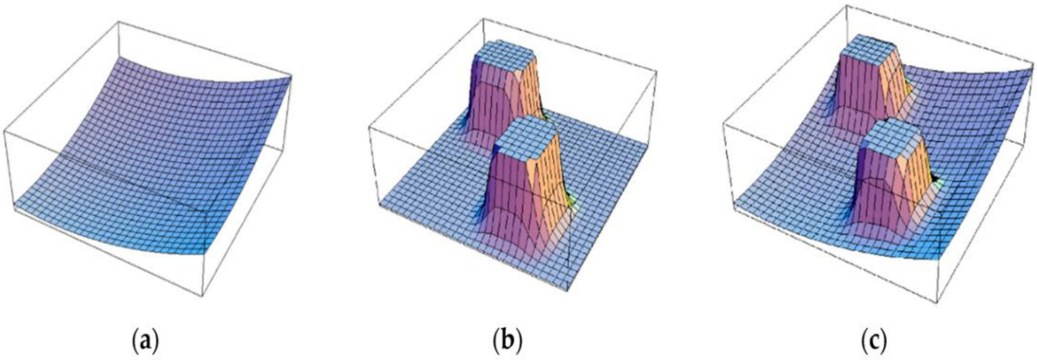


Figure 2.8: The potential field method
(a) attractive forces; (b) repulsive forces; and
(c) the sum of attractive forces and repulsive forces for potential filed [44].

A global off-line path planning methodology was applied using an energy-based approach recognized as Artificial Potential Field (APF) for Multi-Robot Systems (MRSs). Based on the potential field, a developed artificial potential field path planning technique was hosted and it was more operative in finding the shortest path [47]. Another potential field method used the kinematics of a six-wheel rover for motion on rough 3D terrain where comparative significance of the paths was obtained from four dissimilar cost functions with respect to energy, traction force, slip and deviation from a straight line. Wide experiments and simulations revealed that this technique was better in obtaining paths [48].

2.4.1.2 Graph Search Algorithms

Graph search algorithms have been used extensively in past studies for energy efficient path planning [49]. It generally determines a path from starting to target points by checking some nodes/states. After the representation of an environment by a particular method, e.g., configuration space representation method, as a second step the graph search algorithms are implemented in path planning [36]. Figure 2.9 shows the classification of graph search method.

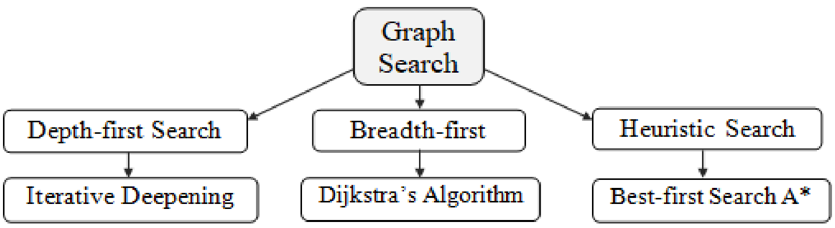
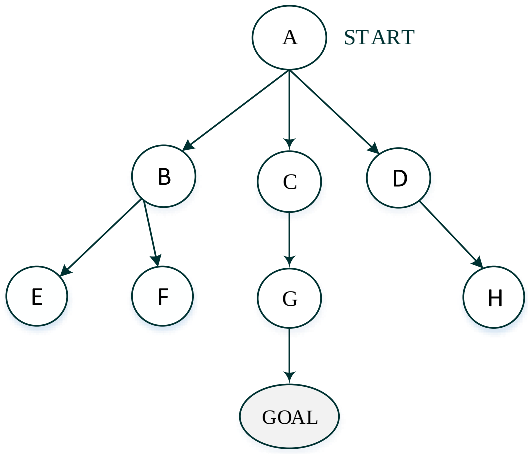


Figure 2.9: Graph search algorithms (modified) [12]

The graph search algorithms such as Dijkstra's, A*(D* and M* are modified A*), Depth-first search, Breadth First, Best first) are applied to search for an optimal path among the generated path [50], [51]. Without any existing path, it will report failure. Several graph search methods are discussed below.

a) Depth-first search

Depth-first search (DFS) moves towards the goal as quickly as possible and searches a path till getting the dead end. DFS may miss large portions of the workspace [50], [51] since it tries to search several paths at a time before completing one path. **Figure** 2.10 shows that, DFS can be applied to find a path among many possible paths. However, DFS is an uninformed search since the cost function is not used in deciding the suitable direction and in estimating that how far is the target point from the current node. DFS may be slightly faster in case it picks the leaf node path that contains the required node. When numerous solutions are in the tree where everyone is at a comparable `depth', then there is a chance to miss a larger part of the tree from exploring. Conversely, there is a chance for it to stuck in the lengthy blind alleys, whereas fewer steps solution path exists and hence, it is not the best solution. When the depth values in the search are fixed, it prevents the above issue. But this method is not much effective. DFS is good in selecting one solution among many possibilities without any prior knowledge. DFS not suitable when only one or the shortest solution exits. In DFS, the required memory is linear against the search graph making it advantageous. It keeps the record of the nodes in the 'current' path leading to less memory requirement for tree search. It is an exhaustive and systematic search method that utilizes every node in the finite search-space.



Step 1: Explore paths A→B (Goal not found)
Step 2: Explore paths A→B→E (Goal not found) A→B →F
Step 3: Explore paths A→C (Goal not found)
Step 4: Explore paths A→ C→G (Goal not found)
Step 5: Explore paths A→ C→ G → GOAL (Goal found)

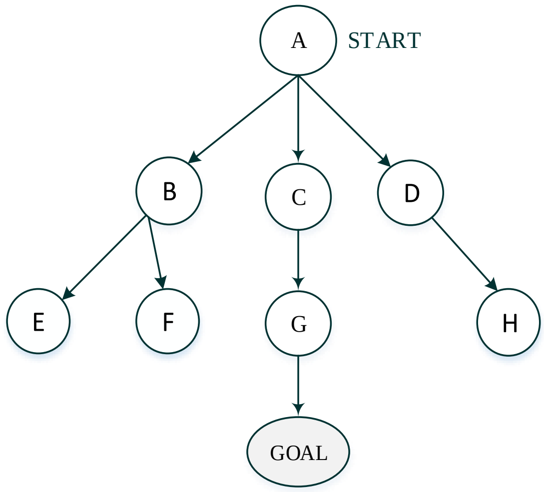
Figure 2.10: Depth-first search (modified) [14]

DFS incorporated Genetic algorithm to discover the optimal processing sequence of features of a part (PSFP) that reduces the feature transitions' energy consumption by 28.60 % [52]. A smaller search space was explored faster with reduced cost by another extended depth-first search (EDFS) algorithm [53].

b) Breadth-first Search

Moore introduced the Breadth-first Search algorithm in 1957 [54]. It is a systematic search algorithm because it first expands the shallow nodes by searching all the next level nodes of the path and then it takes the next step [55]. However, like DFS, Breadth-first Search is an uninformed search to find the shortest path in first attempt. It is applicable in limited solutions that use comparatively minimum steps.

Figure 2.11 shows that, Breadth-first Search can be applied to find a path among many possible paths.



Step 1: Explore paths A→ B, A→ C, A→ D (Goal not found)
Step 2: Explore paths A→B→E, A→B→F, A→C→G, A→D→H (Goal not found)
Step 3: Explore paths A→C→G→ GOAL, (Goal found); In the event of tie, the left node is chosen first

Figure 2.11: Breadth-First search (modified form) [14]

Breadth-first Search algorithm uses more memory and traverses all nodes. It always provides first solution in finding shortest path or determines a path with minimum steps without getting stuck in any blind alleys. The main feature of Breadth-first Search is that when all the graph's edges have no weight or same weight, the shortest path lies within the first visited node and the source node. This algorithm is complete if one exists. It is also a systematic and exhaustive search technique that eventually tries all the nodes in the search space. Breadth-first Search is faster than DFS when the required information is closer to the root of the beginning of the search. However, the total speed depends on the information storage procedure. The memory requirement of Breadth-first Search is high because it saves each level record, and this is its main limitation. It is mainly used to find the shortest path between any two nodes in a graph such as road networks, computer networks: social networks (e.g., Facebook).

c) Dijkstra's Algorithm:

Edsger Dijkstra's introduced this systematic search algorithm in 1959 [56] to find an optimal path [57] in between the initial and all other points in the graph as per the costs associated with traversal. The priority queue saves the cost of the nodes which is non-negative.

Dijkstra's algorithm measures the distance of node *n* which is denoted by $g(n)$ with respect to the starting node in the graph. The cost of the node is non-negative and stored in a priority queue. For example, a node *n* that is stored in priority queue has the cost of

$$f(n)=g(n)$$

$f(n)$ is also called the backward cost or cost-to-come. The cost is calculated gradually during the algorithm execution. As the cost is non-negative, the cost is monotonically increased. For example, if the next node of n is n', and the distance between them is $l(n, n')$, then the cost-to come is updated to

$$f(n') = f(n) + l(n, n') = g(n') \tag{2.1}$$

Since $l(n, n')$ is non-negative, $f(n')$ is therefore greater than $f(n)$.

Dijkstra's algorithm visits all the nodes within the graph starting from an initial point (S_p) and extends outward within the graph, until all nodes are visited. Dijkstra's is complete if a solution exists. It does not calculate the distance between each node and the target in optimal cases, if no prior knowledge of the graph exists. As a result, Dijkstra's algorithm is a systematic search algorithm. To establish the steps in Dijkstra's algorithm, let $d(p)$ be the distance from a source node x to a node p; and let $l(p,q)$ be the cost between adjacent/neighboring nodes p and q. The steps of Dijkstra's algorithm are then as follows:

Step 1: Set the priority queue, $PQ=\{x\}$. For each node p not in PQ, set $d(p) = l(x,p)$. For all nodes that are not adjacent to x, set their values to infinity.

Step 2: At each subsequent step, find a node q that is not in PQ where $d(q)$ is minimum. Then add q in PQ and set the parent of q to p. Subsequently update $d(p)$ for all the remaining nodes which are not in PQ by finding its minimum cost using:

$$d(p)=\min\,[d(p),\, d(q)+l\,(p, q)] \tag{2.2}$$

Step 2 is done recursively until node q is the target point. Figure 2.12(a) to (f) show the working principle of Dijkstra's algorithm.

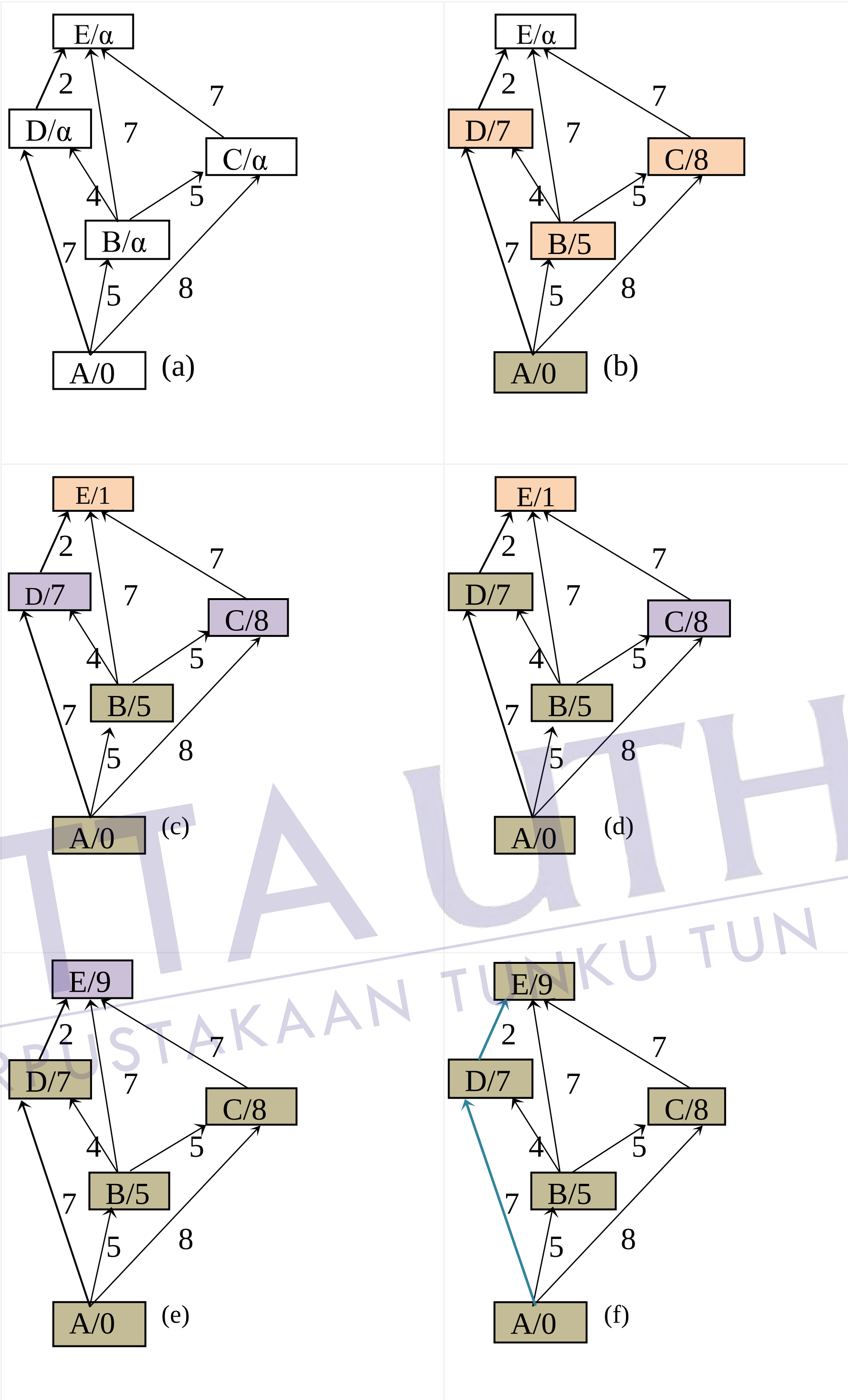


Figure 2.12: Dijkstra's algorithm illustrations (modified) [14]

Considering a scenario in Figure 2.12(a) where Dijkstra's algorithm demonstrates how it finds a path from source node *A* to goal node *E*. Dijkstra's algorithm starts at node *A* and therefore, the node is set in the priority queue (*PQ)* as shown in Figure 2.12(b).

REFERENCES

[1] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, "Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use," *Remote Sens.*, vol. 4, no. 6, pp. 1671–1692, 2012.

[2] F. Ling, J. Chen, and C. Du, "Multi-obstacle Path Planning of UAV Based on Improved Ant Colony System Algorithm." In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 1731-1735, June 2020. IEEE.

[3] D. S. Ponchak, F. L. Templin, G. Sheffield, P. Taboso, and R. Jain, "Advancing the Standards for Unmanned Air System Communications, Navigation and Surveillance," *IEEE Aerosp. Conf. Proc.*, vol. 2019-March, pp. 1–9, 2019.

[4] F. F. Milan and A. B. Tabrizi, "Small Wars & Insurgencies Armed, unmanned, and in high demand: the drivers behind combat drones proliferation in the Middle East behind combat drones proliferation in the Middle East," *Small Wars Insur.*, vol. 31, no. 4, pp. 730–750, 2020.

[5] N. Mehallegue, M. Djellab, and K. Loukhaoukha, "Efficient Use of UAVs for Public Safety in Disaster and Crisis Management," *Wirel. Pers. Commun.*, vol. 116, no. 1, pp. 369–380, 2021.

[6] C. Van Nguyen, T. Van Quyen, A. M. Le, L. H. Truong, and M. T. Nguyen, "Advanced hybrid energy harvesting systems for unmanned ariel vehicles (UAVs)," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 1, pp. 34–39, 2020.

[7] L. Sai, Z. Wei, and W Xueren, "The development status and key technologies of solar powered unmanned air vehicle." In *IOP conference series: materials science and engineering*, Vol. 187, No. 1, pp. 012011. IOP Publishing, 2017.

[8] M. G. Sürer and H. T. Arat, "State of art of hydrogen usage as a fuel on aviation," *European Mechanical Science*, vol. 2, no. 1, pp. 20–30, 2018.

[9] J. Senthilnath, M. Kandukuri, A. Dokania, and K. N. Ramesh, "Application of UAV imaging platform for vegetation analysis based on spectral-spatial

methods," *Comput. Electron. Agric.*, vol. 140, pp. 8–24, 2017.

[10] S. M. S. M. Daud, M. Y. P. M Yusof, C. C. Heo, L. S. Khoo, M. K. C.Singh, M. S. Mahmood and H. Nawawi, "Applications of drone in disaster management: A scoping review. " *Science & Justice*, *62*(1), pp.30-42, 2022.

[11] M. Dunbabin, S. Brosnan, J. Roberts, and P. Corke, "Vibration isolation for autonomous helicopter flight," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2004, no. 4, pp. 3609–3615, 2004.

[12] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, no. October 2019, pp. 270–299, 2020.

[13] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, and A. García-Cerezo, "Path Planning for Autonomous Mobile Robots: A Review." *Sensors*, 21(23), pp.7898, 2021.

[14] R. B. Omar, "Path planning for unmanned aerial vehicles using visibility line-based methods." *PhD diss.*, University of Leicester, 2012.

[15] Z. Qian, G. Wang, J. Wang, and Y. Shi, "Route Planning of UAV Based on Improved Ant Colony Algorithm," *Computer & Digital Engineering*, pp. 1421–1426, 2015.

[16] I. Sariçiçek and Y. Akkuş, "Unmanned Aerial Vehicle hub-location and routing for monitoring geographic borders," *Appl. Math. Model.*, vol. 39, no. 14, pp. 3939–3953, 2015.

[17] C. M. Tseng, C. K. Chau, K. Elbassioni, and M. Khonji, "Autonomous Recharging and Flight Mission Planning for Battery-operated Autonomous Drones," *arXiv preprint arXiv*:1703.10049, 2017.

[18] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle Routing Problems for Drone Delivery," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, 2017.

[19] V. K. Shetty, M. Sudit, and R. Nagi, "Priority-based assignment and routing of a fleet of unmanned combat aerial vehicles," *Comput. Oper. Res.*, vol. 35, no. 6, pp. 1813–1828, 2008.

[20] J. Zhang, L. Jia, S. Niu, F. Zhang, L. Tong, and X. Zhou, "A space-time network-based modeling framework for dynamic unmanned aerial vehicle routing in traffic incident monitoring applications," *Sensors (Switzerland)*, vol. 15, no. 6, pp. 13874–13898, 2015.

[21] G. J. Leishman, 2006. *Principles of helicopter aerodynamics with CD extra*. Cambridge university press.

[22] C. S. Veerappan, P. K. K. Loh, and R. J. Chennattu, "Smart Drone Controller Framework—Toward an Internet of Drones". In *AI and IoT for Smart City Applications*, Springer, Singapore. pp. 1-14, 2022.

[23] D. Floreano and R. J. Wood, “Science, technology and the future of small autonomous drones,” *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.

[24] M. Ferry, Z. Tu, L. Stephens, and G. Prickett, “Towards true UAV autonomy,” *Conf. Proc. 2007 Information, Decis. Control. IDC*, pp. 170–175, 2007.

[25] E. Sholes, “Evolution of a UAV autonomy classification taxonomy,” In *2007 IEEE Aerospace Conference*, pp. 1-16, 2007. IEEE.

[26] V. S. Ajith and K. Jolly, “Unmanned aerial systems in search and rescue applications with their path planning: a review,” *J. Phys. Conf. Ser.*, vol. 2115, no. 1, p. 012020, 2021.

[27] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[28] D. Yang, L. Zhang, R. Ran, and X. Xie, “A new evolutionary algorithm for the shortest path planning on curved surface,” *2006 7th Int. Conf. Comput. Ind. Des. Concept. Des. CAIDC*, 2006.

[29] N. Tran, D. T. Nguyen, D. L. Vu, and N. V. Truong, “Global path planning for autonomous robots using modified visibility-graph,” *2013 Int. Conf. Control. Autom. Inf. Sci. ICCAIS 2013*, pp. 317–321, 2013.

[30] H. Kaluder, M. Brezak, and I. Petrovic, “A visibility graph based method for path planning in dynamic environments,” *2011 Proc. 34th Int. Conv. MIPRO*, pp. 717–721, 2011.

[31] H. Choset, K. M. Lynch, S. Hutchinson, W. Burgard, and L. E. Kavraki, “Principles of Robot Motion: Theory, Algorithms, and Implementations [Book Review],” *IEEE Robot. Autom. Mag.*, vol. 12, no. 3, pp. 110–110, 2005.

[32] T. T. N. Nguyet, T. Van Hoai, and N. A. Thi, “Some advanced techniques in reducing time for path planning based on visibility graph,” *Proc. - 2011 3rd Int. Conf. Knowl. Syst. Eng. KSE 2011*, pp. 190–194, 2011.

[33] Y. Ma, G. Zheng, and W. Perruquetti, “Cooperative path planning for mobile robots based on visibility graph,” *Proc. 32nd Chinese Control Conf.*, pp. 4915–4920, 2013.

[34] H. P. Huang and S. Y. Chung, “Dynamic visibility graph for path planning,”

2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst., vol. 3, pp. 2813–2818, 2004.

[35] N. B. A. Latip, R. Omar, "Optimal path planning using equilateral spaces oriented visibility graph method," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3046–3051, 2017.

[36] Z. Ahmad, F. Ullah, C Tran, and Lee, S., 2017. "Efficient energy flight path planning algorithm using 3-d visibility roadmap for small unmanned aerial vehicle." *International Journal of Aerospace Engineering*, 2017.

[37] E. J. Gómez, F. M. Santa, and F. H. M. Sarmiento, "A comparative study of geometric path planning methods for a mobile robot: Potential field and Voronoi diagrams," In *2013 II international congress of engineering mechatronics and automation (CIIMA)*, pp. 1-6, 2013. IEEE.

[38] C. Peng, X. Lu, J. Dai, and L. Yin, "Research of path planning method based on the improved Voronoi diagram," *2013 25th Chinese Control Decis. Conf. CCDC 2013*, pp. 2940–2944, 2013.

[39] R. Gonzalez, C. Mahulea, and M. Kloetzer, "A Matlab-based interactive simulator for mobile robotics," *IEEE Int. Conf. Autom. Sci. Eng.*, vol. 2015-Octob, pp. 310–315, 2015.

[40] D. H. Kim *et al.*, "Path tracking control coverage of a mining robot based on exhaustive path planning with exact cell decomposition," *Int. Conf. Control. Autom. Syst.*, no. September 2016, pp. 730–735, 2014.

[41] M. Kloetzer, C. Mahulea, and R. Gonzalez, "Optimizing cell decomposition path planning for mobile robots using different metrics," *2015 19th Int. Conf. Syst. Theory, Control Comput. ICSTCC 2015 - Jt. Conf. SINTES 19, SACCS 15, SIMSIS 19*, pp. 565–570, 2015.

[42] C. Scheurer and U. E. Zimmermann, "Path planning method for palletizing tasks using workspace cell decomposition," In *2011 IEEE International Conference on Robotics and Automation*, pp. 1-4, June, 2011. IEEE.

[43] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots." In *Autonomous robot vehicles*, (pp. 396-404). Springer, New York, NY, 1986.

[44] G. Y. Jeon and J. W. Jung, "Water sink model for robot motion planning," *Sensors (Switzerland)*, vol. 19, no. 6, 2019.

[45] G. Li, A. Yamashita, H. Asama, and Y. Tamura, "An efficient improved artificial potential field based regression search method for robot path planning," *2012 IEEE Int. Conf. Mechatronics Autom. ICMA 2012*, pp. 1227–

1232, 2012.

[46] E. N. Sabudin, R. B. Omar, A. Joret, A. Ponniran, M. S. Sulong, and H. A. Kadir, “Improved Potential Field Method for Robot Path Planning with Path Pruning.” In *Proceedings of the 11th National Technical Seminar on Unmanned System Technology*, pp. 113-127, 2019. Springer, Singapore.

[47] Y. B. Chen, G. C. Luo, Y. S. Mei, J. Q. Yu, and X. L. Su, “UAV path planning using artificial potential field method updated by optimal control theory,” *Int. J. Syst. Sci.*, vol. 47, no. 6, pp. 1407–1420, 2016.

[48] R. Raja, A. Dutta, and K. S. Venkatesh, “New potential field method for rough terrain path planning using genetic algorithm for a 6-wheel rover,” *Rob. Auton. Syst.*, vol. 72, pp. 295–306, 2015.

[49] L. Blasi, E. D’Amato, M. Mattei, and I. Notaro, “Path planning and real-time collision avoidance based on the essential visibility graph,” *Appl. Sci.*, vol. 10, no. 16, 2020.

[50] A. R. Reece, “Ground vehicles,” *J. Terramechanics*, vol. 17, no. 3, p. 161, 2003.

[51] R. Omar and D. Gu, “Visibility Line Based Methods for UAV Path Planning,” In *2009 ICCAS-SICE*, pp. 3176-3181, 2009. IEEE

[52] L. Hu, Y. Liu, C. Peng, W. Tang, R. Tang, and A. Tiwari, “Minimising the energy consumption of tool change and tool path of machining by sequencing the features.” *Energy*, *147*, pp.390-402, 2018.

[53] C. Li and M. Ueno, “An extended depth-first search algorithm for optimal triangulation of Bayesian networks.” *Int. J. Approx. Reason.*, vol. 80, pp. 294–312, 2017.

[54] M. Johann and R. Reis, “Net by net routing with a new path search algorithm,” *Proc. - 13th Symp. Integr. Circuits Syst. Des.*, pp. 144–149, 2000.

[55] K. Mehlhorn, P. Sanders, P., and P. Sanders, P., “Algorithms and data structures: The basic toolbox” *Berlin: Springer*, Vol. 55, pp. 56, 2008.

[56] E. Dijkstra, B. Laurent, and M. Marion, "EW Dijkstra, 1959, A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1, p. 269271 Version bilingue et commentée." (2021).

[57] S. Broumi, A. Bakal, M. Talea, F. Smarandache, and L. Vladareanu, “Applying Dijkstra Algorithm for Solving Neutrosophic Shortest Path Problem,” In *2016 International conference on advanced mechatronic systems (ICAMechS)*, pp. 412–416, November 2016. IEEE.

[58] K. Mehlhorn, "Data Structures and Algorithms 1: Sorting and Searching," *Springer Science & Business Media*, vol. 11, p. 336, 2013

[59] A. Immirzi, L. Erra, and C. Tedesco, "Finding crystal structures from few diffraction data by a combination of a random search with genetic algorithms," *J. Appl. Crystallogr.*, vol. 41, no. 4, pp. 784–790, 2008,

[60] J. T. Economou, G. Kladis, A. Tsourdos, and B. A. White, "UAV optimum energy assignment using Dijkstra's Algorithm," In *2007 European Control Conference (ECC)*, pp. 287-292, July 2007. IEEE.

[61] M. Enayattabar, A. Ebrahimnejad, and H. Motameni, "Dijkstra algorithm for shortest path problem under interval-valued Pythagorean fuzzy environment," *Complex Intell. Syst.*, vol. 5, no. 2, pp. 93–100, 2019.

[62] R. E. Korf, "Linear-space best-first search." *Artificial Intelligence, 62*(1), pp.41-78, 1993.

[63] J. B. Rosen, "The Gradient Projection Method for Nonlinear Programming. Part II. Nonlinear Constraints," *J. Soc. Ind. Appl. Math.*, vol. 9, no. 4, pp. 514–532, 2005.

[64] E. Mueggler, M. Faessler, F. Fontana, and D. Scaramuzza, "Aerial-guided navigation of a ground robot among movable obstacles," *12th IEEE Int. Symp. Safety, Secur. Rescue Robot. SSRR 2014 - Symp. Proc.*, no. May 2015, 2014.

[65] H. Berliner, "The B* tree search algorithm: A best-first proof procedure." In *Readings in Artificial Intelligence*, Morgan Kaufmann, pp. 79-87, 1981.

[66] A. K. Guruji, H. Agarwal, and D. K. Parsediya, "Time-efficient A* Algorithm for Robot Path Planning," *Procedia Technol.*, vol. 23, pp. 144–149, 2016.

[67] P. Sudhakara and V. Ganapathy, "Trajectory planning of a mobile robot using enhanced a-star algorithm," *Indian J. Sci. Technol.*, vol. 9, no. 41, 2016.

[68] I. C. Chang, H. T. Tai, F. H. Yeh, D. L. Hsieh, and S. H. Chang, "A VANET-based A * route planning algorithm for travelling time- and energy-efficient GPS navigation app," *International Journal of Distributed Sensor Networks*, 9(7), p.794521, 2013.

[69] M. Korkmaz and A. Durdu, "Comparison of optimal path planning algorithms," *14th Int. Conf. Adv. Trends Radioelectron. Telecommun. Comput. Eng. TCSET 2018 - Proc.*, vol. 2018-April, no. January 2019, pp. 255–258, 2018.

[70] M. Polyakova, G. Rubin, and Y. Danilova, "Method for matching customer and manufacturer positions for metal product parameters standardization," *AIP*

Conf. Proc., vol. 1946, 2018.

[71] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” *Int. J. Rob. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[72] K. Cho, J. Suh, C. J. Tomlin, and S. Oh, “Cost-Aware Path Planning under Co-Safe Temporal Logic Specifications,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2308–2315, 2017.

[73] S. Kamarry, L. Molina, E. A. N. Carvalho, and E. O. Freire,"Compact rrt: A new approach for guided sampling applied to environment representation and path planning in mobile robotics." *In 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, pp. 259-264. IEEE, 2015.

[74] J. Latombe, “Motion Planning : A Journey of Robots , Molecules , Digital,” *Int. J. Rob. Res.*, vol. 18, pp. 1–14, 1999.

[75] J. D. Marble and K. E. Bekris, “Asymptotically near-optimal planning with probabilistic roadmap spanners,” *IEEE Trans. Robot.*, vol. 29, no. 2, pp. 432–444, 2013.

[76] J.J.Chung, N. Lawrance, S.K Gan, Z.Xu, R. Fitch and S.Sukkarieh, "Variable density PRM waypoint generation and connection radii for energy-efficient flight through wind fields". In *of IEEE International Conference on Robotics and Automation*, 2015.

[77] G. Nagib, and W. Gharieb, “Path planning for a mobile robot using genetic algorithms.” *IEEE Proceedings of Robotics*, 185189, 2004.

[78] N. Ozalp and O. K. Sahingoz, “Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms,” *2013 Int. Conf. Unmanned Aircr. Syst. ICUAS 2013 - Conf. Proc.*, pp. 308–317, 2013.

[79] X. Wang, T. Sun, and D. Li, “Energy-optimal coverage path planning on topographic map for environment survey with unmanned aerial vehicles,” *Electron. Lett.*, vol. 52, no. 9, pp. 699–701, 2016.

[80] G. D. M. Souza and C. F. M. Toledo, “Genetic Algorithm Applied in UAV ’ s Path Planning,” In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1-8, July 2020. IEEE.

[81] J. Kennedy and R Eberhart, “Particle swarm optimization.” In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948), 1995. IEEE.

[82] P. Moravec and P. Rudolf, "Application of a particle swarm optimization for shape optimization in hydraulic machinery," In *EPJ Web of Conferences*, Vol. 143, p. 02076. EDP Sciences,2017.

[83] M. Davoodi, F. Panahi, A. Mohades, and S. N. Hashemi, "Clear and smooth path planning," *Appl. Soft Comput. J.*, vol. 32, pp. 568–579, 2015.

[84] P. K. Das, H. S. Behera, and B. K. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning," *Swarm Evol. Comput.*, vol. 28, pp. 14–28, 2016.

[85] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by ant colonies," In *Proceedings of the first European conference on artificial life*, Vol. 142, pp. 134–142, 1991.

[86] D. A. Calvin, "The wiki and the blog: Toward a complex adaptive intelligence community." *Studies in Intelligence*, Vol. 49, no. 3, (2005).

[87] S. Mirjalili, J. S. Dong, and A. eds. Lewis, *Nature-inspired optimizers: theories, literature reviews and applications* (Vol. 811). pp.1-6, Springer, 2019.

[88] M. D. Toksari, "A hybrid algorithm of Ant Colony Optimization (ACO) and Iterated Local Search (ILS) for estimating electricity domestic consumption: Case of Turkey," *Int. J. Electr. Power Energy Syst.*, vol. 78, pp. 776–782, 2016.

[89] T. Zaza and A. Richards, "Ant Colony Optimization for routing and tasking problems for teams of UAVs," *2014 UKACC Int. Conf. Control. Control 2014 - Proc.*, no. July, pp. 652–655, 2014.

[90] L. Huo, J. Zhu, G. Wu, and Z. Li, "A novel simulated annealing based strategy for balanced uav task assignment and path planning," *Sensors (Switzerland)*, vol. 20, no. 17, pp. 1–21, 2020.

[91] T. Turker, O. K. Sahingoz, and G. Yilmaz, "2D path planning for UAVs in radar threatening environment using simulated annealing algorithm," *2015 Int. Conf. Unmanned Aircr. Syst. ICUAS 2015*, pp. 56–61, 2015.

[92] A. Louchene, N. E. Bouguechal, A. Dahmani, S. Yahiaoui, and S. Merrouchi, "Automated guided vehicle path planning without obstacles expansion," In *Proceedings of the 1998 IEEE International Conference on Control Applications (Cat. No. 98CH36104)*, Vol. 2, pp. 1333-1337, 1998. IEEE.

[93] D. Glavaški, M. Volf and M. Bonkovic, "Robot motion planning using exact cell decomposition and potential field methods." In *Proceedings of the 9th WSEAS international conference on Simulation, modelling and*

optimization (pp. 126-131), 2009.

[94] V. D. Hoang, D. C. Hernández, J. Hariyono, and J. Kang-Hyun, "Global path planning for unmanned ground vehicle based on road map images," *Proc. - 2014 7th Int. Conf. Hum. Syst. Interact. HSI 2014*, pp. 82–87, 2014.

[95] D. Chen, L. Zhan, and X. Chen, "Mobile robot Path planning based on Behavior Information Potential Field in unknown Environments," In *2004 IEEE International Conference on Robotics and Biomimetics*, pp. 683–687, 2004. IEEE.

[96] K. Choi, and J. H. Kim, "UAV path planning method for avoiding restricted areas." *Intelligent Service Robotics*, *14*(5), pp.679-690, 2021.

[97] N. B. A. Latip and R. Omar, "Car-like robot path tracker with kinematic constraints," *Proc. - 6th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2016*, no. November, pp. 223–227, 2017.

[98] N. B. A. Latip and R. Omar, "Feasible Path Generation Using Bezier Curves for Car-Like Vehicle," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 226, no. 1, 2017.

[99] R. Datouo, F. B. Motto, B. E. Zobo, A. Melingui, I. Bensekrane, and R. Merzouki, "Optimal motion planning for minimizing energy consumption of wheeled mobile robots," *2017 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2017*, vol. 2018-Janua, pp. 2179–2184, 2018.

[100] V. Sangeetha, R. Krishankumar, K. S. Ravichandran, and S. Kar,"Energy-efficient green ant colony optimization for path planning in dynamic 3D environments." *Soft Computing*, *25*(6), pp.4749-4769.2021.

[101] Z. Yaoming, S. U. Yu, X. I. E. Anhuan, and K. O. N. G. Lingyu, "A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of UAV." *Chinese Journal of Aeronautics*, 34(9), pp.199-209, 2021.

[102] P. Yan, "Research on the Selection of Path Planning Algorithm: A Case Study in Leeds." In *Journal of Physics: Conference Series*, IOP Publishing, vol. 2179, no. 1, pp. 012039, 2022.