*International Journal of Computational and Experimental Science and ENgineering*
*(IJCESEN)*

*Vol. 11-No.1 (2025) pp. 10-39*
*http://www.ijcesen.com*



***ISSN: 2149-9144***

**Review Article**

# A Comprehensive Review of Path Planning Techniques for Mobile Robot Navigation in Known and Unknown Environments

**Shoaib Mohd Nasti[1]*, Zahoor Ahmad Najar[2], Mohammad Ahsan Chishti[3]**

[1]Department of Information Technology, Central University of Kashmir, Ganderbal, Jammu and Kashmir, 191131, India
* **Corresponding Author Email:** meshoaibnasti@gmail.com - **ORCID:** 0000-0003-4942-0033

[2]Department of Information Technology, Central University of Kashmir, Ganderbal, Jammu and Kashmir, 191131, India
**Email:** zahoornajar@cukashmir.ac.in - **ORCID:** 0000-0002-7999-5084

[3]Department of Computer Science and Engineering, National Institute of Technology Srinagar, Jammu and Kashmir, 190006, India
**Email:** ahsan@nitsri.ac.in - **ORCID:** 0000-0003-3500-9825

**Article Info:**

**DOI:** 10.22399/ijcesen.797
**Received :** 20 October 2024
**Accepted :** 27 December 2024

**Keywords :**

Autonomous Robot,
Dynamic Environment,
Heuristic Algorithms,
Obstacle Avoidance,
Path Planning,
Static Environment.

**Abstract:**

The exponential increase in the utilisation of mobile robots in day-to-day life emphasizes the need for effective path-planning algorithms that allow them to navigate safely and reliably through unknown or known environments. Path planning is the procedure in which a prime and secure path needs to be determined for the robot to relocate from source to destination. Discovering a collision-free path may be the most difficult aspect for mobile robots to navigate. Several optimal path-planning techniques have been proposed until now for finding optimal paths from source to sink in the presence of obstacles, which are essential for cost-effectiveness in terms of time of traversal and resource utilization. This paper gives a critical review of classical, heuristic and hybrid path-planning techniques. Classical technologies such as Cell Decomposition, Potential Field Methods and Roadmap Methods are characterized by computation efficiencies which range from time complexity of O(nlogn) to $O(n^2)$, and these techniques have the limitation of being not suitable for dynamic environments. Heuristic techniques that provide more flexibility in dynamic environments include Bacterial Foraging Techniques, Particle Swarm Optimization, Genetic Algorithms ,Artificial Neural Networks, Fuzzy Logic, Ant Colony Optimization, and Particle Swarm Optimization. Ant Colony Optimization and Particle Swarm Optimization provide robust real-time adaptability with very high consumption in computational resources--typically under O(WL) and O(NL) time complexity, respectively. Hybrid techniques indicate that benefits from the classical and heuristic methods reduce the path length and enhance the energy efficiency comparatively to classical methods. Hybrid techniques generally have the order of time complexity, about $O(n^2)$, to find a balance between real-time adaptability and computational efficiency. Path length, smoothness, safety degree, etc., are important optimization criteria. It assesses Key optimization criteria, such as path length, smoothness, safety level, and energy efficiency. This paper also discusses the integration of robot modelling with path planning methodologies, emphasising the importance of considering robot dynamics and kinematics. Finally, the review discusses potential directions of research in this area with a roadmap for futuristic mobile robot path planning techniques.

## 1. Introduction

Mobile robots have become much more common in many applications, such as healthcare, agriculture, logistics, and manufacturing. Using path planning that is effective and reliable for the mobile robot is the basis for improving mobility. The process of path planning is important in controlling the mobile robot for optimal and safe trajectory identification from an origin to destination point. Finding a completely collision-free path is one of the difficult tasks in mobile robot navigation [1]. The challenge in this problem is to locate an optimal path while dealing with the environmental uncertainty. An environment might be known or unknown; it is called a known environment only when the obstacles on the path are static. In contrast, if these particular obstacles on the course path are dynamic or changing from time to time, the navigation environment is treated as an unknown environment. The global strategy for navigation will be used to find a better path for known environments, depending upon already existing information, such as a detailed map. However, in unknown environments where no map is available, a local navigation approach is needed. This approach relies on real-time inputs from the robot's sensors to navigate safely through its environment. In the known as well as in the unknown environment, the robot has to determine a collision-free and shortest path from a starting point to a goal location [2]. Depending on the situation, there can be numerous different possible paths, and the objective of any path planning algorithm is to find the most optimal or at least a close-to-optimal path for a given scenario. This would be of great importance in situations such as rescue missions, where there is a high chance that a victim would need help under life-threatening circumstances. For known environments, the robot requires detailed information about its surroundings before planning a path [3]. The algorithm will consider this data, including the robot's position and nearby obstacles, to create the path connecting the start to the destination. In an unknown environment, prior knowledge is not a necessity since there are sensors that can identify the obstacles in real time [4]. However, finding the best possible route in such situations remains difficult. The path-planning strategies may be roughly classified as classical and heuristic; the further break-up is seen in figure 1. Classical techniques: Based on mathematical models and algorithms, optimal pathfinding is accomplished by exhausting all possible paths. Though classical techniques are efficient-and-optimal, time-consuming generation of paths, especially against complex environments, is often incurred. Heuristic techniques involve trial-and-error methods utilizing rules and guidelines for closing in on a goal. As such, less efficient than classical techniques, but much more resilient, heuristic techniques provide effective performance throughout all environments because they can tackle all types of uncertainties. So, these strategies offer their own set of advantages and disadvantages. Combining multiple strategies can mitigate the disadvantages of two or more strategies. Researchers have focused more on this sector in the last three to four decades [5, 6]. Mobile robots have diverse applications, including household, manufacturing, healthcare, defence, space exploration, and so on [7]. In all these sectors, collision-free path planning is essential for mobile robots to accomplish their tasks. They rely on basic building blocks for navigation in all of these applications [5], as illustrated in figure 2. This paper focuses on wheeled mobile robots due to their widespread use in research and practical applications, as well as their ability to navigate different environments efficiently.

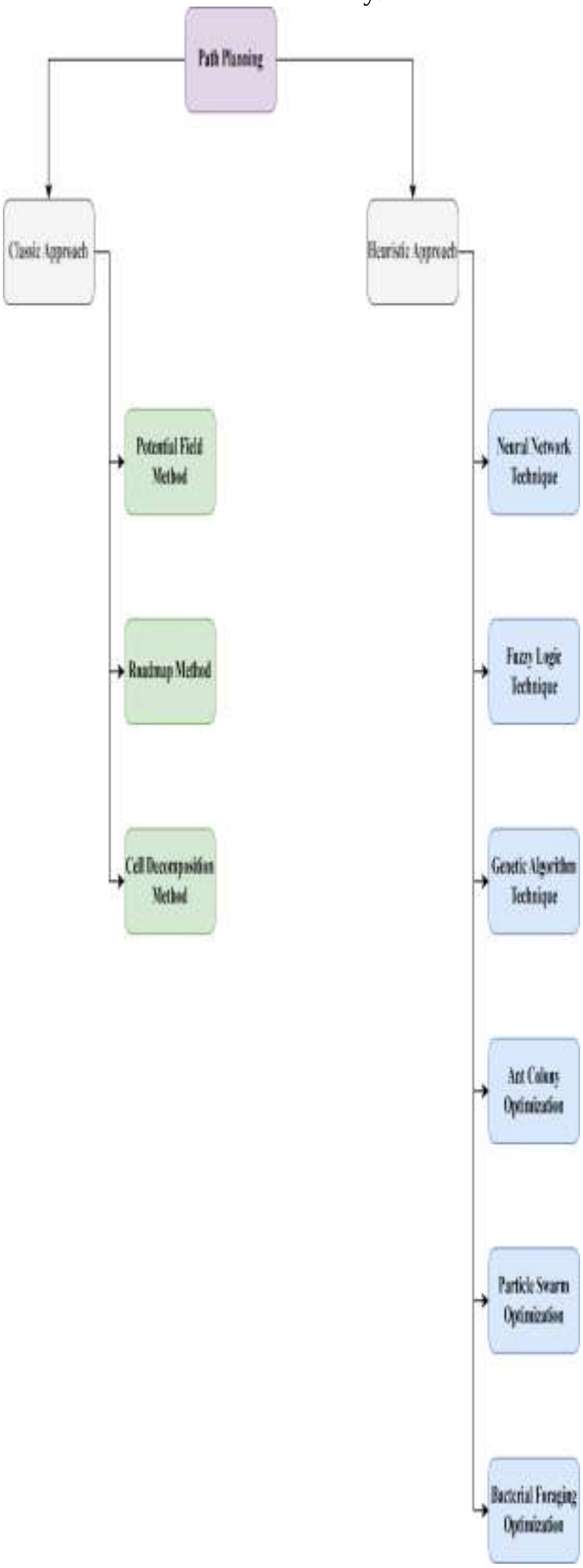


***Figure 1.*** *Mobile Robot Navigation Strategies*

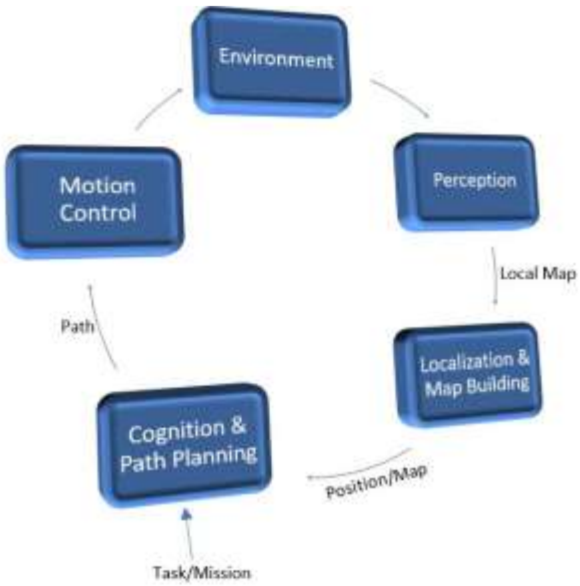


***Figure 2.*** *Mobile Robot Navigation Building Blocks*

These robots can be equipped with sensors for obstacle detection and avoidance, making them excellent for testing path planning in both unknown and known environments. The choice is further motivated by their significant presence in literature and their relatively simple motion models, which allow researchers to concentrate on the algorithms themselves. While other types of robots, like legged and aerial models, exist, wheeled robots give a solid foundation for learning path-planning techniques. Their simplicity and versatility make them a useful platform for developing and testing algorithms.

### 1.1 Main Contributions

The key contributions of this paper regarding mobile robot path-planning techniques include: Comprehensive Literature Survey: This paper surveys classical and heuristic planning techniques describing in detail more-than-100 studies. It also presents their advantages and disadvantages in terms of applications to mobile robot operation in known and unknown environments and later classifies them, evaluating these techniques by definition in relation to their practical application, especially concerning dynamics and unpredictability. Inclusive Algorithms with Time Complexities It significantly contributes by providing all-encompassing pseudocode and time complexity analysis for the various path planning methods, including the Potential Field Method, Roadmap Method, Cell Decomposition Method, Artificial Neural Networks, Fuzzy Logic, Genetic Algorithms, Ant Colony Optimization, Particle Swarm Optimization, and Bacterial Foraging Techniques. These algorithm-specific details will offer the researchers and practitioners insight on what is involved in bringing about computational costs during real-world applications.

Robot Modelling and Path Planning: Another important contribution is the integration of robot modelling, particularly kinematic and dynamic models, with path-planning approaches. The research demonstrates how robot dynamics affect the feasibility and efficiency of the planned paths. This integration is essential for ensuring the development of path-planning algorithms that can be utilised in environments involving mobile robots with different mechanical and operational constraints.

Critical Evaluation of Hybrid Approaches: The paper analyses hybrid path-planning techniques, which combine classical and heuristic methods to enhance performance in dynamic and uncertain environments. They show how to optimise these parameters, such as computation time and energy efficiency, with these hybrid approaches. How these strategies can achieve a balanced solution with real-time adaptability without compromising any more accuracy or efficiency puts these strategies in line for use in complex real-world applications, which are presented in this paper.

Optimisation Criteria: Key optimisation metrics such as smoothness, safety degree, path planning and energy efficiency are given as optimization criteria for evaluating path length algorithms. Here, the paper presents these critical optimisation metrics by which researchers and developers can accurately evaluate and compare the performance of different algorithms.

Comparative Analysis: A comparative analysis of classical techniques is provided in this research paper, applying heuristic approaches, concentrating on the performance when working with different types of navigation environments.

Future Research Directions: The paper concludes by outlining future research directions and identifying research gaps, including the development of adaptive, scalable, and computationally efficient hybrid algorithms for managing dynamic environments. These research directions provide a clear roadmap for advancing the field and addressing some of the current limitations of these techniques.

### 1.2 Research Question

The study is structured around several key questions that guide our investigation into mobile robot path planning:

RQ1: How do current path planning techniques perform in known versus unknown environments?

RQ2: How do optimization criteria such as smoothness, path length, safety degree, and energy efficiency influence the effectiveness of path planning techniques?

RQ3: How does the integration of kinematic and dynamic robot modelling influence the performance and accuracy of path-planning techniques?
RQ4: What are the limitations of heuristic path planning and classical approaches in terms of adaptability to environmental uncertainties?
RQ5: Can hybrid path planning approaches that integrate multiple algorithmic strategies offer superior performance in terms of efficiency, accuracy, and robustness?
RQ6: What future directions can be explored to further the development of resilient path-planning mechanisms for mobile robots?

### 1.3 Structure of Paper

The second section of the paper discusses existing techniques of path planning for mobile robot navigation through a review of classical and heuristic methods. Robot modeling is the subject of Section 3; it highlights the value of kinematics and dynamics models in predicting and analyzing robot behavior for the purpose of path planning. Section 4 elaborates on the optimization criteria in path planning such as path length, smoothness, safety degree, and energy efficiency. Section 5 deals with the classical techniques of path planning, such as Potential Field Method, Roadmap Method, Cell Decomposition Method, and where they work, advantages, and disadvantages. Section 6 contains heuristic methods like Artificial Neural Networks, Fuzzy Logic, Genetic Algorithms, Ant Colony Optimization, Particle Swarm Optimization, and the Bacterial Foraging Technique; principles behind them, applications, advantages, and disadvantages are mentioned. Comparative discussions between two techniques-classical and heuristic-are also presented in Section 7. The performance, strengths, and limitations of each technique in real scenarios will be discussed. Conclusively, Section 8 ends this paper by summarizing the results that would be concluded from the study and stating the significance of choosing suitable path-planning techniques and future research directions for enhancing the reliability and efficiency of such methods.

## 2. Literature Review

Mobile robot path planning is one core element of an autonomous navigation system that directs the robot to explore very complicated scenarios efficiently and safely. The different types of methodologies and algorithms developed to overcome various aspects of the path-planning problem are very many. Xiao et al. [8] have proposed one such method that integrates the distance metric from the robot to the target point into the repulsive function model and minimizes its adverse effect on the trajectory towards the target. Some other techniques include repulsive angle deflection and a virtual target point to deal with the local minima problem. Palacín et al. proposed a mobile delivery robot path-planning algorithm for a multi-floor building [9]. The proposed solution utilized a predetermined navigation tree coming from Dijkstra's algorithm to program the robot's pathway, assuming the existence of floor maps, origin and destination points known, self-localization sensors, and remotely controlled elevators. The objective was to better estimate the overall distance of the delivery trip accurately.
Emmi et al. showed the use of the Guiding Manager for the ground mobile robots in agriculture laser-based application designed for weeding instruments [10]. The study indicated that controllers could efficiently work even in different field conditions such as loose soil, stones, and humidity. In this way, they wanted to improve the autonomy and efficiency of their mobile robots in agricultural applications. The typical path planning approaches were studied and developed intensively to take care of the problems related to dynamic environments, moving targets, and complicated terrain. Ojha et al. introduced a real-time obstacle avoidance algorithm for dynamic environments based on the Probabilistic Road Map technique and demonstrated its efficaciousness in determining optimal paths by robots [11]. Rocha et al. carried out an analysis of classical path planning strategies, comparing, and statistically evaluating pros and cons between algorithms for indoor and outdoor environments [12]. Thus, the study necessitates selecting the best appropriate path planning algorithm for individual mission requirements. The focal point of Sun et al. was local path planning for mobile robots using fuzzy dynamic window algorithm [13]. The work was to integrate global path information into the dynamic window approach intended to improve the robots' capability of navigating a variety of environments. The research emphasizes heuristic path planning for augmented autonomy and flexibility of robotic systems in authentic scenarios. Shah et al. have studied employing semantic guesswork generated by language models as a heuristic for planning algorithms into navigation problems [14]. By employing language models to help robots achieve their goals, the authors presented a novel technique to integrate heuristic information from natural language processing into path-planning strategies. The study brings up new potential for enhancing robot navigation using semantic cues as heuristic guidance. By employing heuristic information to path planning algorithms, researchers have been able to optimise navigation strategies, improve efficiency, and boost autonomy in complex

environments. While significant advances have been made in both classical and heuristic path-planning techniques, several challenges remain. Classical approaches, though optimal for static environments, struggle with real-time adaptability in dynamic environments. However, heuristic methods, despite their flexibility, often require high computational resources for effective implementation. The hybrid approach offers a promising balance but faces scalability issues, particularly in large-scale, real-world applications.

The summary of mobile robot path planning literature highlights significant advancements in mobile robot path planning. A key development. However, issues related to real-time adaptability, computational efficiency, and scalability in large-scale, real-world applications remain areas of active research. Table 1 summarises the literature.

While the literature showcases a variety of path-planning algorithms, the effectiveness of these techniques heavily depends on how well the robot's behaviour is modelled. Thus, the next section focuses on robot modelling, which is a fundamental component in understanding and predicting robot movements. The advantages and disadvantages of all the algorithms studied are given in tables 2 and 3.

## 3. Robot Modelling

Modelling holds a special place in mobile robot path planning being the base for the algorithms and simulation empirical and real practical implementation in this algorithm. Because with an accurate model, researchers and engineers can predict and analyze robot behaviour, leading to enhanced and reliable techniques for path planning [15]. However, in path planning, modality is almost entirely about the representation of the motion of the robot concerning its environment. These models serve various essential purposes:

1. Prediction: Models allow us to predict the robot's future state based on current inputs, which is important when planning feasible paths.
2. Simulation: Before implementation in the real world, models allow one to test and validate path planning algorithms extensively in simulated environments without spending as much time and money.
3. Algorithm Development: Currently, most path-planning algorithms rely upon specific robot models, and hence, accurate modelling is important for algorithm effectiveness.
4. Error Analysis: Analysis and measurement of the differences between the planned and executed paths can improve the planning approaches.
5. Constraint Representation: Models can represent the physical limitations of the robot, like maximum velocity or turning radius, so planned paths can truly be executed.

Kinematic and dynamic models are the two types of models used to plan the path of mobile robots. Kinematic models use a geometric perspective of the motion without considering forces, while dynamic models include mass, inertia, and force effects. Generally, the selection of a kinematic or dynamic model depends on the particular application requirements. Often simpler and computationally less expensive to solve, kinematic models are more suited to high-level path planning. While more complex dynamic models are more accurate for robot motion, their benefits come with increased complexity, which is preferable when creating applications that require precise path tracking or when handling high-speed movement.

### 3.1. Kinematic Modelling

Kinematic modeling with the help of practical and theoretical insights is a very important asset to mobile robots for path planning. Kinematic modeling describes the motion of a robot considering no forces or masses [16]. Kinematic modeling forms the backbone of global path planning, that is, planning a global path, and local path planning, namely, finding its way among the immediate obstacles.

Most of the traditional path-planning techniques discussed in other sections of this article, such as potential field methods, roadmap methods, and cell decomposition, make extensive use of kinematic models to verify that the paths planned can actually be executed by the robot. Typically, kinematic models concerning wheeled mobile robots, which are the main objects of this discussion, model the robot in terms of position and orientation within a 2D plane [17].

The kinematic model for a differential drive wheeled robot can be defined by the following equations [18]:

$$\begin{aligned} x' &= v\cos(\theta) \\ y' &= v\sin(\theta) \\ \theta' &= \omega \end{aligned} \qquad (1)$$

Where,

$(x', y')$ is the robot's position

$\theta'$ is the orientation angle

$v$ is linear velocity ω is the angular velocity

These equations form the basis for several path-planning algorithms covered in this paper. They allow researchers to focus on the geometric aspects of motion while abstracting away the complexities of robot dynamics. For robots with different drive systems, such as car-like robots or omnidirectional robots, the kinematic equations may change [19]. For instance, a car-like robot with a minimum turning radius R might be described as:

$$\begin{aligned} x' &= v\cos(\theta) \\ y' &= v\sin(\theta) \\ \theta' &= \frac{v}{R}\tan(\phi) \end{aligned} \quad (2)$$

Where,
$R$ is the minimum turning radius.
$\phi$ is the steering angle.
Kinematic models are highly beneficial in path planning for various reasons:

Simplicity: They are computationally efficient, making them appropriate for real-time planning.
Sufficient for many applications: In cases where dynamics are less important (e.g., low-speed navigation), kinematic models often provide proper precision. Easy integration with sensors: Many sensors immediately offer position and orientation data, which aligns well with kinematic state representations.
However, kinematic models have limitations. They don't account for forces, mass, or inertia, which can become significant factors in high-speed operations, especially when dealing with heavy payloads. In such cases, dynamic modelling becomes important for appropriate path planning and control.

***Table 1.*** *Summary*

| Ref | Environment | Key Features | Challenges Addressed | Results/Findings | Future Research |
|---|---|---|---|---|---|
| [8] | Dynamic | Addresses target inaccessibility and local extremities | Dynamic environmental changes | Demonstrates effective local path planning in dynamic environments using the improved algorithm | Improve efficiency in densely populated areas |
| [9] | Indoor (Multi-Story) | Utilizes a navigation tree and Dijkstra's algorithm | Navigating multi-level buildings | Path planning allows for dynamic recalculation and shortest path determination using a sparse graph | Implement the method using the APR-02 prototype and address door operations |
| [10] | Agricultural | Control architecture for trajectory tracking | Various terrain conditions | Evaluated over different terrains with effective robot orientation and tool use | Validation on diverse platforms and real-time soil condition analysis |
| [11] | Dynamic | Reuses initial paths while addressing dynamic obstacles | Dynamic obstacle presence | Proposes a method for efficient local re-planning and continuity of paths | Testing the approach with various node configurations |
| [12] | Indoor and Outdoor | Compares classical path planning algorithms | Algorithm selection for tasks | Highlights advantages and disadvantages of classical algorithms | Development of a benchmark framework for real-world applications |
| [13] | Dynamic | Improved obstacle avoidance through fuzzy logic | Dynamic environments | Outperforms traditional methods in obstacle avoidance and path planning | Further validation in complex environments |
| [14] | Indoor | Utilizes language models for heuristic navigation | Limitations of cloud-hosted models | Suggests a method for improved navigation using language models | Exploration of applicability in different environments |

### 3.2. Dynamic Modelling

Dynamic modelling is crucial for accurate robot motion representation. It considers forces and torques acting on the robot. This approach is more complex than kinematic modelling. It provides a complete description of robot behaviour.
The general form of a dynamic model for a wheeled mobile robot is [15]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)\tau - A^T(q)\lambda \quad (3)$$

Where,
$\ddot{q}$ is the vector of generalized coordinates
$M(q)$ is the inertia matrix
$(q,\dot{q})$ represents Coriolis and centrifugal forces
$G(q)$ is the gravitational force
$B(q)$ is the input transformation matrix
$\tau$ is the vector of actuator torques
$A^T(q)\lambda$ represents constraint forces

For a differential drive robot, the dynamic model can be simplified to:

$$\begin{aligned} m\ddot{x} &= F_x \\ m\ddot{y} &= F_y \\ I\ddot{\theta} &= \tau \end{aligned} \quad (4)$$

Where,
$m$ is the robot's mass
$I$ is the moment of inertia
$F_x$ and $F_y$ are forces in x and y directions
$\tau$ is the torque
Dynamic models offer several advantages:
1. They account for inertial effects.
2. They consider wheel slip and skid.
3. They allow for accurate trajectory tracking.
4. They enable energy-efficient path planning.

However, dynamic models have limitations:
1. They are computationally expensive.
2. They require precise parameter identification.
3. They can be overly complex for some applications.

In path planning, dynamic models are used for:
1. Generating smooth and feasible trajectories.
2. Optimizing energy consumption.
3. Handling high-speed navigation scenarios.
4. Improving control in uneven terrains.

With a solid understanding of robot modelling, it is essential to explore the criteria that guide the selection of optimal paths. The following section deals with the most important optimisation criteria, such as path length, smoothness, safety degree, and energy efficiency, to guarantee efficient and reliable robot navigation.

## 4. Optimization Criteria for Path Planning

The best-suited solution or ultimate goal for an algorithm designed for a robot is set under optimization criteria in path planning. These criteria are usually general and include shortest way, time or distance taken by the robot for a specific output, energy used while moving, as well as safety of robot itself or its environment.
Optimization criteria prove to be most important factors with respect to path planning for mobile robots since these are the basis by which efficiency, safety, and interference-free navigation are judged in a robot.
Choosing the most appropriate optimization criteria for a specific robot and environment allows developers to formulate path-planning algorithms that fit specific needs of the robot while improving its performance. Various parameters are involved in devising optimization criteria on path planning for mobile robots. In order to identify the optimal strategy, four basic requirements for optimization should be identified, according to [20].

### 4.1 Path Length (d)

Path length optimization is one of the most important and popular criteria in planning the path. This implies that the effort should be made to find the shortest path between the two points with respect to the possible obstacles because it can be defined as finding the path for which the distance or time of travelling from the starting point to its destination is minimized. In most cases, a reduced path length directly leads to a smaller amount of time taken to travel along the path. The most important objective of this is the search for the shortest possible feasible path, which, in turn, means summing the distances of every individual event to the total event in the path from the starting point to its destination. Many methods exist to optimize distance loss: graph methods, the potential field, and sampling methods, to name a few. Each method has its advantages and disadvantages, and the technique chosen would depend on the specific application's need. The formula for this is as follows [21]:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5)$$

In general

$$d = \sum_{j=0}^{n-1} \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \quad (6)$$

Where in (2), $n$ is the number of nodes from the starting location to the goal location.

### 4.2 Smoothness (Sp)

Smoothness is another optimization criterion in path planning, which means finding a path that is not only free of obstacles but also as smooth as possible. A smooth path is preferred as it decreases the wear of the mechanical parts of the robot, increases stability, and simplifies the task of controlling the movements of the robot. In smoothness optimization, the task is to reduce the rate of change of velocity and acceleration along the motion path. The aim is to obtain a pattern that will not produce large jumps in the movement of the robot and will not make the robot unstable. Polynomial interpolation, splines and the Fourier series are among the mathematical strategies that can be applied to perform smoothness optimization. These methods should strive to come up with a path that is free from collision and, at the same time, is smooth, continuous, and differentiable. Smoothness optimization is used together with another criterion, such as the length of a path. For instance, sometimes, a path with minimum path length might not be continuous and therefore, adding another constraint, like a smooth path, could make the path better.
Also, smoothness optimization is considered to be a significant part of path planning because only with

its help can one be sure of the stability, predictability, and safety of the motion of the robot. The smoothness criterion looks for a path that will remain straight as much as possible to do so. This approach is advantageous in the sense that energy is saved, as the mobile robot covers the distance directly, in contrast to a curved path, which requires a lot of energy. The following equation [22] is used to define the smoothness of a path.

$$s_p = \sum_{j=1}^{m} (180^\circ - \theta_j) \qquad (7)$$

Where, in (3), $m$ is the number of angles produced from the starting location to the goal location, $\theta_j$ is the angle value.

### 4.3. Saftey Degree (Sd)

Safety degree optimization criterion in path planning implies minimization of the probability of the occurrence of a collision between the robot and any obstacles in the environment. This criterion is significant because an object may be periodically or irregularly relocated. To prevent the robot from colliding with an obstacle and also to prevent any object that may be nearby from getting harmed in the process, it is necessary to use a path planning mechanism that will guide the robot and keep a safe distance from any obstacles. However, safety degree optimization may encompass other components, such as speed, acceleration, and jerk, in addition to collision avoidance. For example, a path that involves sudden changes in direction or speed may result in instability or damage to the robot or its cargo. Therefore, it is important to design a path that considers the robot's physical capabilities and limitations, as well as any external constraints imposed by the environment or task requirements. Safety degree is also crucial to identify the directions that are free of collision, so it is relevant for environments where robots and humans coexist. Undoubtedly, path planning for safety has even greater importance for mobile robots when it is going to be performed in unknown spaces having a very high degree of freedom. Safety degree, which may be treated as the probability of collision, is defined as follows [21]:

$$s_d = \sum_{p=1}^{p=n} s_p = \begin{cases} 0, & \text{if} d \geq \lambda \\ \sum_{k=1}^{n-1} e^{\lambda - d_k}, & \text{if} d < \lambda \end{cases} \qquad (8)$$

Where in (4), d represents the shortest distance, and $\lambda$ is the safety degree threshold.

### 4.4. Energy Efficiency (E)

Energy efficiency is a crucial factor in mobile robot path planning. It's especially important for battery-powered robots and long-duration missions.
Energy efficiency can be represented as [23]:

$$E = \int_{t_0}^{t_f} P(t)dt \qquad (9)$$

Where, $E$ is the total energy consumed, $(t)$ is the power consumption at time t, $t0$ is the start time, and $t_f$ is the finish time.
The goal is to minimize E while achieving mission objectives.
Factors affecting energy efficiency include:
1. Motor power consumption
2. Path length
3. Terrain characteristics
4. Robot velocity profile

With a clear understanding of the optimization criteria, we proceed to explore the classical path planning techniques. This section presents a detailed analysis of techniques like the Potential Field, Roadmap, and Cell Decomposition techniques. It discusses how they are applied in path planning and their relative advantages and disadvantages.

## 5. Classical Approach Techniques

Classical approaches were initially popular for handling robot navigational tasks since artificial intelligence techniques were not yet developed. These are widely used techniques in mobile robot navigation that are based on pre-existing maps of the environment. This approach is suitable for known environments where the robot has prior knowledge about the environment and obstacles. In this approach, the path planning is performed offline, and the robot then follows the planned path. Potential fields, cell decomposition, and roadmap are the most important classical path-planning techniques.

### 5.1. Potential Field Method

This method generates a fictitious virtual force field that repulses the robot from obstacles and draws it toward the goal. The robot then follows the direction because of the resulting force to reach the goal. Khatib [24] was the first rather to put potential field methods to solve the obstacle algorithm for mobile robots during navigation. The mobile robot be pictured as a point and treated as a particle under the influence of an artificial potential field. The idea is that the robot will be pushed away from repulsive fields (obstacles) toward attractive fields (goal location), Figure 3 applies this concept in regard to a mobile robot navigation [25].
The sensor finds a problem on a surface treatment, then exceeds the re-planning of the coverage path via the artificial potential approach on cost related to that [26]. Nevertheless, the problem in this approach is that it may get stuck in local optimum and hence becomes quite difficult to get the robot out of the dead zone.

In order to get rid of this limitation, proposed to address the problem of getting stuck at local minima by developing a potential field based on path cost and information gain, which helped a robot discover an optimal path for avoiding obstacles and reaching its goals[27]. An artificial potential source was used as a strategy directing formation and terrain coverage for a multirobot system [28]. Simulation results have shown that this method offered superior area coverage, together with real-time planning [29]. This is part of enhancing path planning via the use of the integrated decision tree concept under the artificial potential field method. It was improved obstacle avoidance performance through adaptive artificial potential fields applied to dual-arm robots[30]. It was developed in potential view of artificial fields and augmented reality, a local path planning algorithm to overcome local minima [31]. It was proposed gradient-based methods now with potential fields for navigation of mobile robots for obstacle avoidance problems [32]. Artificial potential fields have been combined with ant colony optimization algorithm which improves the efficiency of path planning and speed convergence [33]. A hybrid parallel-legged walking robot is coming out, the wheel-foot: An improved artificial potential field towards path planning for enhanced efficiency and better obstacle avoidance performance[34]. It was proposed new Membrane Evolutionary Artificial Potential Field (memEAPF) mobile Robot Path Planning method based on the combination of membrane computing with a genetic algorithm and APF technique[35]. The memEAPF constructs parameters to improve path length, smoothness, and safety from a single-layer membrane architecture. It incorporates membrane computing, mimicking compartmentalization and evolutionary principles of living cells, in minimizing path length effectively. The memEAPF that was developed has demonstrated effectiveness in both static and dynamic environments over existing potential-field-based path planning in terms of path length, safety, and computation efficiency. The authors did experimentation in twelve benchmark environments. The evidence was shown proving the memEAPF yields better results over other latest algorithms such as PEAPF, PBPF, and BPF. Not only that, but also the memEAPF algorithm utilizes parallel computing in a significant decrease computation time than all the other methods.

The above approach has made a great impression among researchers, as it makes the platform proficient in handling both static and dynamic obstacles and also does not add many complications. Such a potential field method comes across lots of advantages, like computationally less expensive and easy to implement, while it also faces some limitations by falling in local minima paths, which mostly give suboptimal paths and cannot easily handle complex environments. There is no end to such modifications, and there are hybrid approaches to the potential field method with other techniques that have been suggested by scientists. These researchers have encouraged some good results for the performance and robustness of the potential field method in mobile robot path planning applications.

Algorithm 1 provides pseudocode for the potential field approach. The goal location and obstacles are modeled as potential fields, and the total potential at each point is computed and the robot moves along the negated gradient until it comes to the goal position.

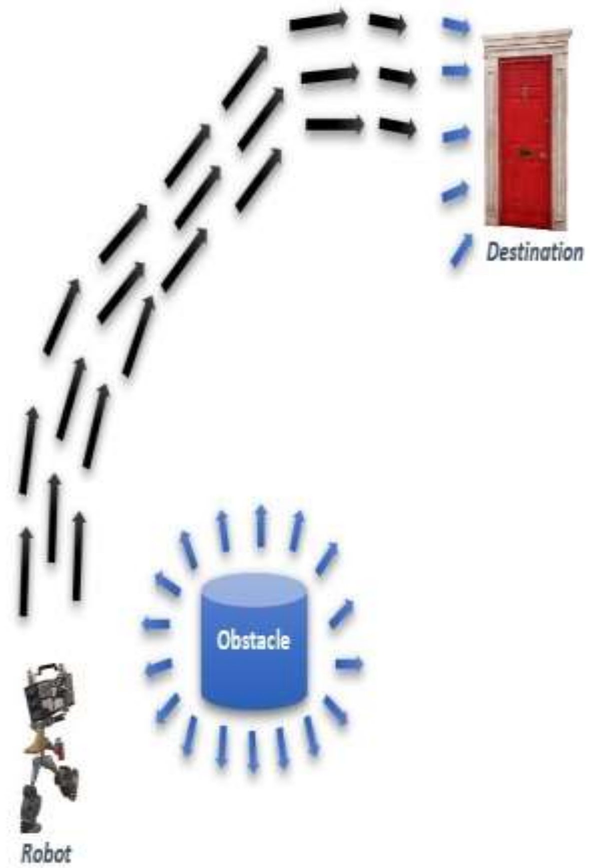


***Figure 3.*** *Artificial Potential Field Illustration*

This algorithm has overall worst-case time complexity O(nm) because of the following:

1. Initialization takes O(1) constant time
2. The outer while loop iterates up to n times depending on the distance to the goal. This contributes O(n) to the complexity.
3. Inside the while loop, the for loop iterates over all m obstacles.

This contributes O(m) to the complexity.

Multiplying these together gives the overall worst case of O(nm). In conclusion, the algorithm has a worst-case time complexity of O(nm) due to the

***Algorithm 1** Potential Field Method*

**Procedure** Potential Field Method
Initialize goalLocation, obstacles[], currentLocation
Initialize goal attraction potential function $U_g()$
Initialize obstacle repulsion potential function $U_o()$
$path \leftarrow$ [currentLocation]
**While** currentLocation $\neq$ goalLocation **do**
  $U_{\text{total}} \leftarrow 0$ Reset total potential
  $U_{\text{att}} \leftarrow U_g(\text{goalLocation}, \text{currentLocation})$
  $U_{\text{total}} \leftarrow U_{\text{total}} + U_{\text{att}}$
  **For each** obstacle **in** obstacles[] **do**
    $U_{\text{rep}} \leftarrow U_o(\text{obstacle}, \text{currentLocation})$
    $U_{\text{total}} \leftarrow U_{\text{total}} + U_{\text{rep}}$
  **End for**
  $F \leftarrow \text{calculateForce}(U_{\text{total}})$
  currentLocation $\leftarrow$ currentLocation $+ F$
  $path \leftarrow \text{append}(path, \text{currentLocation})$
**End while**
**Return** $path$[]
**End procedure**

nested loop structure depending on the number of steps to the goal and the number of obstacles.Let us now examine the key advantages, disadvantages, and rationale for selecting the Potential Field Method (PFM).
1. Advantages: The simplicity and computational efficiency of PFM make it ideal for real-time navigation, especially in environments with static obstacles. It is fast and easy to implement, making it well-suited for embedded systems with limited computational power.
2. Disadvantages: A well-known issue with PFM is the possibility of getting stuck in local minima, where the robot may be unable to reach its goal. This method also struggles in dynamic environments where obstacles can change position unexpectedly, reducing its practical application for real-world mobile robots.
3. Rationale for Selection: Despite its limitations, PFM remains a useful technique for basic obstacle avoidance and serves as a foundation upon which hybrid methods can improve.

### 5.2. Roadmap Method

Roadmaps are made up of a number of pathways which are collision free, and these paths are used to extract a route. Path planning is therefore limited to taking out the path from the start location to the goal location through a sequence of routes [36]. There are more methods to extract the best feasible path, which include visibility graphs and Voronoi. In the visibility graph method, you connect starting location and goal location through a map using nodes. Figure 4 shows a visibility graph with a blue line denoting the path from the start location to the goal location while the black region denotes obstacles [37]. To develop a visibility graph, all obstacles have to be represented as a polygon. Visibility graphs are very useful in environments where the obstacles would take polygonal forms [38]. However, the working of this strategy will be very significantly reduced in a highly dynamic environment. Using visibility graphs in the roadmap methods form the basis of [37]. However, a typical difficulty with visibility graphs is that the generated pathways make collisions with obstacles (polygons), potentially causing robot collisions. Therefore, putting into practice might be difficult. Voronoi diagrams shown in figure 5 are another roadmap approach used for robot path planning and can resolve this problem. Since graph edges will be built at the maximum possible distance from each other adjacent obstacles, thus the robot will be on the safest possible path [39].


***Figure 4**. Visibility Graph*

Algorithm 2 shows the pseudocode for implementing the visibility graph. The Visibility Graph algorithm creates a graph by connecting visible vertices, and then it searches this graph to find the shortest collision-free path. This algorithm has an overall worst-case time complexity of O (n2):
1. Adding obstacle vertices is O(n) where n is the number of obstacle vertices
2. Checking visibility between all pairs is O ($n^2$) with nested loops
3. Finding the shortest path is O ($n^2$) using Dijkstra's algorithm
Therefore, the overall worst-case time complexity is O ($n^2$) due to the nested looping through all vertices to check visibility.

***Algorithm 2** Visibility Graph*

```
Procedure VisibilityGraph(start, goal, obstacles):
V = {start, goal}
For each o in obstacles:
    V = V ∪ Vertex(o)
E = ∅
For each v1 in V:
   For each v2 in V:
      If Visible(v1 , v2 , obstacles):
         E = E ∪ (v1 , v2 )
path = ShortestPath(V, E, start, goal)
Return path
End procedure
```

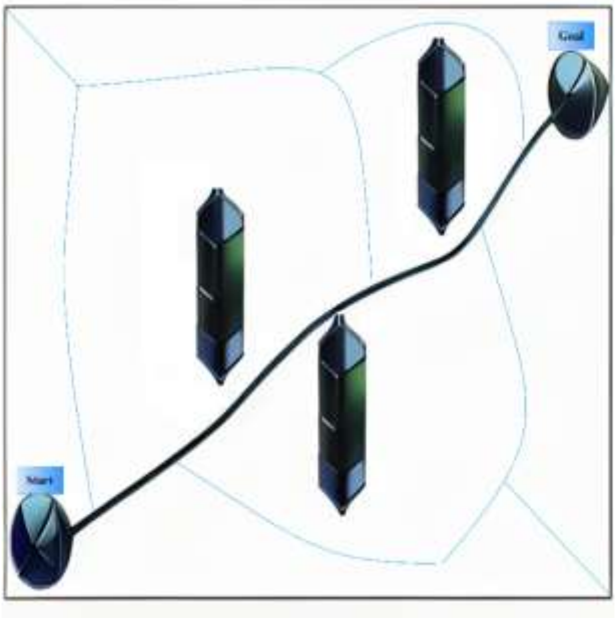


***Figure 5.** Voronoi Diagram*

The Algorithm 3 shows the pseudocode for implementing the Voronoi diagram. The Voronoi Diagram algorithm creates a diagram partitioning the free space based on obstacle distance. This algorithm has an overall worst-case time complexity of O(n log n):

1. Constructing diagram: O(n log n)
2. Finding start/goal regions: O(log n) with point location
3. Traversing regions: O(n) with n regions

Therefore, the overall worst-case time complexity is O(n log n), dominated by the Voronoi Diagram construction. From solving the optimality problem, it suggests a safe path planning method based on Voronoi diagrams according to the words of [40]. In addition, It was suggested new improvements that cause effectiveness in path planning aiming at efficiency and limitations such as abrupt turns and great loops in the Voronoi diagram [40]. It was hybridized the potential field method, visibility graph, and Voronoi diagram to achieve path optimization through hybrid approach [41]. But it is said that this method fails to find an optimized path. For successful path planning, It was proposed a hybrid strategy that uses a visibility graph and a Voronoi diagram to find an optimal path [42]. In an earthquake case, search and rescue missions can be nearly impossible because there can be presence of obstacles and the need to search a large area quickly. Due to the need for better solutions in search and rescue, several algorithms have suggested using probabilistic roadmaps (PRM). One such solution is the grid-based potential field PRM, which has been used in [43] search and rescue applications. It was suggested the integration of Probabilistic Roadmap (PRM) and the Artificial Bee Colony (ABC) algorithms in path planning [44]. This is an objective function that reflects the shortest length, safest, and smoothest pathing because it is safer and easier to move from one point to another without compromising on the total path length. By altering the parameters that control the possible path, including the angles of turns and the distances in between the possible waypoints, the efficiency of the path can be optimized together with the possible obstacles to the way. The integration of the two algorithms has a higher efficiency and reliability in solving problems in robotics and UAV applications. It was aimed at the narrow passages, it uses both particle swarm optimization (PSO) and probabilistic roadmap (PRM) to build the path planning [45]. The approach outlined in [45] involves sharing with the initial sampling points that were placed close to the obstacle the information about open space, and this makes those points to move in any direction in the free space during subsequent movements. This improves the connectivity of the undirected graph without an increment in the overall time taken for sampling. The simulation results indicate that the method proposed improves both sampling point efficiency and the success rate for path planning through narrow passages.

Various forms of modifications and hybrids of the roadmap method are reviewed along with their result as much techniques for adding the efficiency, accuracy, and robustness of this method in various situations. The applications of such a method for path planning have shown good promise when used in either a known, unknown, or dynamic scenario. Other advantages of this approach are its applicability in problems that are high-dimensional and problems with many start and goal points. Some drawbacks are as follows: the construction of the roadmap is computationally expensive, it exhibits poor performance when the working space contains narrow passages and obstacles, and it depends on the choice of roadmap parameters. The reviewed papers offer novel solutions and insights to address these limitations and further enhance the performance of the roadmap method in mobile robot navigation.

***Algorithm 3*** *Voronoi Diagram*

**Procedure** VoronoiDiagram ($start, goal, obstacles$)
$VD \leftarrow ConstructVoronoiDiagram(obstacles)$
$startRegion \leftarrow Region(start, VD)$
$goalRegion \leftarrow Region(goal, VD)$
$path \leftarrow []$
**While** $(currentRegion \neq goalRegion)$ $do$
$currentRegion \leftarrow Neighbor(currentRegion)$
$path.add(Centroid(currentRegion)$
**End while**
**Return path**
**End Procedure**

Let us now examine the key advantages, disadvantages, and rationale for the Roadmap Method.

1. Advantages: These methods efficiently handle environments with complex obstacle configurations. On the other hand, the visibility graph provides a direct path, whilst the Voronoi diagram guarantees that the robot takes the route safest from all surrounding obstacles by maintaining the maximum distance from them.
2. Disadvantages: Visibility maps sometimes generate conflicts with obstacles on the robot's path, while Voronoi diagrams also do not produce optimal paths; thus necessitating subsequent adaptations. The computational complexity of constructing these roadmaps is another challenge, especially in dynamic or high-dimensional spaces.
3. Rationale for Selection: These techniques were included because of their ability to model complex environments and provide reliable path planning for both indoor and outdoor scenarios, particularly in static environments.

### 5.3. Cell Decomposition Method

This strategy partitions the environment into several non-overlapping cells, and it makes use of connectivity graphs in order to join them. The traversal is employed in order to plan a path from the starting location to the target location by checking those occupied cells, which can be identified as pure cells without any obstacles. The pure cell is made up of two new cells separated by the cell containing the obstacles, that is, it is recognized as a corrupted cell, which will then be included in the sequence while finding the optimal path from the starting place to the goal point. Such start and end cells in cell decomposition (CD) method corresponds to the location of the starting location leading to the goal location, as illustrated in figure 6. This strategy is formulated in [46] as a technique for real-time mobile robot path planning. It can be standed to be among these other examples of such method of robot motion planning [47]. Meanwhile, it was proposed a new approximate cell decomposition algorithm [48]. Planning space has been performed to develop a regular grid with a given shape and size [49], thus simplifying the implementation of approximate cell decomposition. It was presented a sensor-based cell CD model combined with a laser scanning strategy to handle mobile robot tasks in unknown environments [50]. The cell decomposition method is a strategy where the workspace is divided into smaller cells and the robot travels through the cells to reach the target location. The robot uses laser scanning to detect the obstacles in the cells and navigate around them. Another Research [51] compared various techniques for path planning based on cell decomposition. Researchers Iswanto et al. [52] used the CD approach together with fuzzy logic to develop a path-planning algorithm for aerial vehicles. The algorithm proposed demonstrates the possibility of using CD and other techniques for path planning for both mobile robots and aerial vehicles. Cell decomposition approaches have been adopted widely in path planning for several categories of vehicles, including unmanned aerial vehicles, for considerations such as shortest path, computation time, memory, safety, completeness, and optimality [53].

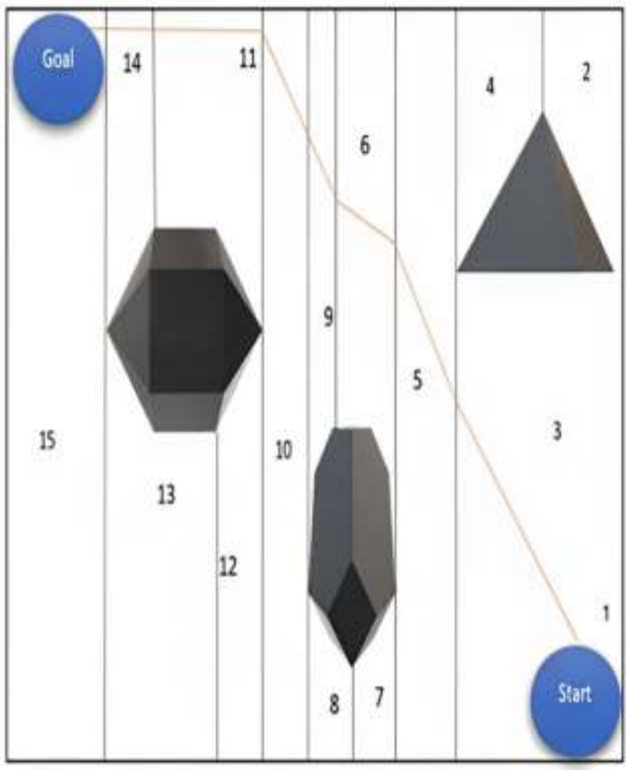


***Figure 6**. Cell Decomposition*

***Algorithm 4*** *Cell Decomposition*

```
Procedure CellDecomposition (robot_pos, goal_pos, obstacles)
Initialize grid_map as empty grid
Mark robot_pos in grid_map
Mark goal_pos in grid_map
For each obstacle in obstacles do
 Mark cells as occupied in grid_map where obstacle is located
End for
Decompose grid_map into cells
Initialize graph G as empty
For each cell in grid_map do
  If cell is free space then
    Add cell as node to G
  End if
End for
For each cell in G do
   For each neighbor of cell do
      If neighbor is reachable then
         Add edge between cell and neighbor in G
      Enf if
   End for
End for
path ← AStarSearchG, robot_pos, goal_pos
Return path
End Procedure
Function AStarSearch (G, start, goal)
Initialize priority queue Q
Add start to Q with priority 0
Initialize distances dictionary
Initialize predecessors dictionary
While Q is not empty do
  current ← Pop minimum from Q
  If current == goal then
  Break
  End if
  For each neighbor of current do
     If neighbor in G then
        dist ← distances[current] + distance(current, neighbor)
      if dist < distances[neighbor] then
        distances[neighbor] ← dist
         Add predecessors[neighbor] ← current
         Add neighbor to Q with priority dist
      End if
     End if
  End for
End while
Return ConstructPath(predecessors, start, goal)
End function
```

The algorithms used in cell decomposition use path planning for mobile robots. Cell decomposition is a method of dividing the environment into non-overlapping cells and connecting them to each other by means of connectivity graphs during path-planning. The advantages of the cell decomposition method are flexibility in the complex environments and highly efficient in terms of method employed. The disadvantages of the method include failing to propagate dynamic obstacles, and dependence on the shape and size of that particular grid which was used to divide the environment. The simulation results for these algorithms depict that they are efficient and really easy to implement. It can handle a complex environment and provide a global path for it.

Algorithm 4 describes Cell Decomposition process such as creating an environmental graph and joining adjacent cells by applying A* path planning involving a search for the minimum path from initial to goal points. The time complexity of the cell decomposition algorithm using A* search can be written as follows:

1. Decomposing the grid map into cells is O(N), where N is the number of cells in the grid map.
2. Creating the graph G of free cells is O(N) since we loop over all cells and add free ones to G.
3. Connecting neighbouring free cells is O(V+E), where V is the number of free cells (nodes), and E is the number of edges between neighbouring free cells.
4. The A* search is O(E + VlogV), assuming a binary heap priority queue is used. This is because each edge is examined once, and priority queue operations are O(logV).

So the overall time complexity is:

O(N) + O(N) + O(V+E) + O(E + VlogV)

= O(N + V + E + VlogV)

Since E is O(V2) in the worst case for a grid graph, this simplifies to:

O(N + V2 + VlogV) = O(N + V2)

Where N is the number of grid cells, and V is the number of free cells.

So, the overall worst-case time complexity is $O(N + V^2)$ for the cell decomposition with the A* path planning algorithm.

Let us now examine the key advantages, disadvantages, and rationale for Cell Decomposition.

1. Advantages: These methods are easy to apply and are capable of solving path-planning problems in complex environments and return global solutions. This makes convergence fast with the potential of simplifying large environments into several cells that can be easily navigated.
2. Disadvantages: Cell decomposition is not so efficient with dynamic obstacles and demands much memory when working with large maps. The technique also offers geometrical and solution accuracy control in terms of a required grid density and overall computational expense.
3. Rationale for Selection: Given the flexibility of this approach in dividing environments into smaller,

navigable sections, it was selected for its applicability to complex but static environments.
Such a classical approach gives the very essence of path planning, having associated both advantages and disadvantages. For example, the Potential Field Method is fast and easy in computation and very good for low powered robots, but traps at local minima makes it unfit mostly for dynamic environments. Roadmap methods are flexible in high-dimensional spaces and guarantee collision avoidance, but present too high computational overheads and suffer from deadlock in narrow passages. By nature, cell decomposition generally provides a simple and easy-to-implement solution for the global path problem, but the algorithm is highly computationally intensive and not very adaptive for changes in the environment. One must bear in mind that classical methods are very good to implement but inflexible in the rapidly changing environment. Therefore, the next section details on heuristic approaches because they are more adapted to handling uncertainties and dynamic environments and can be the alternative way of mobile robot path planning.

## 6. Heuristic Techniques

Heuristic techniques are methods that can be employed to solve problems through efficient search techniques. One example is in path planning, where distances or costs between two nodes are generally computed using heuristics in order to guide a search for the best path from start to destination. Moreover, heuristic algorithms have always been proved superior to conventional ones in terms of path planning of mobile robot because of their efficiency in uncertainty related to the environment. Mobile robot path planning using QAPF learning can be regarded as the improved Q-learning combined with the Artificial Potential Field (APF) improvement by [54]. This overcomes the slowness associated with the traditional methodology of Q-learning and theoretically provides a superior learning rate and efficiency while performing path planning. The QAPF method allowed improvements of 18.83 % in path length efficiency, 169.75% in path smoothness, and 74.84% in training time compared to a Q-learning technique, thus validating the performance of the method in both online and offline path planning applications. In this regard, these results signify the potential of how heuristic-based learning algorithms may further boost path planning in a dynamic environment. It was proposed a new method for AMR path planning, called Membrane Pseudo-Bacterial Potential Field (MemPBPF), which is based on membrane computing, pseudo-bacterial genetic algorithm, and Artificial Potential Field (APF) approach[55]. Hybrid methods exceed standard algorithms in that they are capable of optimizing the performance of time complexity and path length, collision avoidance, and smoothness according to defined performance evaluations. Apart from that, the MemPBPF makes the design scalable for parallel computation on modern hardware such as GPUs. It represents, then, the best hope for use in both static and dynamic environments. Experiments have revealed MemPBPF to be superior in path efficiency, execution time, and overall success when compared to other potential field-based path planning algorithms. Here are the heuristic algorithms.

### 6.1. Artificial Neural Network

An artificial neural network is a smart algorithm constructed with various interlinked layers of processing nodes to offer differing outputs. It is made of three layers that are input, output, and the hidden layer, as shown in figure 7. For processing, the input layers interact with hidden layers first, and then within those hidden layers are connected to the output layer to obtain output. It was studied the application of artificial neural networks for solving problems associated with robotic path planning, thereby showing how they can enhance the performance of such systems [56]. Furthermore, most of the time artificial neural networks find applications when it comes to solving optimization search criteria and pattern recognition because they can reach optimum findings [38]. It was conducted a study on Self Supervised Learning with regard to the learning behavior of a robot while performing a certain task[57]. It was provided a hybrid technique in which a neural network and fuzzy logic work together to utilize both cognitive processes in the navigation of multi-mobile robots in chaotic environments [58]. It was introduced an artificial neural network-based path planner for single planning of multiple robots paths while avoiding obstacles in unknown environments[59]. However, there are certain restrictions attached to the approach of artificial neural networks, training terms associated with it may have to be huge volumes in some cases before obtaining statistically significant results [60]. For example, in supervised learning, minimizing the error between an actual output and its expected output usually becomes a difficult task. In study [61], there is introduction for a new concept known as artificial neural tissue control to the field addressing the coverage problem. Two more or less identical performance-oriented techniques were used as comparison benchmarks for the redesignable coverage with a reconfigurable robot: the feedforward neural network and adaptive neuro-fuzzy inference system as can be seen in [62].

Finally, it was tapped into a fuzzy inference system that does allow exploration of the trade-off between energy and area coverage[63]. In [64], innovative real-time, online path planning methods for highly cluttered and unknown environments were proposed. The suggested method is based on deep neural network (DNN) techniques on the process of developing pathways towards near-optimal path planning. It used a switching scheme and a line of sight check was executed in order to optimize the quality of the path and to enhance the planner's effectiveness further. The neural network approach proposed in [64] predicts the accessible areas where feasible paths might be found within a given environment. Such knowledge is then used in the planning of paths, with considerable improvement in performance. Furthermore, the neural network model here proposed can be extended to typical path planning algorithms. This paper proposed a methodology to plan inspection routes for substation robots [65]. The proposed system is based on an artificial neural network (ANN) system comprising a backpropagation neural network (BPNN), and which performs nonlinear fitting and prediction together with reinforcement Q-learning that incorporates tremendous online learning efficiency.
Simulation results also suggest that the proposed method converges more quickly than the Artificial Potential Field (APF) method and the final value on convergence remains stable and does not vary so much with the best possible solution. The above papers propose various path-planning techniques that combine ANNs with different heuristic algorithms to optimize the path for mobile robots. ANN-based path planners have been developed to plan paths for multi-robots while avoiding obstacles in unknown environments. However, the training data required for producing statistically valid findings can be enormous, and minimizing the error between calculated and expected output is difficult in supervised learning. The results of these studies demonstrate that these hybrid approaches outperform traditional heuristic methods in terms of efficiency and path optimization.

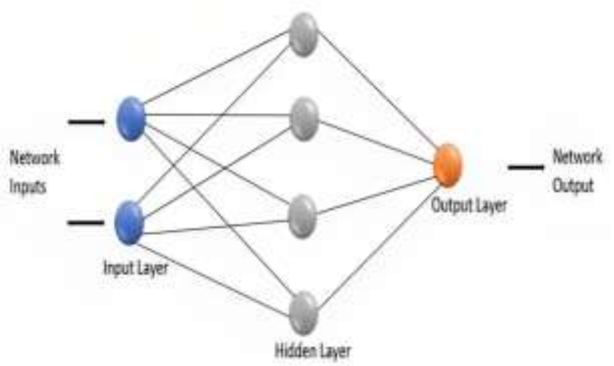


***Figure 7.*** *Artificial Neural Network*

***Algorithm 5*** *Artificial Neural Network*

**Procedure** NNPathPlanning $(start, goal, obstacles)$
$network \leftarrow InitializeNNArchitecture()$
$input \leftarrow EncodeState(start, goal, obstacles)$
$output \leftarrow network.Predict(input)$
$\boldsymbol{While}$ $not\_reached(goal)$ **do**
$current \leftarrow get\_current\_state()$
$input \leftarrow EncodeState(current, goal, obstacles)$
$action \leftarrow network.Predict(input)$
$ExecuteAction(action)$
**End while**
**Function** InitializeNNArchitecture
$network.add(InputLayer)$ Encode state
$network.add(HiddenLayers)$ Fully connected
$network.add(OutputLayer)$ Decode actions
$network.compile()$ Optimizer, loss
$network.fit(training\_data)$ Train network
**Return** $network$
**End Function**

The text that describes algorithms 5 for the pseudo-code for artificial neural networks is all about using that network as robot path planning by training it on predicting actions that are collision-free at different configurations during driving toward a goal state starting from an initial state. These are the two significant portions of the time Complexity: prediction per step and the number of steps to be traveled for reaching the goal.

1. The prediction is done by feeding forward in the network and is O(W) where 'W' is the number of weights in the network.
2. Steps to goal: This is defined based on the size of the environment, and on optimal path length L.

Hence, the overall time complexity per episode is O(W * L).

Let us now look at the advantages, disadvantages, and the rationale for ANN.

1. Advantages: ANN can learn from very complex environments and can improve with time, which qualifies ANN to be a powerful tool in the management of the unknown and dynamism. ANN is best suited for pattern recognition and path generation and can do this efficiently, near to optimal.
2. Disadvantages: ANN needs a huge volume of data for training and very expensive computational resources, which may restrict their suitability for real-time applications. ANNs are also seen to be unpredictable in their performance without adequate training.
3. Rationale for the Selection: ANNs were chosen for their flexibility and robustness in handling dynamic environments, especially in multi-robot navigation scenarios.

### 6.2. Fuzzy logic

Zadeh [66] was the first to introduce the notion of fuzzy sets in 1965. Humans can perceive and complete navigation tasks without the need for accurate calculations. To create a mobile robot, one must imitate this ability [67]. Fuzzy logic has been used to handle various problems in both known and unknown environments [68]. It has been proposed some simulink model with fuzzy logic controller for collision free navigation of mobile robots in unknown environment [69]. Autonomous mobile robot with path planner using fuzzy logic with filter smoothing is incorporated in unknown environment [70]. It has been proposed a hybrid controller able to solve fundamental limitations for path planning where this robot may take the fastest path to reach its destination while avoiding any obstacles on its way[71]. Moreover, it creates available guidelines and decisions toward achieving intended results under various complicated conditions. Based on the simulation outcomes, the employment of the hybrid fuzzy technique suggested to enables the robot to navigate its way past barriers and reach its destination promptly and effectively [71]. It has been presented a layered approach for fuzzy motion planning that focuses on achieving specific goals while avoiding obstacles [72]. The experimentation states that the algorithm does quite well in its goal or target seeking and also avoids its obstacles quite easily along with having good performance in real-time. While reasoning in terms of the environment, the new algorithm improves navigation of robots, thus mostly resembling human reasoning. The path planning of mobile robots is suggested through a grid partition technique in a T-S type fuzzy inference system [73]. By testing various scenarios via the V-REP mobile simulator and model, simulation results prove that the robot is capable of doing any kind of terrain when applying the proposed method. Generally, the simulation results show that T-S type fuzzy inference system path planning model is superior in performance in each parameter. It has been concentrates on performance enhancement of a developing fuzzy logic controller for navigation of mobile robots in complicated environments having more than expected from two dynamic obstacles [74]. Then it can be further improved to handle unknown dynamic obstacles in indoor environments, where it uses sensor's onboard information. As part of the study, simulations in MATLAB were conducted in structured 2D environments to show that the controller achieves good robot navigation in both scenarios, thus proving validity of design [75]. A new approach is introduced for automated reasoning using fuzzy sets of vertical structured General Type-2 version; it is then extended to trajectory-planning of a mobile robot in the presence of dynamic obstacles. Proposed algorithm performance evaluation considered seven benchmark workspaces that accounted for specular reflection and multipath influence of sonar transducer. As determined from the results, the mechanism indeed outperformed other existing mechanisms as far as path length and computational overhead are concerned. It has been introduced a novel predictive control system for mobile robots that is independent of the robot's dynamics and working environment [76]. A Type-3 fuzzy logic system is created to recognize mobile robot dynamics online. The developed predictive approach enhances accuracy and speed of convergence while simultaneously resolving uncertainties and taking control input limits into account. Furthermore, a chaotic-based approach was developed for secure path planning, which generated an unanticipated and complicated reference trajectory suitable for patrol mobile robot applications.

Fuzzy logic is used in many studies related to mobile robot navigation and operates successfully through unknown and dynamic environments. The use of fuzzy logic mimics the ability of human beings to perceive and fill-up its task regarding navigation without having to rely solely on well-calculated determinations. Most of the proposed methods use fuzzy inference systems to analyze the risk inherent in all possible paths while selecting the one that is most appropriate. The advantage of fuzzy logic in path planning includes safe and efficient path generation. On the other hand, the disadvantages include the complexity in the models themselves as well as no easy way to decide what best rules and membership functions will be most appropriate. These fuzzy logic approaches will be promising in planning paths for mobile robots in unknown and dynamic environments.

***Algorithm 6*** *Fuzzy Logic Path Planning*

**Procudure** Fuzzy Logic Path Planning $(start, goal, obstacles)$
$fuzzifier \leftarrow InitializeFuzzifier()$
$rulebase \leftarrow InitializeRulebase()$
$defuzzifier \leftarrow InitializeDefuzzifier()$
$current \leftarrow start$
**While** $not\_reached(goal)$ **do**
$inputs \leftarrow EncodeState(current, goal, obstacles)$
$fuzzy_{inputs} \leftarrow fuzzifier.Fuzzify(inputs)$
$fuzzy_{output} \leftarrow rulebase.EvaluateRules(fuzzy\_inputs)$
$crisp_{output} \leftarrow defuzzifier.Defuzzify(fuzzy\_output)$
$action \leftarrow DecodeAction(crisp\_output)$

```
        ExecuteAction(action)
        current ← get_current_state()
    End while
End procedure
```

Algorithm 6 describes pseudocode illustrated for fuzzy-based path planning, which is a rule-based soft-trained approach to robot path planning based on fuzzy sets and rules. Overall, the time complexity of fuzzy logic path planning algorithm is O(L), in which L is a measure of length from the start to a goal position.This can be derived as follows:

In each iteration of the main loop, the following operations are performed:

1. Encoding the state inputs (O(n), where n is the number of input features).
2. Fuzzification of inputs (O(n)).
3. Evaluating the rule base (O(m), where m is the number of rules).
4. Defuzzification (O(p), where p is the number of discretized output fuzzy sets)
5. Decoding the action (O(1), assuming constant time operation).

Given that n, m, and p are constant (not depending on the growth with the problem size), the algorithm's time complexity for executing each iteration is O(1). The main loop runs until the goal is achieved, and the number of iterations depends on the length of the path denoted by L. Thus, the final run time for the algorithm becomes O(L * 1) = O(L), meaning that it is linear in the length of the path L that lies between the start position and the goal.

Let us now examine the key advantages, disadvantages, and rationale for Fuzzy Logic.

1. Advantages: Fuzzy logic mimics human decision-making, allowing the robot to navigate efficiently in environments with uncertainties. It handles imprecise data well and enables smooth, collision-free navigation.
2. Disadvantages: Designing effective fuzzy rules and membership functions is challenging and often requires domain expertise. Furthermore, its complexity increases with the number of variables involved.
3. Rationale for Selection: Fuzzy logic is ideal for scenarios where precise measurements are not available, making it a valuable addition for uncertain or imprecise environments.

### 6.3. Genetic Algorithm

The genetic algorithm, now a highly popularly recognized search-based optimization technique, has been initially discovered by Bremermann [77] in 1958. However, the first introduction of genetic algorithms to the field of computer science is attributed to Holland [78] in 1975. Genetic algorithms make use of the process whereby all possible solutions to a problem are encoded into chromosomes and undergo some of the following basic processes: selection, crossover, and mutation [79]. A path planning method is based on genetic algorithms that utilize chromosomes with variable lengths. Likewise, it was suggested applying genetic algorithms for navigation with a mobile robot in known environments [80]. Introduced herein [80] is a global path planner that employs genetic algorithms reducing the length of binary strings by converting 2D points to 1D points. As of now, existing studies mainly applied genetic algorithms to navigate in known environments, while it was proposed a method for navigation in the presence of obstacles in an unknown environment [81]. To further enhance robot path planning outcomes, numerous researchers have explored combining genetic algorithms with other intelligent algorithms, leading to the development of hybrid techniques with the potential for improved performance and adaptability. A hybrid methodology combining genetic algorithm and fuzzy logic is provided for the monitoring of moving object [82]. Genetic algorithm has an excellent global search capability for area coverage, but in return it will decrease the stability due to large search space complexity which causes long computation times [83]. In contrast, while the first one [84] discusses the generation of global and multiple local area coverage paths using the Simulated Annealing (SA) algorithm and GA algorithm respectively, the time taken for executing both algorithms in parallel is reduced in terms of computational costs. To overcome the shortcomings of the simple genetic algorithm (SGA) in mobile robot route planning, such as pathways that are too smooth, prone to local optima, and have an unstable algorithm, the improved genetic algorithm (IGA) is proposed [85].

The above papers highlight the importance of evaluating fitness using appropriate cost or fitness functions and exploring different combinations of genetic algorithm parameters to improve the performance of the algorithm. Some studies have proposed hybrid techniques that combine genetic algorithms with other intelligence algorithms to improve outcomes. The advantages of genetic algorithms include their ability to handle complex problems and the potential for global optimization. However, the drawbacks include their reliance on a fixed fitness function and the difficulty in determining the optimal parameters for the genetic operations. Algorithm 7 presents a genetic algorithm in its pseudo-code, a population-based heuristic optimization approach based on the model of natural selection. The genetic algorithm for path planning has an overall time complexity of O(N * L), where

***Algorithm 7*** *Genetic Algorithm for Path Planning*

**Procedure** GAPathPlanning $(start, goal, obstacles)$
$population \leftarrow InitializePopulation()$
$fitness \leftarrow EvaluateFitness(population)$
**While** $not\_converged()$
$new\_population \leftarrow \{\}$
**For** $i \leftarrow 1$ **to** $population\_size/2$ **do**
$parent_1, parent_2 \leftarrow SelectParents(population, fitness)$
$offspring_1, offspring_2 \leftarrow Crossover(parent\_1, parent\_2)$
$offspring\_1 \leftarrow Mutate(offspring\_1)$
$offspring\_2 \leftarrow Mutate(offspring\_2)$
$new_{population} \leftarrow new_{population} \cup \{offspring\_1, offspring\_2\}$
**End for**
$population \leftarrow new\_population$
$fitness \leftarrow EvaluateFitness(population)$
**End while**
$best\_path \leftarrow GetBestIndividual(population, fitness)$
**Return** $best\_path$
**End procedure**

N is the population size, and L is the path length, assuming a constant number of generations. Let us now examine the key advantages, disadvantages, and the rationale for Genetic Algorithm (GA).

1. Advantages: GAs excel in global optimization, offering robust solutions for complex environments. They are capable of finding solutions which are gradually modified over time and does better than many other algorithms in avoiding local minimum traps.
2. Disadvantages: The literature reveals that GAs can be more or less demanding in computation time and may depend on the choice of parameters such as population size and mutation rates.
3. Rationale for Selection: The global search capabilities of GAs make them highly effective for complex environments with many potential obstacles; hence their inclusion in this review.

### 6.4. Ant Colony Optimization

The basic principle of ant colony behavior is that every ant while looking for food will deposit pheromone, an excretion on the way to provide a reference and will sense emissions from other ants. Such pheromones permit ants to communicate and establish routes between them. Also, ants will then, while traveling through the pathways more pheromone-concentrated than certain other paths, naturally reach that destination and add more pheromone in that direction, raising the pheromone concentration even further. After a while, the total concentration of pheromones on the shorter path will then be followed by an increase in the number of ants that adopt it, while the pheromones for the other paths become less and less until finally there are none. In the end, the whole ant colony converges to the best path. The ant colony algorithm flowchart can be visualized in figure 8 [86]. The ant colony algorithm was introduced as a population-based approach by Marco Dorigo [87]. It was proposed avoidance of collision using an ant colony algorithm among multi-robots in a well-known environment [88]. The authors [89] have made some improvements over the algorithm [88] to increase its speed of convergence. It was described ant colony algorithm for the navigation of mobile robots in unknown environments [90]. In addressing mobile robot path planning, it was introduced a novel approach known as the chaotic ant colony system [91]. This method not only outperforms conventional ant colony techniques in terms of success but also enhances the overall scope of global search capabilities. It was proposed a modified version of the Ant Colony Optimization algorithm which incorporates a pheromone updating rule to avoid getting stuck in local minima [92]. This aids in exploring a wider search space and resulting in discovering more optimal solutions. It was made use of the ACO algorithm to achieve optimization in sub-areas coverage based on the distance matrix [93]. The concept is established that, when distances of the sub-areas are taken into consideration, the ACO algorithm should be able to establish the most efficient routes for completing coverage. Such mapping can be employed in various usage, such as environmental monitoring or robotic navigation. It was proposed a travel salesman problem solution based on block sequences' optimization through an ACO algorithm[94]. In this way, the algorithm locates the blocks with such linkage that will minimize overall distance travel by a particular individual thereby increasing the fitness of the overall solution. The global inspection routing optimization, based on the ACO algorithm, is discussed in elsewhere [28]. The algorithm produces optimized inspection routes across industrial environments with many constraints such as the following- minimum path length, inspection time, etc. An improved version of the conventional ant colony algorithm has been proposed for mobile robot path planning in [95] in order to address convergence precision issues and early local optimum trapping. It is the widest and positively reinforced approach in computing among other algorithms in robot path planning. However, stinging limitations include very slow search speed and early convergence. Such advanced terminal distance index-based multi-step ant colony optimization (TDIMSACO) has been recommended for the purpose of mobile robot path planning to upgrade the total efficiency of the ACO [96].

ACO algorithms are actually important because they were used in the following way: to capture all points into consideration, multi-robot path planning was performed in dynamic environments through the population-based technique of ant colony optimization (ACO). The core concept of ACO is that an ant will, upon returning to the bundle, release pheromones along its selected path as a reference for other ants when they venture out again. Ants communicate using pheromones and then autonomously determine pathways. ACO algorithms consider multiple cases, including dynamic environments with moving obstacles and local minima, unknown environments with obstacles and moving obstacles, and multiple robots with time constraints. The advantages of ACO include effective collision avoidance and global search capabilities. However, the disadvantages of ACO include the time-consuming process of releasing and sensing pheromones and the difficulty of handling environments. Some ACO algorithms have been proposed with improvements, such as an adaptive parameter update mechanism, dynamic obstacle prediction mechanism, and hybrid algorithms that combine the genetic algorithm and ACO.

***Algorithm 8*** *Ant Colony Optimization for Path Planning*

**Procedure** ACOPathPlanning $(start, goal, obstacles)$
  $pheromone\_map \leftarrow InitializePheromoneMap()$
  **For** $iteration \leftarrow 1$ **to** $max\_iterations$ **do**
    $ant\_paths \leftarrow \{\}$
    **For** $ant \leftarrow 1$ **to** $num\_ants$ **do**
      $path \leftarrow ConstructPath(start, goal, pheromone\_m$
        $ant_{paths} \leftarrow ant_{paths} \cup \{path\}$
    **End for**
    $pheromone_{map} \leftarrow UpdatePheromoneMap(pheromone\_map, a$
  **End for**
  $best\_path \leftarrow GetBestPath(ant\_paths)$
  **Return** $best\_path$
**End procedure**
**Function**ConstructPath $(start, goal, pheromone\_map, obstacles)$
    $path \leftarrow \{start\}$
    $current \leftarrow start$
    **While** $current \neq goal$ **do**
      $next \leftarrow SelectNextNode(current, pheromone\_m$
        $path \leftarrow path \cup \{next\}$
      $current \leftarrow next$
    **End While**
    **Return** $path$
**End function**

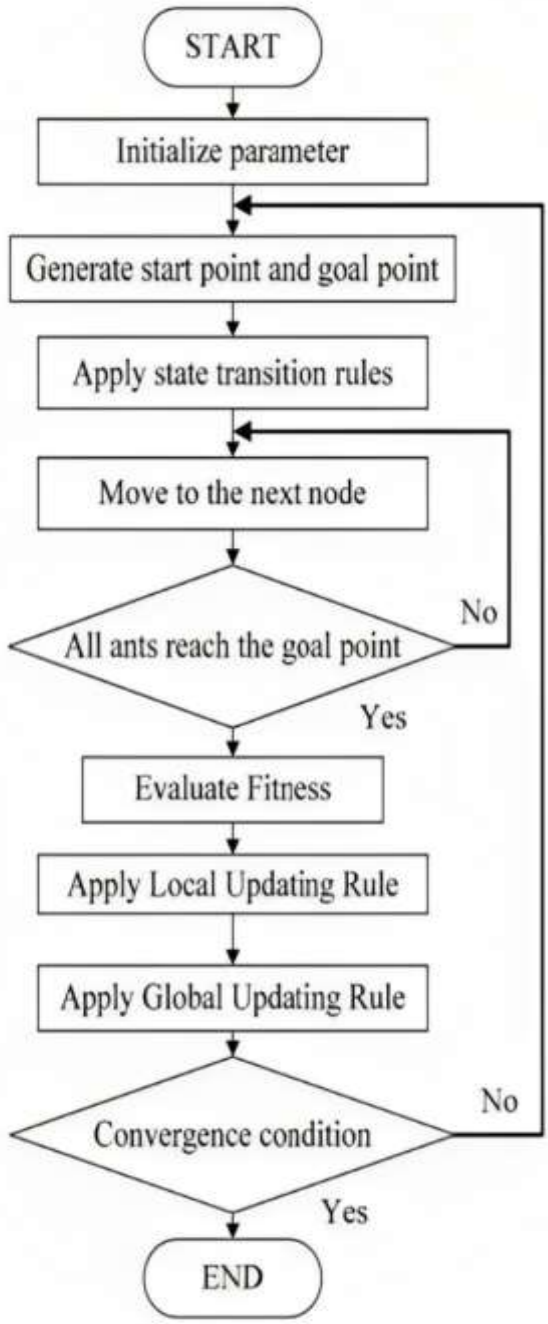


***Figure 8.*** *Ant Colony Algorithm Flowchart [86]*

This shows the pseudo-code of Ant Colony Optimization algorithm in algorithm eight: Time complexity of Ant Colony Optimization algorithm on the whole is O(I * K * N2) for different parameters where N is the number of nodes, K is the number of ants and I is the number of iterations. Now let's discuss the merits as well as demerits of Ant Colony Optimization (ACO)-based methods.

1. Advantages: ACO technique excels in multi-agent systems for its capability to find globally optimal paths through positive feedback mechanisms. ACO is particularly effective in avoiding obstacles and finding the shortest path by optimizing its search space based on prior solutions.

2. Disadvantages: ACO can suffer from slow convergence and is computationally expensive, especially when applied to larger or highly complex environments. The process of depositing and evaporating pheromones requires many iterations to converge, making it time-consuming.
3. Rationale for Selection: ACO is chosen for its strong performance in multi-robot systems and its effective use in dynamic environments, where real-time obstacle avoidance is critical.

### 6.5. Particle Swarm Optimization

A very well-known particle swarm optimization (PSO) was created in 1995 by Eberhart and Kennedy [84]. It imitates the behavior of social animals but does not have a group leader to achieve its aims. For example, when a flock of birds goes out to forage for sustenance, it requires no leaders; birds are just following the nearest bird to the food source. The result, thus, desired is effective communication with the rest of the birds. Recently, PSO is applied to solving robot path planning problems. PSO flowchart is shown in figure 9 [97]. Researchers in [98] used PSO to model a solution for localization in robot navigation in a highly dynamic environment. It was proposed a multi-objective optimization for an obstacle avoidance problem using particle swarm optimization in unknown environment [99]. It was introduced a modified version of PSO for tackling cul-de-sac problems during obstacle avoidance in robots[100]. In the field of multi-robot search optimization, it was applied PSO-based algorithms in their local neighbourhood versions, obtaining results that outperformed the genetic algorithm [101]. Contrasted this method with PSO-based robot obstacle avoidance for known and unknown environment navigation [101]. Particle PSO successfully implemented humanoid robot navigation [103] and Aerial robot navigation in a 3D unknown environment [104]. The optimization is made through PSO method which gives the optimum path globally under the provided user path configuration [105]. It is proved through simulated results that the proposed scheme can generate the robot's path at very short time with no collisions from the start state to the goal state. It was compared applicable methods such as fuzzy logic, neural network, genetic algorithm, and PSO to report their individual findings on the best navigation path and bring out the mixed performance of fuzzy logic with PSO in travelling distance [106]. By employing a set of sampled paths with the PSO mechanism in [107], the cost associated with a coverage path is optimized in terms of quality and efficiency. The global best particle then updates the particle exploration with least cost picked from the camera view, which helps to overcome the problem of premature convergence. However, the time taken for the computations done by this technique is always high even with the above enhancements, especially so in the case of larger model sizes.

Since particle swarm optimization also offers efficient and effective search optimal solutions from the specific search space, several articles have used it in mobile robotics path planning. They have already successfully applied it in different areas, including robotics, such as robotics navigation, obstacle detection and avoidance, and multi-robot search optimization. However, PSO has some drawbacks for instance it is not very effective in dynamic environments. To counter this, many researchers have come up with different modified versions of the PSO algorithms which include the dynamic adaptation of parameters, multi-output fitness function, and a local search. Some of these algorithms also employ other methods, for instance artificial neural networks in its operations. Traditional algorithms are outperformed by these algorithms in path length, obstacle avoidance, and computational efficiency.

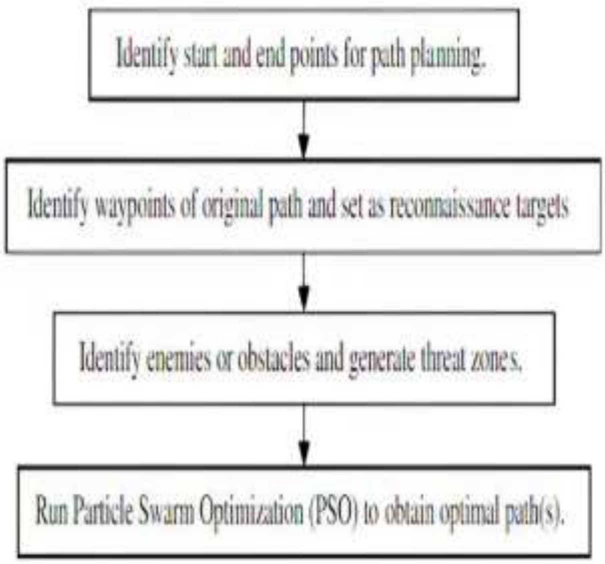


***Figure 9**. Particle Swarm Optimization Flowchart [97]*

Algorithm 9 offers the pseudocode for the particle swarm optimization algorithm. The overall time complexity of the Particle Swarm Optimization algorithm is O(N * L), where N is the number of particles in the swarm, and L is the path length (number of points).

***Algorithm 9** Particle Swarm Optimization for Path Planning*

**Procedure** PSOPathPlanning $(start, goal, obstacles)$
$swarm \leftarrow InitializeSwarm(start, goal, obstacles)$
**For** $iteration \leftarrow 1$ **to** $max\_iterations$ **do**

**For** $particle \leftarrow 1$ **to** $num\_particles$ **do**
$fitness \leftarrow EvaluateFitness(particle.position)$
***If*** $fitness < particle.best\_fitness$ **then**
$particle.best_{position} \leftarrow particle.position$
$particle.best\_fitness \leftarrow fitness$
**End if**
**If** $fitness < swarm.global_{best_{fitness}}$ **then**
$swarm.global_{best_{position}} \leftarrow particle.position$
$swarm.global\_best\_fitness \leftarrow fitness$
**End if**
**End for**
**For** $particle \leftarrow 1$ **to** $num\_particles$ **do**
$particle.velocity \leftarrow UpdateVelocity$
$particle.position \leftarrow UpdatePosition$
**End for**
**End for**
$best\_path \leftarrow swarm.global\_best\_position$
**Return** $best\_path$
**End procedure**

Let us now examine the key advantages, disadvantages, and rationale for Particle Swarm Optimization (PSO).

1. Advantages: PSO is effective in dynamic and noisy environments, providing robust solutions for mobile robot navigation and obstacle avoidance. PSO is computationally efficient and can quickly converge to a solution.
2. Disadvantages: PSO may experience premature convergence, particularly in environments with many local minima. It requires careful parameter tuning to avoid suboptimal solutions. Also, it struggles in highly dynamic environments without modifications that incorporate adaptive strategies.
3. Rationale for Selection: PSO is selected due to its balance between exploration and exploitation in path planning. It is well-suited for real-time applications and environments with moderate uncertainty.

### 6.6. Bacterial Foraging Optimization

A unique nature-inspired optimization methodology based on E.coli and M. Xan bacteria was proposed by Passino in 2002 [108]. These microorganisms hunt nutrient sources as they elevate the energy yield per unit time. According to Xiao-dan Liang et al., [109], such a robot bears semblance to the bacterium, governing optimal movement gratuitous of obstacle between starting and destination points in an ambit bounded by obstacles.

It was first used bacterial foraging optimization for mobile robot navigation in a known environment [110]. It was enhanced classical bacterial foraging optimization, adding improvements to the planning of paths for wheeled robots [111]. It was derived a bacterial foraging optimization technique in a multi-robot navigation environment [112].

The bacterial foraging optimization technique is hence very applicable to path planning because it uses a non-linear fitness function for decision making [113]. Hence, it can be executed in a short time in a realistic complex real-world environment.

The bacterial foraging optimization (BFO) algorithm is an optimization algorithm that draws its inspiration from nature and is used in mobile robot path planning applications. It is found to be efficient in finding the optimum path, whether in a predetermined or unknown environment or in a dynamic environment that may even change while the path is being drawn due to the movements of some obstacles.

However, some of the disadvantages are the use of sensors for the mapping and detection of obstacles and the need for a non-linear fitness function that could involve an extensive amount of computation. Improvement over BFO has been suggested by a number of researchers, such as the introduction of new operators to enhance searchability and convergence speed or modifications to how bacteria update their positions by taking into consideration the information on the best possible position. The experimental results showed that these modified algorithms outperformed existing ones.

Algorithm 10 shows the pseudo-code for the Bacterial Foraging Optimization algorithm.

Time Complexity Analysis:

Let N be the population size, L be the path length (number of points), I be the number of iterations, C be the maximum number of chemotaxis steps, and R be the maximum number of reproduction steps.

Population Initialization: O(N * L)

Fitness Evaluation (per bacterium, per iteration): O(L) (assuming path cost is computed in linear time)

Chemotaxis Loop (per bacterium, per iteration): O(C * L)

Reproduction Loop (per bacterium, per iteration): O(R * L)

Elimination-Dispersal Loop (per bacterium, per iteration): O(L) The total time complexity for one iteration is O(N * (C * L + R * L + L)).

Since the algorithm runs for I iterations, the overall time complexity is O(I * N * (C * L + R * L + L)).

In practice, the number of chemotaxis steps C, reproduction steps R, and iterations I are often limited by maximum values or convergence criteria so that the complexity can be approximated as O(N * L).

***Algorithm 10*** *Bacterial Foraging Optimization for Path Planning*

```
Procedure BFOPathPlanning (start, goal, obstacles)
  population ← InitializeBacterialPopulation()
  fitness ← EvaluateFitness(population)
  For iteration ← 1 to max_iterations do
    new_population ← {}
    For bacterium ← 1 to population_size do
      new_position ← ChemotaxisLoop(bacterium.position)
      new_position ← ReproductionLoop(new_position)
      new_position ← EliminationDispersalLoop(new_position)
      new_population ← new_population ∪ {new_position}
    End for
    population ← new_population
    fitness ← EvaluateFitness(population)
  End for
  best_path ← GetBestIndividual(population, fitness)
  Return best_path
Function ChemotaxisLoop(position)
  For step ← 1 to max_chemotaxis_steps do
    new_position ← TumbleOrSwim(position, fitness)
      if fitness(new_position) < fitness(position) then
        position ← new_position
      End if
  End for
  Return position
End function
Function ReproductionLoop position
  new_position ← ReproduceOffspring(position, fitness)
  return new_position
End function
Function EliminationDispersalLoop(position)
  new_position ← EliminateOrDispersePosition(position, fitness)
  return new_position
End function
```

Let us now examine the key advantages, disadvantages, and rationale for Bacterial Foraging Optimization (BFO).

1. Advantages: BFO can balance exploration and exploitation, preventing the algorithm from getting trapped in local minima.
2. Disadvantages: The convergence rate of BFO is relatively slow, particularly in environments with many local minima. Also, BFO relies on a non-linear fitness function, which can require significant computational resources, especially when applied to larger environments or complex tasks.
3. Rationale for Selection: BFO was chosen for its ability to handle non-linear, multi-objective optimization problems in dynamic environments, making it highly suitable for mobile robot navigation tasks in unknown environments.

The advantages and disadvantages of heuristic approaches are presented in table 3.

***Table 2.*** *Advantages & Disadvantages of Classical Approach*

| S.No | Technique | Advantages | Disadvantages |
|---|---|---|---|
| 1 | Potential Field Method | Simple and easy to implement, requiring no complex algorithms. It also has relatively simple computations, making it suitable for low-powered computers or embedded systems. Moreover, it is fast and efficient, making it ideal for applications that require speed. | One of the major drawbacks is the possibility of encountering local minima, leading to the algorithm getting stuck in a particular region without finding an optimal solution. Additionally, it does not consider dynamic changes in the environment, such as obstacles or moving objects, which may cause unexpected results. |
| 2 | Roadmap Method | Requires less time to reach goal location using Visibility graph and good collision avoidance capability in Voronoi graph method. Moreover, the roadmap method can handle complex and high-dimensional spaces and multiple start and goal points. It also has good computational efficiency. | The robot collision problem in Visibility Graph approach, and non-optimality and non-convergence in Voronoi Graph method. Moreover, some other limitations include the computational complexity of constructing the roadmap, the difficulty of handling narrow passages and obstacles, and sensitivity to the selection of roadmap parameters. |
| 3 | Cell Decomposition Method | Very fast convergent. Straightforward to implement. It can handle complex environments and can provide a global path. | High-dimensional configuration spaces require intensive computation. It also requires a lot of memory and can be difficult to scale for large environments. |

***Table 3.*** *Advantages & Disadvantages of Heuristic Approach*

| S.No | Technique | Merits | Demerits |
|---|---|---|---|
| 1 | Neural Network Technique | It is a commonly used path planning technique to learn and model complex and nonlinear relationships.<br>Good at learning complex, non-linear relationships between inputs and outputs.<br>Can generalize well to unseen data.<br>Can handle noisy or incomplete data.<br>Can be trained using online or offline methods.<br>Can learn from multiple inputs and multiple outputs simultaneously. | Requires a lot of data to train accurately.<br>Can be slow to train.<br>Can suffer from overfitting or underfitting if not properly tuned.<br>Can be difficult to interpret how the network arrived at a particular decision.<br>Time complexity is high, and convergence occurs very soon.<br>Moreover, developing a neural network architecture to describe a dynamic environment is challenging. |
| 2 | Fuzzy Logic Technique | Can handle imprecise or uncertain data.<br>Can deal with incomplete or ambiguous information.<br>Can be applied to a wide range of problems.<br>Can be easily integrated with other techniques.<br>Provides a transparent and interpretable way of decision making.<br>It is ideal for complex autonomous mobile robots. | Can be computationally expensive.<br>Requires domain expertise in order to define the fuzzy sets and rules properly.<br>Can be sensitive to changes in input scaling or fuzzification methods.<br>May require tuning of the fuzzy sets and rules in order to achieve optimal results.<br>May not be suitable for problems where precise control or decision-making is required. |
| 3. | Genetic Algorithm | It is robust, and it has high search efficiency.<br>Can handle non-linear and non-differentiable functions.<br>Can find the global optimum rather than getting trapped in a local optimum.<br>Can provide a diverse set of solutions.<br>Can handle noisy or incomplete data.<br>Can work well for large, complex problems.<br>Moreover, it is also able to solve problems with multiple objectives. | Can be computationally expensive.<br>Can get trapped in local optima if the population size or mutation rate is not properly tuned.<br>Can take a long time to converge on an optimal solution.<br>May require a large amount of computational resources in order to explore the solution space effectively.<br>Can suffer from premature convergence if diversity is not maintained in the population. |
| 4 | Ant Colony Optimization | Can handle non-linear and non-differentiable functions.<br>Can find the global optimum rather than getting trapped in a local optimum.<br>Can handle multiple objective functions.<br>Can provide a diverse set of solutions.<br>Can work well for large, complex problems.<br>Can adapt to dynamic environments. | Can be computationally expensive.<br>May require a large number of iterations to converge on an optimal solution.<br>May require domain expertise in order to set the parameters appropriately.<br>Can be sensitive to changes in the environment or problem formulation.<br>Can suffer from premature convergence if diversity is not maintained in the population. |
| 5 | Particle Swarm Optimization | Can handle non-linear and non-differentiable functions.<br>Can find the global optimum rather than getting trapped in a local optimum.<br>Can handle multiple objective functions.<br>Can provide a diverse set of solutions.<br>Can work well for large, complex problems.<br>Can adapt to dynamic environments. | Can be computationall expensive. May require a large number of iterations to converge on an optimal solution.<br>May require tuning of the swarm parameters in order to achieve optimal results.<br>Can be sensitive to changes in the environment or problem formulation.<br>Can suffer from premature convergence if diversity is not maintained in the population. |
| 6 | Bacterial Foraging Technique | Efficient optimization of multimodal, non-convex, and noisy functions.<br>Can handle various optimization problems, including multi objective optimization.<br>Requires fewer evaluations of the objective function, computationally cost-effective.<br>Balances between exploration and exploitation of the search space to prevent getting stuck in local minima. | Slow convergence rate, especially in complex problems with many local minima.<br>Lacks a strong theoretical foundation, which can make it difficult to understand and analyze the behaviour of the algorithm.<br>Performance can be sensitive to parameter values, and the optimal values can be problem-dependent. |

While heuristic techniques demonstrate substantial advantages, particularly in dealing with dynamic and uncertain environments, they also introduce complexities such as high memory and computational costs. These factors make it essential to evaluate their effectiveness alongside classical techniques. The following section discusses how these methods compare in terms of real-world applicability, highlighting their strengths and weaknesses across different environments.

## 7. Discussion

The comparison of heuristic and classical approaches is presented in table 4. The implementation of classical approaches is easy. However, classical approaches have several drawbacks, including the inability to handle maximum uncertainty, entrapment in local minima, the need for exact environmental information, and so on. When using the classical technique, there is always a question about whether a solution will be found or whether such a solution exists. However, many researchers have attempted to improve the classical algorithms. However, these strategies do not outperform heuristic approaches in real-world problems. Moreover, classical approaches require initial information about the working environment. So, they are often deployed for navigation in a known environment. Furthermore, classical algorithms have low memory requirements, which makes them common and appropriate algorithms for low cost mobile robots. On the contrary, heuristic techniques that can deal with high levels of uncertainty are used for navigation in unknown environments. Heuristic techniques are better than classical techniques. However, they also have several disadvantages, such as being highly time-consuming and large memory requirements, which are incompatible with low-cost robots. But in terms of pure classical approaches, the potential field method outperforms roadmap and cell decomposition methods in known environments for navigation. Still, the hybrid approach of cell decomposition has been used significantly more than that of potential field and roadmap method in recent years. Today, heuristic techniques are gaining more popularity and usage than classical techniques used in indoors or outdoors because they could be very efficient in known and unknown environments. From these, however, many heuristic approaches were developed and used in an unknown environment where dynamically moving obstacles are present. Moreover, newly developed heuristic methods such as bacterial foraging optimization algorithm and particle swarm optimization have also been successful in an unknown environment where the movement of obstacles is dynamic. Many researchers have used the hybrid approach to deal with complex problems. Additionally, the review finds the establishment of hybridization of algorithms with the most suitable optimization criteria to be the most convenient way of performance enhancement. Nevertheless, it may be possible to achieve various optimal results by hybridizing algorithms. Rather, challenges may arise when attempting to hybridize incompatible approaches. The yield from combining two incompatible algorithms could be even worse than employing each separately.

Combined approaches have come up with state-of-the-art mobile robot techniques for navigation which have paved way for drastic improvement in path planning as far as dynamic and unknown environments are concerned. However, existing techniques like Potential Field Method (PFM), Roadmap Methods and Cell Decomposition are quite static in nature, and they show limitations in dynamic environments like local minima as well as having large computational complexity in high-dimensional spaces. Innovations in hybrid methods using heuristic algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Ant Colony Optimization (ACO) have shown exceptional promise in making such configurations adaptable, computationally efficient and globally optimizing compared to classical methods. Such a technique under consideration is Membrane Evolutionary Artificial Potential Field (memEAPF), which combines membrane computing with genetic algorithms. This technique surpasses the classical methodologies, such as PFM and Cell Decomposition, in dealing with local minima very effectively while providing a very long path and enhanced safety and computational efficiency. The memEAPF technique surpasses other methods in case of dynamic environments because it catered for real-time planning constraints and lesser computation times. PSO has also been hybridized with Artificial Neural Networks (ANNs) to even augment the adaptability and performance efficiency of navigation systems within dynamic environments. PSO effectively and globally optimizes such navigation applications in the resolution of cul-de-sac type of problems typically encountered during the obstacle avoidance tasks. ANNs learned with dynamic environments typical of using investment strategies over time have thus greatly enhanced these systems' efficiency and applicability for various uses in mobile robot navigation. The newest development also incorporates Fuzzy Logic systems with PSO, showing great promise for optimized navigation paths even in unknown environments; for example, such fuzzy systems can be used to optimize path efficiency because they can treat uncertainty or imprecise data very well. After discussing the of classical and heuristic approaches, it is clear that choosing the right method is crucial for effective robot navigation. The following conclusion will summarize these insights and suggest areas for future research.

## 8. Conclusion and Future Research Directions

So, this paper has studied classical and heuristic path-planning techniques of mobile robots in known and unknown environments from comprehensive

***Table 4** Comparison of Classical and Heuristic Path Planning*

| S.No. | Criteria | Classical Path Planning | Heuristic Path Planning |
|---|---|---|---|
| 1 | Efficiency in Known Environments | High | Medium |
| 2 | Efficiency in Unknown Environments | Low | High |
| 3 | Ability to handle uncertainty | Limited | High |
| 4 | Ability to avoid local minima | Limited | High |
| 5 | Application Range | Limited | Wide |
| 6 | Intelligence | Low | High |

perspectives. Our findings say that path-planning algorithms should be selected for a particular operation and needs to stress on the parallelism to be balanced against computation efficiency, adaptability, and real-time operation in dynamic environments. Classical algorithms are found to be well-suited for static and structured environments, delivering computationally efficient solutions. However, their limitations in real-time adaptability and dealing with dynamic obstacles limit their applicability in unknown environments. Heuristic approaches demonstrated better results in environments with high uncertainty. These algorithms are exceptional in solving any problem and are able to emerge with strong solutions in dynamic and unstructured environments in which conventional methods cannot provide a solution. Their computational intensity can, however, pose hurdles for applications intending to operate in real-time and at a large scale. Again, the paper emphasized that more studies were continuing to show the increasing demand for hybrid approaches comprising classical and heuristic methods. Hybrid approaches not only balance the computing efficiency of classical algorithms with the adaptability of heuristic methods, but they also offer increased performance in terms of path optimisation, obstacle avoidance, and scalability. These techniques provide an appropriate solution for applications in autonomous robots, multi-robot coordination, and complex industrial environments where dynamic, real-time decision-making is crucial. The paper also emphasised the importance of optimisation criteria such as path length, smoothness, safety, and energy efficiency. These criteria are particularly significant in mission-critical applications such as search-and-rescue, where the ability to navigate unknown environments quickly and safely can be a matter of life and death. In industrial automation, energy efficient path planning becomes essential, particularly for long-duration or resource-constrained missions. The future of mobile robot navigation lies in scalable, adaptive, and computationally efficient solutions capable of navigating the increasing complexity and dynamism of real-world environments. The scope for further research in mobile robot path planning holds tremendous potential for advancing the field of robotics. The following key areas represent promising avenues for future investigation and development:

1. Real-world Implementation and Testing: Future research should focus on bridging the gap between simulation and real-world application. This includes building and testing algorithms in complex, unpredictable environments that closely replicate the real-world scenario. Researchers should examine aspects such as different terrain, dynamic obstacles and changing weather conditions. Long-term studies in different environments will be important to validate the robustness and adaptability of path-planning algorithms.
2. Multi-robot Systems and Swarm Intelligence: As the deployment of numerous robots becomes more frequent, there’s a pressing need for advanced algorithms that can efficiently manage swarms of robots. This research direction should examine decentralized decision-making processes, task distribution mechanisms, and collision avoidance among robots. Investigators should also investigate ways to maximize the collective behaviour of robot swarms to achieve difficult tasks that single robots cannot accomplish.
3. Hybrid Algorithms and Fusion of Techniques: The development of hybrid algorithms that combine the strengths of classical and heuristic approaches is a viable route for research. Scientists should research creative approaches to merge algorithms, which could lead to algorithms that are both computationally efficient and capable of handling complex, dynamic environments.
4. Advanced Machine Learning incorporation: The incorporation of advanced machine learning techniques into path planning algorithms offers remarkable possibilities. Research should focus on utilizing reinforcement learning for adaptive path planning, applying deep learning for increased environmental understanding, and studying transfer learning to apply knowledge obtained in one environment to different environments. The development of algorithms that can learn and improve their performance over time in varying conditions will be particularly valuable.

5. Energy-aware and Resource-efficient Planning: As mobile robots are increasingly deployed for long-duration missions, energy conservation becomes critical. Future research should build algorithms that not only optimize for path length and safety but also account for energy consumption. This could require establishing precise energy models of robots, incorporating terrain-dependent energy costs, and developing multi-objective optimization techniques that balance energy efficiency with other performance objectives.
6. Human-Robot Interaction and Social Navigation: With robots increasingly working in human-populated situations, research into socially aware navigation is essential. This entails designing algorithms that can predict human behaviour and handle dynamic environments without causing discomfort to humans. Researchers should also explore how robots can effectively communicate their intentions to humans and how human feedback can be incorporated into path-planning decisions in real-time.
7. Cognitive Mapping and Semantic Understanding: Future research should focus on enabling robots to create more sophisticated representations of their environments. This includes developing algorithms for semantic mapping, where robots not only build geometric maps but also understand the function and context of different spaces. Research in this area could lead to improved decision-making in path planning, allowing robots to prioritize paths based on a better understanding of their environment.
8. Bio-inspired and Novel Optimization Techniques: There's significant potential in investigating new bio-inspired algorithms for path planning. Researchers should examine algorithms inspired by various biological systems beyond the typically employed ant colony or particle swarm optimizations. This could include studying the navigation strategies of migratory birds, the foraging behaviour of other insects, or even the decision-making processes in microbial communities.
9. Integration with Emerging Technologies: Future studies should also investigate how path planning algorithms might integrate with and take advantage of emerging technologies. This involves examining how 5G and future communication technologies will enable more effective multi-robot coordination, how edge computing can be used for more responsive planning, and how developments in sensor technologies can be incorporated to increase environmental perception and mapping.

By pursuing these research directions, the field of mobile robot path planning can make significant advancements towards developing more intelligent and efficient robots capable of functioning independently in increasingly complex and dynamic environments.

## Author Statements:

- **Ethical Approval:** The conducted research does not include either human or animal use.
- **Conflict of Interest:** The authors declare there is no actual or potential conflict arising from any financial, personal, or other relationships with other people or organizations associated with this paper.
- **Acknowledgment:** The authors state that they have nobody or no company to acknowledge.
- **Author Contributions:** The authors hereby claim equal rights on this paper.
- **Funding Information:** The authors declare there are no fundings to be acknowledged.
- **Data Availability Statement:** The data that support the findings of this study are available from the corresponding author on reasonable request. The data are not publicly available due to privacy or ethical restrictions.

## References

[1] Nasti SM, Chishti MA. (2024). A Review of AI-Enhanced Navigation Strategies for Mobile Robots in Dynamic Environments. In: 2024 *ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS*);p. 1239–1244.
[2] Basavanna M, Shivakumar M. (2019). An Overview of Path Planning and Obstacle Avoidance Algorithms in Mobile Robots. *International Journal Of Engineering Research & Technology (IJERT*)8(12). DOI:10.17577/IJERTV8IS120252
[3] Raheem FA, Raafat SM, Mahdi SM. In: Furze JN, Eslamian S, Raafat SM, Swing K, editors. Robot Path-Planning Research Applications in Static and Dynamic Environments Cham: Springer; 2022. p. 291–325.
[4] Djekoune AO, Achour K, Toumi R. (2009). A Sensor Based Navigation Algorithm for a Mobile Robot Using the DVFF Approach. *International Journal of Advanced Robotic Systems* 6(2). DOI:10.5772/6797
[5] Raja P, Pugazhenthi S. (2012). Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences* 7(9):1314–1320.
[6] Zafar MN, Mohanta JC. (2018). Methodology for Path Planning and Optimization of Mobile Robots: A Review. *Procedia Computer Science* 133:141–152. International Conference on Robotics and Smart Manufacturing (RoSMa2018).
[7] Mester G. Applications of Mobile Robots. In: 7th International Conference on Food Science; 2006. .
[8] Xiao X, Tian W. Mobile robot based on artificial potential field method Path planning method research. In: Dong H, Yu H, editors. *Ninth International Conference on Mechanical Engineering, Materials, and Automation Technology (MMEAT 2023),* vol.

12801 International Society for Optics and Photonics, SPIE; 2023. p. 128015K. https://doi.org/10.1117/12.3007127.

[9] Palacín J, Rubies E, Bitriá R, Clotet E. (2023) Path Planning of a Mobile Delivery Robot Operating in a Multi-Story Building Based on a Predefined Navigation Tree. *Sensors* 23(21). https://www.mdpi.com/1424-8220/23/21/8795.

[10] Emmi L, Fernández R, Gonzalez-de Santos P. (2024)An Efficient Guiding Manager for Ground Mobile Robots in Agriculture. *Robotics* 13(1). https://www.mdpi.com/2218-6581/13/1/6.

[11] Ojha P, Thakur A. Real-Time Obstacle Avoidance Algorithm for Dynamic Environment on Probabilistic Road Map. In: 2021 *International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA);* 2021. p. 57–62.

[12] Rocha L, Vivaldini K. Analysis and Contributions of Classical Techniques for Path Planning. In: 2021 Latin American Robotics Symposium (LARS), *2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE);* 2021. p. 54–59.

[13] Sun Y, Wang W, Xu M, Huang L, Shi K, Zou C, et al. (2023). Local Path Planning for Mobile Robots Based on Fuzzy Dynamic Window Algorithm. *Sensors* 23(19). https://www.mdpi.com/1424-8220/23/19/8260.

[14] Shah D, Equi M, Osinski B, Xia F, Ichter B, Levine S, (2023). Navigation with Large Language Models: Semantic Guesswork as a Heuristic for Planning; https://arxiv.org/abs/2310.10103.

[15] Siegwart R, Nourbakhsh IR, Scaramuzza D. (2021) Introduction to autonomous mobile robots. The MIT Press, Massachusetts Institute of Technology. Accessed: 2024-9-3.

[16] Tzafestas SG. (2023) Introduction to Mobile Robot Control. Philadelphia, PA: Elsevier Science Publishing.

[17]LaValle SM. (2006). Planning Algorithms. Cambridge, England: Cambridge University Press.

[18] Corke P. (2017). Robotics, vision and control. 2 ed. Springer tracts in advanced robotics, Cham, Switzerland: Springer International Publishing.

[19] Campion G, Bastin G, D'Andrea-Novel B. (2006) Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE transactions on robotics and automation* 12(1):47–62.

[20] S. M. Nasti and M. A. Chishti, (2024). Framework for Logistics During Hajj Using Autonomous Mobile Robots," 2024 *1st International Conference on Logistics (ICL),* Jeddah, Saudi Arabia, pp. 1-5, doi: 10.1109/ICL62932.2024.10788585.

[21] Wang XY, Zhang GX, Zhao JB, Rong H, Ipate F, Lefticaru R. (2015). A modified membrane-inspired algorithm based on particle swarm optimization for mobile robot path planning. *International Journal of Computer Communications and Control* 10:732–745.

[22] Hidalgo-Paniagua A, Vega-Rodriguez MA, Ferruz J, Pavón N. (2015) MOSFLA-MRPP, multi-objective shuffled frog-leaping algorithm applied to mobile robot path planning. *Engineering Applications of Artificial Intelligence* 44:123–136.

[23] Mei Y, Lu YH, Hu YC, Lee CSG.(2004) Energy-efficient motion planning for mobile robots. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04.* 5 IEEE. p. 4344–4349.

[24] Khatib O. (1985) Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings - IEEE *International Conference on Robotics and Automation,* 2. p. 500–505.

[25] Garibotto G, Masciangelo S. Path planning using the potential field approach for navigation. In: Fifth international conference on advanced robotics Pisa, Italy; 1991. p. 1679–1682.

[26] Nykolaychuk M, Ortmeier F. In: Liu H, Kubota N, Zhu X, Dillmann R, Zhou D, editors. Coverage path re-planning for processing faults, vol. 9245 of *Lecture Notes in Computer Science Cham*, Switzerland: Springer; 2015. p. 358–368.

[27] Wang C, Meng L, Li T, Silva CWD, Meng MQH. Towards autonomous exploration with information potential field in 3D environments. In: Proc. *18th Int. Conf. Adv. Robot. (ICAR) Hong Kong;* 2017. p. 340–345.

[28] Huang C, Li W, Xiao C, Liang B, Han S. (2018). Potential field method for persistent surveillance of multiple unmanned aerial vehicle sensors. *International Journal of Distributed Sensor Networks* 14(1):1–13. https://doi.org/10.1177/1550147718755069

[29] Lin X, Wang ZQ, Chen XY. Path Planning with Improved Artificial Potential Field Method Based on Decision Tree. In: 2020 27th Saint Petersburg *International Conference on Integrated Navigation Systems (ICINS);* 2020. p. 1–5.

[30] Zhang H, Zhu Y, Liu X, Xu X. (2021). Analysis of Obstacle Avoidance Strategy for Dual-Arm Robot Based on Speed Field with Improved Artificial Potential Field Algorithm. *Electronics* 10(15). https://www.mdpi.com/2079-9292/10/15/1850.

[31] Szczepanski R, Bereit A, Tarczewski T. *(*2021) Efficient Local Path Planning Algorithm Using Artificial Potential Field Supported by Augmented Reality. *Energies*14(20). https://www.mdpi.com/1996-1073/14/20/6642.

[32] Dubey V, Patel B, Barde S. (2023) Path Optimization and Obstacle Avoidance using Gradient Method with Potential Fields for Mobile Robot. In: 2023 *International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. p. 1358–1364.

[33] Xu Y, Jin Q, Zhang Y. (2023). path planning method for mobile robots incorporating artificial potential field method and ant colony algorithm. In: Ladaci S, Kaswan S, editors. *International Conference on Automation Control, Algorithm, and Intelligent Bionics (ACAIB 2023),* vol. 12759 International Society for Optics and Photonics, SPIE; 2023. https://doi.org/10.1117/12.2686536.

[34] Tang X, Pei H, Zhang D. (2024) Path Planning for a Wheel Foot Hybrid Parallel Leg Walking Robot. *Sensors* 24(7), 2178; https://doi.org/10.3390/s24072178

[35]Orozco-Rosas U, Montiel O, Sepúlveda R. (2019) Mobile robot path planning using membrane evolutionary artificial potential field. *Applied Soft Computing* 77:236–251. https://www.sciencedirect.com/science/article/pii/S1568494619300420.

[36]Siegwart R, Nourbakhsh IR. Introduction to Autonomous Mobile Robots. Cambridge, Massachusetts; London, England: The MIT Press; 2004.

[37] L L, A E. A comparative study between visibility-based roadmap path planning algorithms. In: 2005 IEEE/RSJ *International Conference on Intelligent Robots and Systems;* 2005. p. 3263–3268.

[38] Atyabi A, Powers DMW. (2013). Review of classical and heuristic-based navigation and path planning approaches https: //api.semanticscholar.org/CorpusID:16832195.

[39] Choset H, Burdick J. (1995). Sensor based planning. I. The generalized Voronoi graph. In: Proceedings of 1995 *IEEE International Conference on Robotics and Automation,* vol. 2. p. 1649–1655.

[40] Bhattacharya P, Gavrilova ML. (2008) Roadmap-based path planning using the Voronoi diagram for a clearance-based shortest path. *IEEE Robotics and Automation Magazine* June;p. 58–66.

[41] Masehian E, Amin-Naseri MR. (2004) A Voronoi diagram- visibility graph-potential field compound algorithm for robot path planning. *Journal of Robotic Systems* 21:275–300.

[42] Wein R, Berg JPVD, Halperin D. (2007). The visibility-voronoi complex and its application. *Computational Geometry* 36:66–87.

[43] Dias A, Fernandes T, Almeida J, Martins A, Silva E. In: Silva MF, Virk GS, Tokhi MO, Malheiro B, Guedes P, (2017).editors. *3D path planning methods for unmanned aerial vehicles in search and rescue scenarios Singapore:* World Scientific; p. 213–220.

[44] Alpkiray N, Torun Y, Kaynar O. (2018) Probabilistic Roadmap and Artificial Bee Colony Algorithm Cooperation For Path Planning. In: 2018 *International Conference on Artificial Intelligence and Data Processing (IDAP)* Malatya, Turkey; 2018. p. 1–6.

[45] Chai Q, Wang Y, He Y, Xu C, Hong Z. (2022) Improved PRM Path Planning in Narrow Passages Based on PSO. In: 2022 *IEEE International Conference on Mechatronics and Automation (ICMA)* Guilin, Guangxi, China. p. 41–46.

[46] Tunggal PT, et al. (2016). Pursuit algorithm for robot trash can based on fuzzy-cell decomposition. *International Journal of Electrical and Computer Engineering* 6(6):2863–2869.

[47] Keil JM, Sack JR. (1985). Minimum Decompositions of Polygonal Objects. *Machine Intelligence and Pattern Recognition* 2:197–216. https://doi.org/10.1016/B978-0-444-87806-9.50012-8

[48] Cai C, Ferrari S. (2009). Information-Driven Sensor Path Planning by Approximate Cell Decomposition. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B 39(3):672–689.

[49] Conte G, Zulli R. (1995) Hierarchical path planning in a multi-robot environment with a simple navigation function. *IEEE Transactions on Systems, Man, and Cybernetics* 25(4):651–654.

[50] Dugarjav B, Lee SG, Kim D, Kim JH, Chong NY. (2013). Scan matching online cell decomposition for coverage path planning in an unknown environment. *International Journal of Precision Engineering and Manufacturing* 2013;14(9):1551–1558.

[51] Gonzalez R, Kloetzer M, Mahulea C. Comparative study of trajectories resulted from cell decomposition path planning approaches. In: *Proceedings of the 2017 21st International Conference on System Theory*, Control and Computing (ICSTCC) Sinaia, Romania; 2017. p. 49–54.

[52] Iswanto I, Wahyunggoro O, Cahyadi AI.(2016). Quadrotor path planning based on modified fuzzy cell decomposition algorithm. *Telkomnika* 14(2):655.

[53] Debnath SK, Omar R, Bagchi S, Sabudin EN, Shee Kandar MHA, Foysol K, et al. (2021). Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles-A Review. In: Md Zain Z, Ahmad H, Pebrianti D, Mustafa M, Abdullah NRH, Samad R, et al., editors. Proceedings of the *11th National Technical Seminar on Unmanned System Technology 2019 Singapore: Springer Nature Singapore*; p. 99–111.

[54] Orozco-Rosas U, Picos K, Pantrigo JJ, Montemayor AS, Cuesta-Infante A. (2022)Mobile Robot Path Planning Using a QAPF Learning Algorithm for Known and Unknown Environments. *IEEE Access* 10:84648–84663.

[55] Orozco-Rosas U, Picos K, Montiel O. (2019) Hybrid Path Planning Algorithm Based on Membrane Pseudo-Bacterial Potential Field for Autonomous Mobile Robots. *IEEE Access* 7:156787–156803.

[56] Martin P, Pobil APD. Application of artificial neural networks to the robot path planning problem. In: Proceedings of the *Ninth International Conference on Applications of Artificial Intelligence in Engineering* Pennsylvania, PA, USA; 1994. p. 73–80.

[57] Van Hecke K, de Croon G, van der Maaten L, Hennes D, Izzo D. (2016) Persistent self-supervised learning principle: from stereo to monocular vision for obstacle avoidance. https://arxiv.org/pdf/1603.08047

[58] Pothal J, Parhi D. (2015) Navigation of multiple robots in a highly clutter terrains using adaptive neuro-fuzzy inference system. *Robotics and Automation* 72:48–58. https://doi.org/10.1016/j.robot.2015.04.007

[59] Li Y, Chen X. (2005) Mobile robot navigation using particle swarm optimization and adaptive NN. In: *ICNC*. p. 628–631.

[60] Cannady J. (1998). Artificial neural networks for misuse detection. In: *National Information Systems Security Conference.* p. 443–456.

[61] Kwon B, Thangavelautham J. (2020). Autonomous coverage path planning using artificial neural tissue for aerospace applications. In: *Proceedings of the IEEE Aerospace Conference* Big Sky, MT, USA. p. 1–10.

[62] Samarakoon SMBP, Muthugala MAVJ, Le AV, Elara MR. (2020) hTetro-Infi: A reconfigurable floor cleaning robot with infinite morphologies. *IEEE Access* 8:69816–69828.

[63]Wu K, Wang H, Esfahani MA, Yuan S. (2022)Achieving Real-Time Path Planning in Unknown Environments Through Deep Neural

Networks. *IEEE Transactions on Intelligent Transportation Systems* 23(3):2093–2102.

[64] Wang J, Liu J, Chen W, Chi W, Meng MQH. (2022). Robot Path Planning via Neural-Network-Driven Prediction. *IEEE Transactions on Artificial Intelligence* 3(3):451–460.

[65] Li X, Chen K, Zhao X, Zhu Y, Yang Z. Research on Robot Inspection Path Planning in Substation Based on Artificial Neural Network Algorithm. I*n: 2023 International Conference on Power*, Electrical Engineering, Electronics and Control (PEEEC); 2023. p. 821–825.

[66]Zadeh LA. (1965). Fuzzy sets. *Information and Control* 8:338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

[67] Hong TS, Nakhaeinia D, Karasfi B. (2012). In: Application of fuzzy logic in mobile robot navigation; p. 21–36. DOI:10.5772/36358

[68] Abiyev RH, Ibrahim D, Erin B. (2010). Navigation of mobile robot in presence of obstacles. *Advance Engineering Software* 41;1179–1186.

[69] Nasti SM, Vámossy Z, Kumar N. (2019). Obstacle Avoidance during Robot Navigation in Dynamic Environment using Fuzzy Controller. *International Journal of Recent Technology and Engineering (IJRTE)* 8(2). DOI:10.35940/ijrte.A1428.078219

[70] Yan Y, Li Y. (2016). Mobile robot autonomous path planning based on fuzzy logic and filter smoothing in dynamic environment. In: *World Congress on Intelligent Control and Automation (WCICA) Guilin, China* p. 1479–1484.

[71] Yang X, Moallem M, Patel RV. (2005) A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B (Cybernetics) 35(6):1214–1224.

[72]Mohanty PK, Kundu S, Srivastava S, Dash RN. A New T-S Model Based Fuzzy Logic Approach For Mobile Robots Path Planning. In: *IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)* Bhubaneswar, India; 2020. p. 476–480.

[73] Oleiwi BK, Mahfuz A, Roth H. (2021) Application of Fuzzy Logic for Collision Avoidance of Mobile Robots in Dynamic-Indoor Environments. In: 2nd *International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) DHAKA*, Bangladesh. p. 131–136.

[74] Chang H, Jin T. (2013) Command fusion based fuzzy controller design for moving obstacle avoidance of mobile robot. In: *Future Information Communication Technology and Applications Lecture Notes in Electrical Engineering*. p. 905– 913.

[75] Sanyal S, Konar D, Bhattacharjee A, Chatterjee S. (2024) General Type-2 Fuzzy Reasoning for Path-Planning of a Mobile Robot in a Dynamic Environment under Sensory Uncertainty. In: *2024 IEEE 3rd International Conference on Control, Instrumentation, Energy and Communication (CIEC)*. p. 73–78.

[76] Xue W, Zhou B, Chen F, Taghavifar H, Mohammadzadeh A, Ghaderpour E. (2024). A Constrained Fuzzy Control for Robotic Systems. *IEEE Access* 12:7298–7309.

[77] Bremermann HJ, (1958) The evolution of intelligence. *The Nervous system as a model of its environment. Seattle*, Washington; 1958.

[78] Holland JH. Adaptation in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press; Undated.

[79] Tu J, Yang SX. (2003) Genetic algorithm based path planning for a mobile robot. In: *IEEE International Conference on Robotics and Automation Taipei,* Taiwan. p. 1221–1226.

[80]Shibata T, Fukuda T. Robot motion planning by genetic algorithm with fuzzy critic. In: *8th IEEE International Symposium on Intelligent Control;* 1993. p. 565–570.

[81]Shi P, Cui Y. (2010). Dynamic path planning for mobile robot based on genetic algorithm in unknown environment. In: *Proceedings of the Chinese Control and Decision Conference Xuzhou*, China; p. 4325–4329.

[82]Karim B, Zhu Q. (2013). Genetic fuzzy Logic control technique for a mobile robot tracking a moving target. *International Journal of Computer Science Issues* 10(1):607–613.

[83] Qiu L. In: Li K, Li W, Chen Z, Liu Y, editors. Research on a hierarchical cooperative algorithm based on genetic algorithm and particle swarm optimization, vol. 874 of *Communications in Computer and Information Science Singapore*: Springer; 2018. p. 16–25.

[84] Hu XL, Lin Z. In: He J, Yu PS, Shi Y, Li X, Xie Z, Huang G, et al., editors. Coverage path planning of Penaeus vannamei feeding based on global and multiple local areas, vol. 1179 of *Communications in Computer and Information Science* Singapore: Springer; 2020. p. 687–697.

[85] Zhang R, Shi l, Xia W, Ma L. (2023) Improved Genetic Algorithms for Mobile Robot Path Planning. In: 2023 *International Conference on Artificial Intelligence Innovation (ICAII);* p. 36–43.

[86] Rashid R, Perumal N, Elamvazuthi I, Tageldeen MK, Khan MKAA, Parasuraman S. (2016). Mobile robot path planning using Ant Colony Optimization. In: *2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA*). p. 1–6.

[87] D M. (1997) Ant colony system: a Cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation* 1:53–66.

[88] Liu S, Mao L, Yu J. (2006). Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems. In: *IEEE International Conference of Mechatronics and Automation*.

[89] Liu J, Yang J, Liu H, Tian X, Gao M. (2017). An improved ant colony algorithm for robot path planning. *Soft Computing* 21:5829–5839.

[90] Purian F, Sadeghian E. (2013). Mobile robots path planning using ant colony optimization and fuzzy logic algorithm in unknown dynamic environment. In: *International Conference on Control, Automation, Robotics and Embedded Systems (CARE)* p. 1–6.

[91]You X, Liu K, Liu S. (2016). A chaotic ant colony system for path planning of mobile robot. *International*

*Journal of Hybrid Information Technology* 9(1);329–338. http://dx.doi.org/10.14257/ijhit.2016.9.1.28

[92]Zhang W, Gong X, Han G, Zhao Y. (2017) An improved ant colony algorithm for path planning in one scenic area with many spots. *IEEE Access* 5:13260–13269.

[93] Chibin Z, Xingsong W, Yong D. (2008). Complete coverage path planning based on ant colony algorithm. In: 15th *International Conference on Mechatronics and Machine Vision in Practice Auckland*, New Zealand. p. 357–361.

[94]Mohanty PK, Kumar S, Parhi DR. (2014). A new ecologically inspired algorithm for mobile robot navigation. In: Satapathy S, Biswal B, Udgata S, Mandal J, editors. *3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications*, vol. 327 of Advances in Intelligent Systems and Computing Cham, Switzerland: Springer; p. 755–762.

[95]Sun L. (2023). Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm. In: *2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 11; p. 985–989.

[96]Li D, Wang L, Cai J, Ma K, Tan T. (2023) Research on Terminal Distance Index-Based Multi-Step Ant Colony Optimization for Mobile Robot Path Planning. *IEEE Transactions on Automation Science and Engineering* 20(4):2321–2337.

[97]Foo J, Knutzon J, Oliver J, Winer E. (2006). Three-Dimensional Path Planning of Unmanned Aerial Vehicles Using Particle Swarm Optimization. In: *Proceedings*. DOI:10.2514/6.2006-6995

[98]Atyabi A, (2010) Phon-Amnuaisuk S, Ho CK. Applying area extension PSO in robotic swarm. *Journal of Intelligent Robot Systems* 58:253–285.

[99]Min HQ, Zhu JH, Zheng XJ. (2005). Obstacle avoidance with multi-objective optimization by PSO in dynamic environment. In: 2005 *International Conference on Machine Learning and Cybernetics*, 5;2950–2956.

[100]Hettiarachchi SD, Distributed evolution for swarm robotics; 2007.

[101]Pugh J, Martinoli A, Zhang Y. (2005). Particle swarm optimization for unsupervised robotic learning. In: *IEEE Swarm Intelligence Symposium*, 95–102.

[102] Hu C, Wu X, Liang Q, Wang Y. (2007) Autonomous Robot Path Planning Based on Swarm Intelligence and Stream Functions. In: *Evolvable Systems: From Biology to Hardware Springer Berlin Heidelberg; p. 277–284.*

[103] Kumar P, Pandey K, Sahu C, Chhotray A, Parhi D. (2017). A hybridized RA-APSO approach for humanoid navigation. In: *Engineering (NUiCONE), Nirma University, IEEE International Conference*; p. 1–6.

[104] Rendon M, Martins F. (2017) Path following control tuning for an autonomous unmanned quadrotor using particle swarm optimization. In: IFAC-Papers on Line, 50;325–330.

[105]Li CL, Wang N, Wang JF, Xu Sy. (2023). A Path Planing Algorithm for Mobile Robot Based on Particle Swarm. In: 2023 *2nd International Symposium on Control Engineering and Robotics (ISCER)* p. 319–322.

[106]Algabri M, Hassan M, Hedjar R, Alsulaiman M. (2015). Comparative study of soft computing technique for mobile robot navigation in an environment. *Computers in Human Behavior* 50:42–56.

[107]Shang Z, Bradley J, Shen Z. (2020). A co-optimal coverage path planning method for aerial scanning of complex structures. *Expert Systems with Applications* 158;1–16. https://doi.org/10.1016/j.eswa.2020.113535

[108] Passino K. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems* 22(3):52–67. doi: 10.1109/MCS.2002.1004010

[109] dan Liang X, yu Li L, gang Wu J, ning Chen H. (2013) Mobile robot path planning based on adaptive bacterial foraging algorithm. *Journal of Central South University* 20:3391–3400.

[110]Coelho L, Sierakowski C. (2005) Bacteria colony approaches with variable velocity applied to path optimization of mobile robots. In: *ABCM Symposium Series in Mechatronics*, vol. 2; p. 297–304.

[111]Abbas N, Ali F. (2017) Path planning of an autonomous mobile robot using enhanced bacterial foraging optimization algorithm. *Al-Khwarizmi Engineering Journal* 12(4):26–35.

[112] Jati A, Singh G, Rakshit P, Konar A, Kim E, Nagar A. (2012). A hybridization of improved harmony search and bacterial foraging for multi-robot motion planning. *In: Evolutionary computation (CEC), IEEE Congress*. p. 1–8.

[113] Hossain MA, Ferdous I. (2014). Autonomous robot path planning in dynamic environment using a new optimization technique inspired by Bacterial Foraging technique. In: *International Conference on Electrical Information and Communication Technology (EICT);* p. 1–6.