



Robot Path Planning and Motion Control: A Systematic Review

Alvin Prasad,^{1,*} Kaylash Chaudhary,² Bibhya Sharma³ and ABM Shawkat Ali⁴

Abstract

Robots contribute significantly to different sectors of the economy, helping humans perform several tasks in different sectors. To do so, the robots must move from the initial position efficiently and smartly to their destination smoothly. The efficiency and intelligence with which robots move are critical for their effective functioning. Hence, new algorithms are required to help the robot move and perform the allocated task. A diverse range of algorithms have been developed for various activities, but no single solution is universally optimal. This study reviews the different algorithms used in the movement of the robots to perform different functionalities. The classical, heuristic, and machine learning methods are analysed and summarised, highlighting each approach's advantages and disadvantages. By scrutinizing these algorithms, the study aims to provide a better understanding of the different attributes that reflect the performance of each of the algorithms. Moreover, suggestions for future research are presented, serving as a guide for researchers and scientists who are fascinated by robots and their movement. The study not only provides an understanding of the current algorithms but also provides direction for future advancements in the field.

Keywords: Navigation; Algorithm; Obstacle avoidance; Dynamic environment; Hybrid; Machine learning.

Received: 01 February 2024; Revised: 11 March 2024; Accepted: 13 May 2024.

Article type: Review article.

1. Introduction

Human actions are replicated with the use of robots, which have differing impacts on humans, their livelihoods, endeavors, and the environment.^[1] The trend of using robots started with the first programmable robot designed to be used in the manufacturing sector to move hot metal.^[2] In the current era, the potential of robots is evident in various sectors of the economy, such as health, construction, farming, defense, service, transport, education, and manufacturing.^[3-6]

To perform the specified task in the different sectors, the robot is supposed to move from an initial position to its destination. The path between the two points might not be straight and the robot will have to avoid obstacles to reach the

destination. This is the path planning problem where a set of different coordinates through which the robot will navigate without collision will be planned. A couple of factors are important while planning motion, such as the environment and the obstacles in the environment. The environment can be known, partially known, and unknown. A known and partially known environment will show the location of the obstacles. However, it will be difficult in an unknown environment. Obstacles of different sizes and shapes affect path planning. Moving and static obstacles are handled differently. All these factors are important when determining the fastest and safest path. After the path is planned, considering obstacle avoidance, the robot control system executes low-level motion control to perform movement.

Robots are of different types, like wheeled robots, mechanical arms, aerial vehicles, autonomous ground vehicles, tractor-trailers, walking robots, humanoids, and others. All of these robots require movement in different environments alone or in groups.^[7-9] Development and improvements in the area of sensors, communication technology, networking, and artificial intelligence evolved robots to multi-task and be more independent.^[10]

To plan the path and control a robot's motion, the robot's characteristics and the environment in which the robot will be navigating should be captured and analyzed. The environment

¹ School of Science and Technology, The University of Fiji, Private Mail Bag, Lautoka, Samabula, PO Box 4245, Fiji Islands

² School of Information Technology, Engineering, Mathematics and Physics, The University of the South Pacific, Private Mail Bag, Suva, PO Box 3537, Fiji Islands

³ School of Information Technology, Engineering, Mathematics and Physics, The University of the South Pacific, Private Mail Bag, Suva, PO Box 3537, Fiji Islands

⁴ Department of Computer Science and Engineering, Bangladesh University of Business and Technology, Dhaka, 1000, Bangladesh

*Email: alvinp@unifiji.ac.fj (A. Prasad)

can be classified into a structured environment and an unstructured environment. In a structured environment, the characteristics of the obstacles are known. However, in an unstructured environment, the obstacles are dynamic, and the characteristics of the obstacles are unknown.^[11] In an unstructured environment, the information about the environment is captured by multiple devices like inbuilt sensors, sensor networks, cameras, or a combination of both sensors and cameras.^[12] After the robot captures all the necessary information, it starts to plan the path to reach the destination with the help of path planning algorithms. The path that is planned during path planning will be executed with the help of motion control algorithms. The above is a summary based on articles.^[13-15] and the complete path planning process is summarised in Fig. 1.

Although multiple algorithms are available in the literature, most have their strength and weaknesses depending on applicability and situation. Most of these algorithms focus on path planning and not motion control. In this study, the aim is to analyze the different algorithms that are commonly used to plan the motion of a robot and control its movement. The merits and demerits of each algorithm are identified. Moreover, the limitation of the different research is discussed to provide a view of where the field of path planning is heading. The research also proposes ideas for future research in this area.

After the introduction of the paper, section 2 outlines the literature selection criteria. Followed by an overview of classical, heuristic, and machine learning path planning algorithms in sections 3, 4, and 5, respectively. Section 6 provides some insights into motion control algorithms. Section 7 highlights some of the hybrid path planning algorithms, followed by a discussion and conclusion in sections 8 and 9. Finally, section 10 provides the bibliography of the different articles cited.

2. Literature review search procedure and criteria

- A. The search was based on the following terms:
- robot motion planning and control algorithms
 - robot motion planning algorithms
 - robot motion control algorithms
 - robot path planning algorithms
 - robot path control algorithms
- B. The review is mostly based on literature from IEEE, SCOPUS, ACM, and Springer databases.
- C. The literature is mostly within the five years range from 2018 - 2023. However, random literature was selected from other sources only for information on the history and description of some of the required concepts.
- D. While selecting papers from the different databases focus was mostly on journal articles, magazines, and books. Conference articles are cited for the origin of the articles.
- E. From approximately 557 papers, a selection was carried out for the most relevant articles directly related to the research area.

F. Repeated articles and articles in different languages were excluded.

3. Motion planning

Motion planning and motion control are performed at two levels, namely local motion planning and global motion planning, respectively. Local motion planning is suitable in an unstructured environment where the obstacles are dynamic, and the information about the environment is captured using the data collection mechanism stated earlier.^[16] Whereas in global motion planning, the environment is structured, and the obstacles are static. The path from the initial position to the destination is planned before execution.^[17]

In the process of motion planning, several approaches and algorithms are developed, which can be categorized into, traditional classical algorithms, heuristic algorithms, and machine learning algorithms.^[17,18] However, we will consider later in the study that there is a significant shift to hybrids with different combinations.

4. Classical algorithms

Numerous classical methods are presented in the literature for different types of robots. In 1959, E W Dijkstra proposed one of the earliest motion planning algorithms.^[19] In the classical approach, the environment in which the robot navigates should be known by the algorithms before motion planning and control. The concept continued with another early algorithm for avoiding known polyhedral obstacles by a polyhedral object.^[20] Since then, researchers have shown more interest in motion planning and control. Hence, multiple other algorithms have been studied and introduced to the literature. Dijkstra's, A*, D*, cell decomposition, potential field, rapidly-explore random tree, and probabilistic road map are some of the classical-based algorithms discussed in detail in this study.

4.1 Dijkstra's algorithm

Dijkstra's algorithm is a graph-based algorithm that aims to find the shortest path between any two nodes. The algorithm initially calculates the distance from the initial node to all the other nodes. For this purpose, information about visited and unvisited nodes needs to be reordered.^[1,4] The algorithm initially searches for the neighboring nodes and keeps track of the visited and unvisited nodes. Based on the recording of visited and unvisited nodes, the algorithm takes the next step to an unvisited node, which has a shorter distance, and identifies the unvisited neighbors from the next node. The same concept is applied to finally reach the destination. The algorithm is very popular in solving routing problems.^[21,22] It has been expanded to solve the fuzzy shortest path problem in a dynamic environment.^[23] The extended Dijkstra's algorithm proposed to improve the traditional algorithm, making it more efficient while finding the path on a curved surface, which extends Dijkstra's to perform surface path planning.^[2,4] Dijkstra's algorithm is a cost-effective approach that can easily be implemented.^[25,26] However, it takes a longer time for the

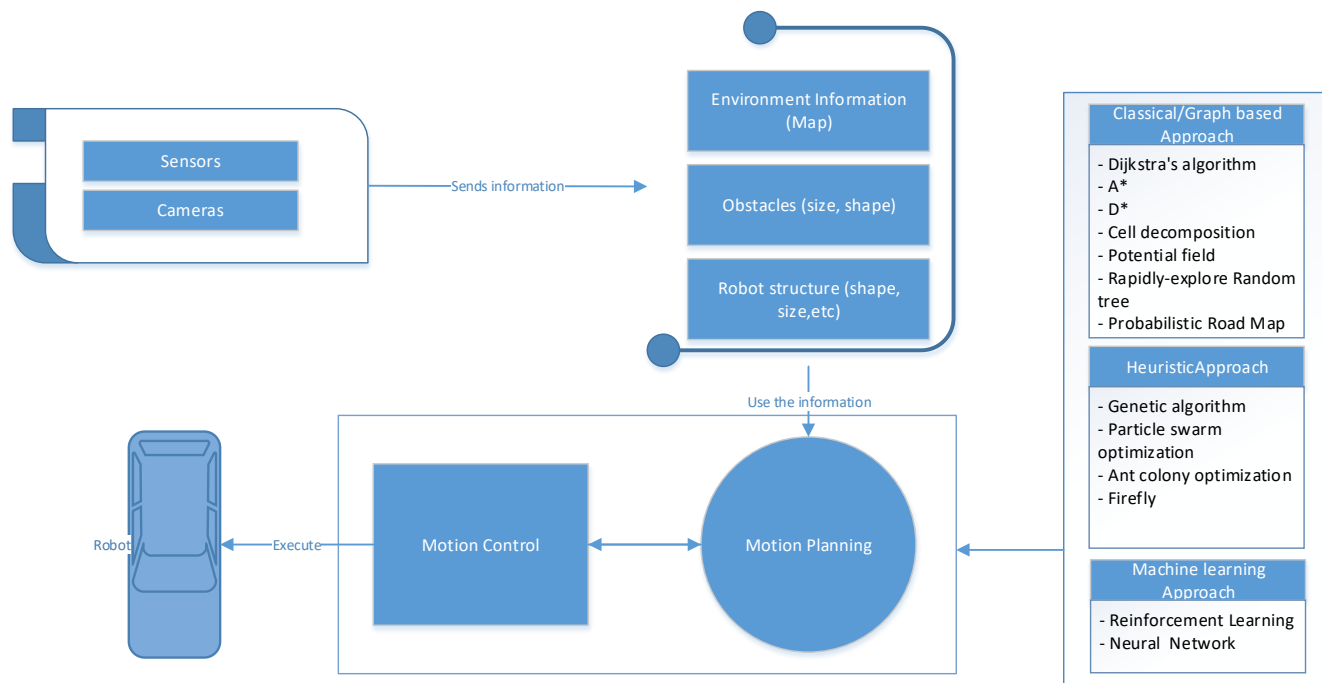


Fig. 1: The summary of the Path planning and control process to execute the robot.

whole process of planning the path.^[27]

4.2 A* algorithm

A* was proposed by Peter E Hart *et al.* in 1968,^[28] inspired by the Dijkstra approach to path search, where it uses the heuristic function, which allows the algorithm to ignore some of the nodes before reaching the destination.^[27] A* is more well-known to be used in static environments, but it is also used in dynamic environments. Most of the features in A* are like Dijkstra. When it comes to large maps, A* is more applicable.^[29] It is a commonly used graph search-based planning algorithm that stores information about the points that have been visited and points that are yet to be explored.^[30] This stored information helps the algorithm save time as the nodes visited are already known, and the robot travels on the unexplored path and is cost-effective. After the cost is evaluated between the different nodes the path which is cost effective and is directly connected to the destination is considered. Both motion planning and motion control modules are proposed in Ref. [31], and the new proposed model combines the advantages of the A* algorithm and the fuzzy analytic hierarchy process to plan the path. The traditional A* algorithm has many different variants used in different scenarios. A new Successive Linearization in a Feasible set algorithm is proposed in Ref. [32] for motion planning of low-speed and unstructured environment for vehicle parking where the initial guess of the shape of the vehicles in the environment is analyzed with an A*-Reshaping algorithm.^[33] The proposed algorithm is claimed to solve the non-convex collision constraints and deals with nonlinear robotic dynamic constraints. The other popular variant is Hybrid A*,^[34] which is more popular in uncertain environments and produces the shortest path with more territory coverage. A combination of

the classical A* and Hybrid A* is applied to vehicle parking search and U-turning on a blocked road is known as Hybrid-state A*.^[35] Guided Hybrid A* is introduced to enhance the hybrid A* algorithms using the visibility diagram for searching the shortest path and thereafter allowing the hybrid A* to derive the optimum path.^[36] Lifelong Planning A* algorithm uses the heuristics to create the search tree, similar to what is generated by classical A*.^[37] However, the other successive searches follow the initial tree much faster than the original. The reason behind the speed is that it compares the previous search paths and reuses the identical paths from that, making it faster.^[38] Hierarchical A* solves two problems associated with A* where the first one is to solve the Valtorta barrier, and the second it introduces a caching technique that prevents A* from repeating the same node search in each repeated search.^[39] Improved Hierarchical A* considers time, cost, distance, as well as hardware and software cost, presenting a cost-effective approach for the parking guidance system.^[40] A* algorithm is cost-effective, fast, easy to implement, and finds all the potential paths. However, the performance depends on the heuristic algorithm.^[32,41]

4.3 D* algorithm

D* algorithm, which is similar to A*, was proposed by Anthony Stentz with the aim of path planning without a model or floor plan in a cost-effective manner, which allows its cost parameters to change during problem-solving.^[42] It is a graph-based algorithm that searches the shortest path in a dynamic environment where the path is updated in real-time. D* is a dynamic version of Dijkstra’s algorithm without the heuristic function.^[43] D* is also referred to as Dynamic A* as the D* algorithm acts similarly to the A* algorithm with the

difference that A^* keeps only an open list of nodes visited, whereas D^* deals with two separate functions, one that calculates the path cost to generate the optimal path to the goal and another function that updates the cost function that helps in updating the open list,^[44] hence, is dynamic in nature. The author of the original D^* algorithm, Anthony Stentz, proposed a variant of D^* , namely Focused D^* ,^[44] where the algorithm initially starts the path plan from the goal to the start node and thereafter travels towards the goal, updating the initial path and associated cost. It is stated that Focused D^* is better than the D^* in terms of computation time. D^* lite is another variant of the D^* algorithm that plans the path similar to D^* .^[45] However, it uses Lifelong planning A^* , which makes it faster than D^* . The extension of D^* and D^* lite gave us another variant known as field D^* ,^[46] which is used by mobile robots in-house and on difficult terrain. In the same article, the authors present another algorithm, Multiresolution Field D^* , an extension of Field D^* . Field D^* and Multiresolution Field D^* solve two problems of grid-based path planning. One, it improves the quality of the path produced using grids as it uses linear interpolation that allows the connection of nodes from anywhere on the grid, not just the center or corner of the grid. Second, the grid-based plan is costly and requires computational power as it requires uniform resolution. Multiresolution Field D^* solves the problem by planning in nonuniform grids, allowing low-resolution presentation, thus reducing the use of computational resources and reducing cost.^[46] The application of the D^* algorithm and its variants are mostly on mobile robots used indoors and outdoors in challenging environments.^[47,48] The algorithm is better than A^* in large and complex environments. It can be used in dynamic environments to replan paths. However, the drawback of D^* is that as the problem becomes complex, it takes more time to process, and hence, it becomes slow.

4.4 Cell decomposition

Cell decomposition was introduced by Howie Choset in Ref. [49], inspired by Ox, which plows a field back and forth. In the cell decomposition method, the environment will be decomposed or divided into cells which may contain some obstacles. The division will be created with the help of cells to show the free space and the occupied space. Based on the division, multiple paths can be created from the initial position to the target position by connecting the cells, which are free of obstacles.^[14] The method can be classified into a regular grid, adaptive cell decomposition, and exact cell decomposition.^[50] In robotics, traditional grid-based approaches like regular grids (where cell size is fixed) are simpler to implement.^[51] However, adaptive cell decomposition offers a more flexible solution.^[52] It divides the environment into cells of varying sizes, starting with larger cells in open areas and refining the grid with smaller cells in more complex or obstacle-rich regions. The cell is examined, and if obstacles are identified, then the cell is further decomposed using the quad-tree approach. This approach divides the larger rectangle into four

equal proportions to identify the free space. Exact cell decomposition divides the cells into regular cells without specific shapes and sizes with vertical lines and numbers for easy identification, which leads to the cell joining the other cells in the free space.^[18] The shapes generated to identify free spaces can be rectangles, triangles, and trapezoids.^[53] The Exact Cell decomposition dividing the environment with vertical lines gets the name vertical cell decomposition.^[54] In the latest article,^[55] a Radial Cell Decomposition algorithm is proposed where the cell is divided with a set of arches to identify free space to create a path to the destination, claiming to be faster than Vertical Cell decomposition and generates a shorter path compared to the A^* algorithm. Cell decomposition algorithms that use smaller predefined cells increase the number of cells in a larger environment, producing a shorter path but making it slower. The division of cells non-proportionally makes the cell decomposition algorithm faster and decreases the computation time.^[55] The problem with cell decomposition is that it becomes more complex and generates infeasible solutions.^[56]

4.5 Artificial potential field

An artificial potential field (APF) proposed by Oussama Khatib is a classical algorithm that provides real-time information applied to a mechanical arm and aerial robots in early studies.^[57] APF can perform motion planning at both local and global levels. However, the initial focus was only on static obstacles. This approach is based on gravitational and repulsive forces where the goal applies the gravitational force, and the obstacles repel,^[58] as shown in Fig. 2. Hence, the robot is attracted by the target position. However, while on the move, it will avoid the obstacles as the push and the pull concept will act.

It is an effective approach for obstacle avoidance, works well with redundant and non-redundant robots, and is easy and flexible to implement in real-time, but its major drawback is trapping in the local minima and the inability to access the goal if the goal is near to an obstacle.^[60-62] Different artificial potential field algorithms are modified and proposed to tackle the drawbacks in a static environment. A combination of improved artificial potential field algorithm and Q-Learning is proposed in Ref. [63] to solve the local minima problem for routing in an unknown dynamic environment for autonomous vehicles. An image-based path planning method, namely modified artificial potential field constrained-generalized traveling salesman problem (MAPF-CGTSP), which combines the modified potential field and constrained generalized traveling salesman problem to generate a safe path. Furthermore, the trap in the local minima results in oscillations, leading to the robot consuming more energy than necessary.^[61] The problem of high energy consumption is being dealt with by using a predictive artificial potential field algorithm.^[59]

The classic APF has been modified with multiple functions applied to autonomous driving to find a collision-free path.^[64] In Ref. [65], the authors propose an improved APF for motion

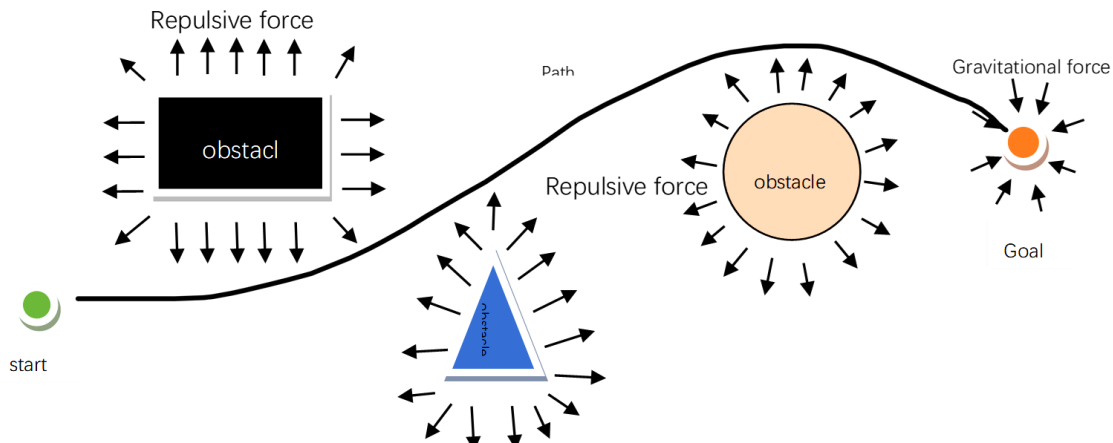


Fig. 2: APF forces.^[59]

planning, which avoids multiple obstacles after identifying the posture and position of the obstacles. The dynamic potential field is used to plan the motion of lane change in autonomous vehicles, and APF combined with constrained Delaunay triangulation is used for motion control.^[66] In Ref. [67], a framework is presented with off-line path planning and on-line motion control algorithms. Path planning is carried out with the updated potential field method, and the attributes of the velocity control method are enhanced for the online motion control. The improved artificial potential field is applied to grounded vehicles, ships, mechanical arms, and unmanned aerial vehicles.^[68-70]

4.6 Rapidly explore random tree

Sampling based algorithm like rapidly exploring random tree (RRT), founded by Steven LaValle and James Kuffner,^[71] creates a tree-like graph by connecting different randomly generated points to find its path in free space. When a random point is created, it relates to the closest node. The generation of the tree ends when a target is met, or the space is filled, as shown in Fig. 3.

This leads to several nodes connected with each other which generates unnecessary connections. RRT is known to generate random paths which can basically be unpredictable

from neural network-based observers.^[72] It generates a tree from the initial point in search of the destination. Sertac Karaman and Emilio Frazzoli proposed RRT*, which was the optimized version of RRT, and finds the shortest path using the same graph technique but fulfills the RRT gap.^[73] It follows the same process as RRT but records the movement and information about its neighboring nodes as well. Together with that, the shortest node selected is checked again with its other previous connection to see if any other path visited previously is shorter than the chosen path.

Hu *et al.*^[74] presented another variant of RRT, which claims to reduce the size of the tree and modifies the steer function, namely Optimal RRT, to control the planned path. A new sampling-based path planning algorithm, Adversarial RRT*, is proposed to deceive intruders from predicting the path of a robot.^[72] An extension to RRT*, namely a reliable and robust rapidly exploring random tree (R2-RRT*) algorithm,^[75] is applied to off-road autonomous ground vehicles, which claims to be cost-effective and efficient. RRT-connect is another variant of the RRT algorithm that creates two trees, one rooted at the start point and another rooted at the goal point, approaching each other for a connection.^[76] RRT and the other algorithms derived from RRT are very efficient in finding the possible paths; however, finding the return path is difficult to

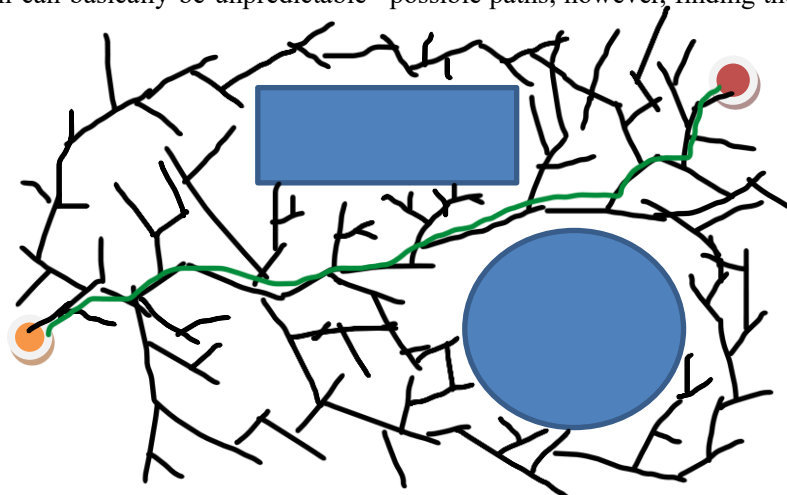


Fig. 3: Rapidly explore the random tree.

remember and is not stable in a dynamic environment.^[77,78]

4.7 Probabilistic road map

Kavraki *et al.*^[79] discussed a probabilistic roadmap (PRM) to solve the problem of identifying the path between the start and the goal node using graphs and random points. The PRM acts like RRT, where a graph is constructed using random points, and then the path is searched for. A point created randomly connects to the nearest neighboring point, which is obstacle-free to create a path. It is a two-phase process where the algorithm first learns and then queries to get the best path.^[80]

Identifying random points leads to challenges faced in narrow passages where identifying the free space and generating an optimal path might require more computation time. To address the motion planning problem for closed-chain constraint, an efficient motion planner is proposed by randomly identifying the feasible nodes while computing the inverse kinematics (IK) solution.^[81] Researchers have suggested improved PRM algorithms.^[82-84] However, most of them still find challenges in planning in real-time and the increased computation time in narrow paths. [Table 1](#) summarises the different variants of the classical algorithm.

Table 1: Variants of classical algorithms.

Problem/Objectives	Algorithm	Reference
To determine the shortest path between two nodes	Dijkstra algorithm	[19]
To extend the simple visibility graph algorithm to more general collision avoidance problem	Visibility graph algorithm	[20]
To choose the best route method using the algorithm	Dijkstra algorithm	[21]
To create a collision free routing for automated guided vehicle in a warehouse	improved Dijkstra algorithm	[22]
To solve the shortest path problem in dynamic environment where edges length is represented in fuzzy numbers	Fuzzy Dijkstra algorithm	[23]
To create an effective shortest path search algorithm inspired by Dijkstra's algorithm	A*	[28]
Solves the non-convex collision constraints and deals with nonlinear robotic dynamic constraints	A*-Reshaping algorithm	[33]
To find the shortest path spreading more coverage on the different nodes.	Hybrid A*	[34]
To perform U-turn on blocked roads, search for parking, and solve collision-avoidance safety and the searching efficiency of the Hybrid A*	Hybrid-state A*	[35]
To create an optimum path using hybrid A* and visibility diagram	Guided Hybrid A*	[36]
This approach reuses the initial search tree, resulting in the path search being faster compared to A*	Lifelong Planning A*	[37]
To break the Valtorta barrier in A* and use a caching method to avoid A* to search the same nodes in all its successive searches.	Hierarchical A*	[37]
To create a cost-effective approach for parking considering hardware and software costs together with the shortest time, shortest distance, and cost-effective search.	Improved Hierarchical A*	[38]
To be used in a dynamic environment without a plan	D*	[42]
To perform better than D* where it starts the plan from the goal node.	Focused D*	[44]
It is based on Lifelong planning A* but behaves like D* to be used on grid graphs and unknown environments.	D* Lite	[43]
To solve two limitations of grid-based path planning which provides quality path plan and reduces the use of computational resources.	Field D* Multiresolution Field D*	[46]
To solve the find path problem of robotics, that is applied by generating the grid on the map to identify free and occupied space. The division of the cell is done with same size grid.	Regular grid (Cell decomposition)	[51]
It is based on a quad-tree approach to identify the free space. It is better than a Regular grid. The cell is created non-proportionally	Adaptive cell decomposition	[52]
Exact cell decomposition divides the cells into regular cells without specific shape and size	Exact cell decomposition (vertical and radial decomposition)	[54], [55]
It is based on gravitational and repulsive forces where the goal	Artificial Potential Field (APF)	[57]

Problem/Objectives	Algorithm	Reference
applies the gravitational force, and the obstacles repel		
It combines the modified potential field and constrained generalized traveling salesman problem to generate a safe path.	MAPF-CGTSP	[61]
To reduce high energy consumption when caught in the local minima	Predictive artificial potential field algorithm	[59]
To avoid multiple obstacles after identifying the posture and position of the obstacles	Improved APF	[65]
To provide a safe, stable, flyable path in dynamic airspace	Dynamic APF	[85]
To find the shortest and most cost-effective path by creating a tree rooted at the start position of the robot.	RRT	[71]
To fill the gap in RRT, RRT* records information of all the nodes and rechecks the shortest path to the goal.	RRT*	[73]
To deceive Neural Network observers.	Adversarial RRT*	[72]
To derive a reliable, safe, and cost-effective path for off-road ground vehicles.	R2-RRT* algorithm	[75]
To create two trees, one rooted at the start and another rooted at the goal.	RRT-connect	[72]
To improve the success rate of RRT as RRT planners project configuration onto a point.	Bidirectional RRT	[86]
To tackle the narrow corridor problem with an atlas and bidirectional RRT.	Atlas RRT	[87]
To plan the path by generating random samples on a curved manifold.	Tangent bundle RRT	[88]
To solve the problem of identifying the path between the start and the goal node using graphs and random points.	Probabilistic Road Map	[79]
To plan motion in real-time for complex environments where it can encode a large number of vertices and edges.	Hierarchical Dynamic Roadmap	[83]
To generate a universal graph, removing unnecessary edges which can be modified in real time with changes in the environment.	Dynamic Roadmap	[84]

5. Heuristic approach

In the classical approach, the algorithm needs to learn about the environment prior to motion planning; however, in the heuristic approach, the algorithm gets into the environment to analyze, discover, and learn by itself based on experience. In this section of the study, we will discuss some of the important heuristic algorithms like genetic algorithm (GA), fuzzy logic (FL), and nature-inspired algorithms like particle swarm optimization (PSO), ant colony optimization (ACO), and firefly algorithm (FA).

5.1 Genetic algorithm

A genetic algorithm (GA) introduced by John Holland is an optimization algorithm that can evolve itself as it approaches a new environment.^[89] It is built based on natural genetics that uses natural selection, crossover, and mutation procedures, as shown in Fig. 4.^[90] GA chooses the fittest path from multiple options; it analyses multiple options in the free space and generates more than one solution that helps the robot jump out of a path if it gets stuck to another. It is a promising technique that searches for the optimal path by creating multiple points with a high possibility of getting the best path. Matrix-binary codes genetic algorithm (MGA) is proposed, which uses the matrix-trace method to sequence the path during search and

combines with GA to search the obstacle-free path to the destination using an array.^[91] GA has its own weaknesses, like falling into local optimization prematurely, newly generated points having worse fitness, and slowing down the convergence rate. To solve this problem, an adaptive elite genetic algorithm with singularity avoidance is presented.^[92] In Ref. [93], a global path planning algorithm based on GA (hTetro-GA) is proposed to be used on a hinged-Tetromino reconfigurable robot that focuses on multi-objective optimization. An improved GA is proposed, where it is combined with fuzzy logic to solve the vehicle routing problem.^[94] The application is applied to a fleet of vehicles in a warehouse. A domain knowledge-based genetic algorithm for single and multiple targets is proposed to find the optimal collision-free path from the start position in a static environment.^[95] In order to minimize the hovering and flying energies used by multiple aerial vehicles, a genetic trajectory planning algorithm with variable population size (GTPA-VP) is proposed.^[96] Another multi-objective genetic (MRPS-MOGA) path planning algorithm is introduced with multiple objective functions to manage objectives like safety, distance, smoothness, traveling time, and collision-free path to find the best path. MRPS-MOGA utilizes tournament selection, ring crossover, and adaptive bit string mutation in the process. The

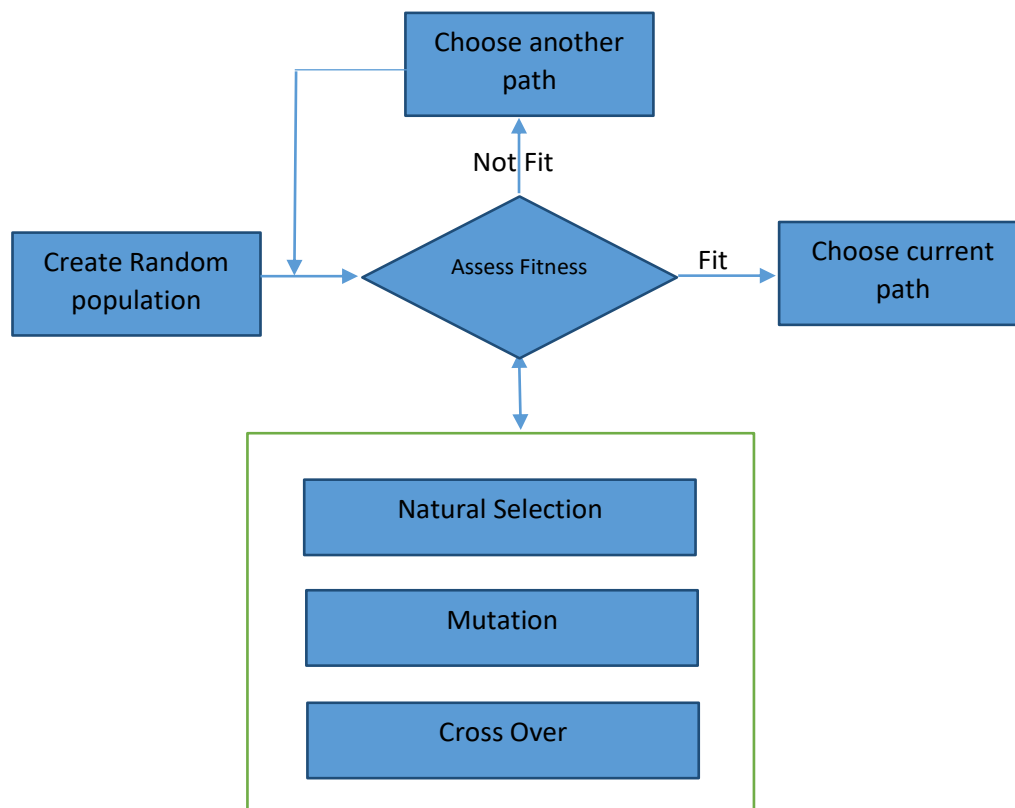


Fig. 4: Genetic algorithm structure.

hybridization of GA is also presented in this study later.^[97]

5.2 Fuzzy logic algorithms

Fuzzy logic algorithms (FLA) are based on making decisions based on reasoning, like humans that can handle obstacles in unknown environments. Zadeh introduced the fuzzy logic concept in 1965.^[98] The performance of FLA depends on the fuzzy rules and adequate parameters, but designing the fuzzy rules is not easy.^[99] The basic structure of FLA consists of fuzzification, rule-base, inference mechanism, and defuzzification, as shown in Fig. 5.^[100] Fuzzification converts input into information that will easily be converted into fuzzy rules and compared against the stored rules from the rule base. The rule-base phase keeps the various rules stored, which is used to screen the inputs based on the stored rules and control the system. This phase uses the “if then” rules. Further, in the inference mechanism phase, the rules are converted, and operator knowledge is applied to understand the relationship between the input and output of the system. In the

defuzzification phase, the output from the inference phase is converted to real control values.^[99-101] Different authors propose fuzzy logic approaches. An interpretable fuzzy logic-based method is proposed in Ref. [102] for multi-robot navigation in a dynamic environment that reduces the number of parameters in the fuzzy rules. A trajectory control scheme is proposed in Ref. [103] that is flexible with wheeled mobile robots in a practical environment where navigation is done in environments with varying curves. It also proposes a robust rule-based controller to improve the navigation performance. FLA are mostly combined with other algorithms for path planning to provide better results.^[104,105]

5.3 Particle swarm optimization

The particle swarm optimization (PSO) algorithm developed by Kennedy and Eberhard was inspired by the smart behavior of animals, particularly bird flocks, to conquer more effectively in uncertain environments.^[106] The swarm changes patterns while searching for food based on experience and

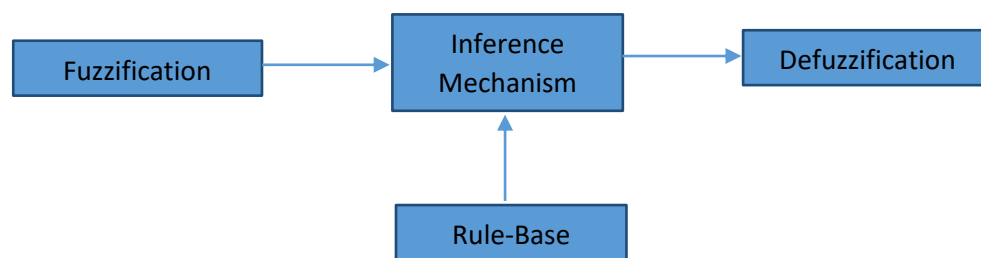


Fig. 5: Fuzzy logic process structure. Reproduced with permission from [99], Copyright 2019, Springer Nature Singapore Pte Ltd.

learning from the neighbor. Like swarm and fish movement, a set of particles are placed in the free space with some rules where the particles will identify and share their personal best path with each other to reach the goal. The PSO that is easy to implement has few parameters to manage and is flexible, but it also has some drawbacks, such as falling in the local optimum, low convergence speed, and poor optimization stability. To overcome the drawback, a novel Levy Flight, power function, and Singer map employed particle swarm optimization (LPSPSO) was proposed.^[107] A new adaptive particle swarm optimization (APSO) approach to path planning in a static environment was presented in Ref. [108]. APSO is based on the PSO algorithm, which creates a new objective function to prepare the path. The authors claim that some attributes of the new approach are better than the classical one. An improved PSO with adaptive weighted delay velocity (PSO-AWDV) to solve the smooth path optimization problem was proposed.^[109] In Ref. [110], a new planning approach was presented where the continuous high-degree Bezier curve is used to smooth the path, and an improved PSO with an adaptive fractional-order velocity is introduced to search the path efficiently.

5.4 Ant colony optimization

Ant colony optimization (ACO) was proposed in the nineties by Dorigo *et al.*, which was initially known as the ant system.^[111-113] An iterative approach inspired by the behavior of ants has been successfully applied to solve the path-planning problem. Ants search for food in the environment and carry it to the nest once food is identified. On the way back to the nest, the ant drops trail pheromones, which is a way of communicating with the other members.^[111] Many variants of ACO have been proposed and implemented for different environments and situations to overcome the drawback of slow convergence speed and the local optimal issue of the traditional ACO.^[114,115] In Ref. [116], the study proposed a new, improved ACO algorithm with fuzzy logic to tackle the local path planning problem in a dynamic environment. The limitation of the study is that the algorithm is not tested on curved obstacles. An Improved ACO was applied to an intelligent parking system for automated guided vehicles, and the valuation function was adapted with the reward/penalty mechanism for an efficient search.^[117] ACO is very effective and easy to structure, but it gets into confusion at the equilibrium point.^[118]

5.5 Firefly algorithm

The firefly algorithm (FA), proposed by Xin-She Yan, is a motion planning algorithm used in uncertain environments that minimizes computational calculations.^[119] It is a swarm intelligence-based algorithm inspired by fireflies and the variation in light brightness.^[120] The solution generated is based on the objective function where the firefly (nodes) signal (communicate) each other using the brightness of the light; brighter light attracts others towards itself, and low

brightness allows the firefly to travel randomly. The algorithm explores the environment, records information, and then exploits the best path using the explored data. The FA has been implemented in several research studies to solve the path planning problem.^[121-124] During the search process, the searched paths collide with obstacles, which will cause other fireflies to cross the same obstacle, leading to conflicts in the exploration ability and triggering computational complexity. To overcome the drawback, an FA with adaptive population size for path planning is proposed that adjusts the population size depending on the collision.^[125] Firefly algorithm 1 to 3 (FA1→3) is also proposed to overcome the drawbacks with different movement styles.^[126] FA's strength is the techniques it uses to find the motion path; however, getting the best and complete path is challenging, leading to the algorithm trapping into the local optima. Table 2 lists the variants of heuristic algorithms.

6. Machine learning approach

Intelligent systems with human-like reasoning and thinking ability are what make the most significant mark in the current era. Machine learning, an important part of artificial intelligence (AI), is a method where computers or machines learn from input data and predict future actions. The more advanced algorithms can learn the environment and act appropriately. The advancement of computer technology has led to a more intelligent robot that requires algorithms to be sufficient for intelligent developments. Machine learning techniques are categorized into supervised learning, unsupervised learning, reinforcement learning, and semi-supervised learning. Supervised learning is where the data for both input and output are available, and prediction is made based on experience from past data. Unsupervised learning is where some unlabelled data is available for input; however, the output is not known as compared to supervised learning. In reinforcement learning, the algorithm learns through experience by understanding the environment. Semi-supervised learning combines supervised and unsupervised learning, where the machine might have little or no data and provides better and cost-effective results.

Machine learning algorithms such as improved Q-Learning,^[128] deep reinforcement learning based robot path planning method,^[129] neural network and hierarchical reinforcement learning, and others are applied in the path planning and control problem. The most popular algorithm used in robot path planning is RL, which will be discussed further in the study.^[130]

Reinforcement learning (RL), introduced by Richard S. Sutton and Andrew G. Barto, is a method in the area of artificial intelligence that learns the environment in which it is and acts as part of the environment.^[131] RL while in the environment, learns about the good and the bad about the environment by trial and error before any action. A globally guided reinforcement learning (G2RL) is presented, which utilizes the global path planning algorithm to generate this

Table 2: Variants of heuristic algorithms.

Problem/Objectives	Algorithm	Ref.
To find path based on natural genetics to solve the path search problem.	Genetic algorithm	[89]
Uses the matrix-trace method to sequence the path during search and combines with GA using an array.	Matrix-binary codes based genetic algorithm (MGA)	[91]
To overcome the optimization prematurity and efficiency weakness in GA.	Adaptive elite genetic algorithm	[92]
An application based on soccer robot navigation in a dynamic environment with the intent to change the crossover probability and mutation probability in genetic operation.	S-adaptive genetic algorithm	[127]
To achieve hinged-tetromino reconfigurable robot.	hTetro-GA	[93]
To solve the vehicle routing problem.	Improved GA	[94]
To plan collision free path for both single and multiple.	Domain knowledge based genetic Algorithm	[95]
To minimize the hovering and flying energies used by multiple aerial vehicles.	Genetic trajectory planning algorithm with variable population size (GTPA-VP)	[96]
Is designed with multi objective functions to reduce cost and obtain a collision free path	Multi-objective genetic (MRPS-MOGA)	[97]
To track trajectories with rules encodes by instantaneous measurements and avoids complexity of encoding with flexibility to model the multi-input and output of mobile robot.	Z-number based fuzzy logic (Z-FL)	[103]
To find collision free path for multiple robots.	Interpretable fuzzy logic-based method	[102]
To efficiently manipulate a nonholonomic mobile robot with decoupled path-trajectory planning approach, flatness concept and fuzzy logic.	Fuzzy-flatness technique	[104]
To plan obstacle free path based on the movement of particles in groups and generating paths with the help of each other to reach the goal.	Particle swarm optimization algorithm	[106]
To overcome the drawbacks of PSO.	LPSPSO	[107]
To plan the path a new objective function is added to the classical PSO.	APSO	[108]
To solve the smooth path optimization problem is proposed.	PSO-AWDV	[109]
To tackle the local trapping and premature convergence issues.	Improved PSO with an adaptive fractional-order velocity	[110]
To solve the problem of finding the minimum cost path for robot path planning.	ACO	[111]
To solve the problems of local optimization, poor convergence, and low search efficiency by updating the pheromone with the introduction of attenuation factor.	Improved ant colony algorithm	[114]
To overcome the drawback of traditional ACO with the inclusion of angle guidance and obstacle exclusion factor.	Improvement adaptive ant colony optimization algorithm (IAACO)	[115]
To plan path in dynamic environment with ACO and fuzzy logic.	Improved ant colony optimization	[116]
To adopt the valuation function and reward/ penalty mechanism in an intelligent parking system.	Improved ACO algorithm	[117]
To provide balance between exploration and exploitation	Tidal force firefly algorithm	[121]
To improve the FA with quantum theory integration as an optimization tool.	Quantum-inspired firefly algorithm with cuckoo search	[122]
To address the problem of permutation optimization.	Discrete firefly algorithm	[124]
The firefly population size is adjusted to improve the optimization ability.	Firefly algorithm with adaptive population size for the path	[125]

plan.^[132] The local reinforcement learning planner uses the global path plan generated to evade static and dynamic obstacles to reach the goal node. In the latest article, a reinforcement learning heuristic A* (RLHA*) is an extension to an artificial neural network (ANN) based algorithm where ANN is used as the learning heuristic function.^[133] The role of this function is to estimate the cost and identify a proposed path. It uses the learning-heuristic function where the algorithm generates and uses its own path rather than the precomputed paths.

Q-learning is the most popular RL algorithm that does not require environmental data for motion planning.^[134] The algorithm is based on a Q-table that captures the robot environment interaction information of reward and state. The table gets updated with continuous environment interaction and produces an optimal path. However, in a complex environment, the Q-learning algorithm deals with issues like low learning efficiency and slow convergence speed. To overcome Q-learning's drawbacks, the entropy-based prioritized regret-QL (EPRQL) algorithm is proposed by adding an obstacle learning matrix to enhance the success rate, where a heuristic reward accelerates the learning process.^[135] The experience-memory Q-learning (EMQL) algorithm is proposed that combines the Q-learning algorithm with the experience memory mechanism to plan a path in an unknown dynamic environment.^[136]

For the robot to work in a more complex environment, Multi-agent reinforcement learning techniques and deep neural networks are applied.^[137] One of the main challenges of RL is the design of features that will be used to learn. In Ref. [138], a learning-based algorithm known as Lyapunov-based soft actor-critics with collision probability prediction (LSAC-CPP) is designed for autonomous vehicles, which predicts collision by combining the Lyapunov function with the soft actor-critic algorithm. However, testing of the algorithm in a real environment with dynamic obstacles is yet to be performed.

In Ref. [139], a deep reinforcement learning algorithm is proposed that uses LiDAR data to collect real-time environmental information to construct the map and an improved deep reinforcement learning algorithm to solve the order of movement based on the map.^[140] A deep reinforcement learning (DRL) based algorithm is presented for use in the agricultural sector. In this study,^[141] a recurrent neural network is applied to store the information on robots' past states, and then a deep deterministic policy gradient algorithm (DDPG) uses the information stored about the states to predict obstacle-free paths. DDPG is based on actor-critic architecture based on the experience replay mechanism and target network idea of deep reinforcement learning to solve action problems.^[142] The study further proposes a long short-term memory in the DDPG network, which guides the robot to move faster towards the goal. In Ref. [143], the robot arm Markov decision process (RAMDP), together with DDPG and hindsight experience replay (HER), is used to search for the optimal path.

Path planning using DDPG is of interest among researchers capable of solving path-planning problems. The agent in DDPG is based on the actor-critic method, where the policy is that the actor identifies possible actions based on the states, and the critic evaluates the actions proposed by the actor, which produces a continuous output. Firstly, the mobile robot determines the type of environment it must work in, incorporating its own state data; secondly, the actor outputs its action per its current network. By evaluating the multiple actions, the critic generates Q-values, which are constantly updated as the actions change, thus generating improved action strategies. The DDPG policy tests multiple samples, reinforcing the actions with higher q-values while abandoning the other actions to obtain the best strategy. Below an algorithm is presented inspired from Ref. [144].

To get the best strategy, the following actions are performed:

1. Set the actor, critic, and target network with the experience pool.
 - i. Actor network $\rightarrow \mu(S|\theta_\mu)$
 - ii. Critic network $\rightarrow Q(S, a | \theta_Q)$
 - iii. Target network $\rightarrow \theta_\mu^t \leftarrow \theta_\mu, \theta_Q^t \leftarrow \theta_Q$
 - iv. Experience pool P
2. The robot obtains data on its state in the environment s_t , get actors' action data a_t , get the reward r , and get the new state S_{t+1}
3. Store state, action, reward, and new state data in the experience pool, as shown in equation (2).

$$P = s_t, a_t, r, S_{t+1} \tag{2}$$

Once the data in the pool reaches a certain number, samples (N) from the pool will be trained.

4. Calculate the current network loss of critic as shown in equation (3).

$$L(\theta_Q) = \frac{1}{N} \sum_{i=1}^N (\gamma_i - Q(S_i, a_i | \theta_Q))^2 \tag{3}$$

where $\gamma_i = r_i + \gamma Q'(S_{i+1}, \mu'(S_{i+1} | \theta_\mu^t) | \theta_Q^t)$ is the target, γ is discount coefficient, and i is the sample.

5. Update the current network parameters of the actor following equation (4).

$$\nabla_{\theta_\mu} j = \frac{1}{N} \sum_i \nabla_{a_i} Q(S_i, a_i | \theta_Q) \nabla_{\theta_\mu} \mu(S_i | \theta_\mu) \tag{4}$$

where ∇j is gradient

6. After the robot actions the data, it is captured (state and action data) and sent to the critic network. The critic network then evaluates the data received and generates a value that indicates the strength of the action taken. The information received after the critic is sent back to the robot where the current network using the critic evaluation continuously improves the action. The target network parameter θ_Q^t of critic and actor θ_μ^t are updated using soft update strategy as shown below in equation (5),

$$\begin{cases} \theta_Q^t \leftarrow \tau \theta_Q + (1 - \tau) \theta_Q' \\ \theta_\mu^t \leftarrow \tau \theta_\mu + (1 - \tau) \theta_\mu' \end{cases} \text{ where } \tau \text{ is the constant coefficient.} \tag{5}$$

7. Motion control

Motion control algorithms direct the robots to move in a particular direction as per plan. Different approaches are presented in the literature for robot motion and path control, such as kinematics, neural networks, artificial potential fields, fuzzy logic, and more, as very few papers consider motion control separate from motion planning and control.^[145-147]

Different researchers propose many different motion control approaches. Tsai *et al.*^[148] emphasized applying nonlinear control theory to construct kinematics controllers. Motion control on mobile robots, which integrates with D* and dynamic window-based approach, is discussed in Ref. [149] that provides efficient avoidance of moving obstacles. Two approaches, both inspired by the classical dynamic window algorithm, were proposed to control the robot in a dynamic environment.^[150] The article presents two algorithms dynamic window for dynamic obstacles (DW4DO) and dynamic window for dynamic obstacles Tree (DW4DOT). Hakobyan *et al.*^[151] proposed a motion control algorithm for a dynamic environment to avoid collision in an environment with moving obstacles.

A reinforcement learning approach is proposed with MC-Dropout and Bootstrapping to create uncertainty-aware navigation.^[152] Wan *et al.*^[153] proposed another deep reinforcement learning-based control mechanism. It uses an improved Lyapunov guidance vector field (LGVF) method where the sensor-based detection is directly mapped on the control signals.

A motion control design is proposed where the path is generated based on the quarter orbits particle swarm optimization algorithm, and the motion control is done based on a neural network controller.^[154] The neural network-based motion controller is presented in other articles as well.^[155,156] A reactive motion controller is proposed for mechanical arm and end-effector pose in static and dynamic environments, but control in confined spaces is challenging.^[157]

8. Hybrid algorithms

Hybrid algorithms for motion planning and control combine different algorithms to create a new approach to motion planning and control. The hybrid algorithms are developed after analyzing different algorithms' strengths and weaknesses and combining the different methods to create an improved technique.

Several hybrid algorithms are available in the literature, but mostly for motion planning and not motion control. Zhang *et al.*^[158] combined the efficiency of probabilistic roadmap methods with the completeness of approximate cell decomposition-based algorithms. The approach is appropriate for low degrees of freedom. Anavatti *et al.*^[159] proposed a hybrid algorithm combining D* Lite, Enhanced D* Lite, and A* algorithms to find the best path. Semnani *et al.*^[160] propose a hybrid control framework combining deep RL and force-based motion planning (FMP) algorithm called DRL-FMP. In the approach, the drawback of reinforcement learning, where

the algorithm cannot produce collision-free paths in dense environments, is improved by a reward function that eliminates the requirement of pre-supervised learning and reduces the chance of collision in congested environments. The proposed framework is not tested on a real robot.^[161]

A two-stage motion planning and control method is proposed in a dynamic environment.^[162] Initially, a plan is generated using the RRT* algorithm for the movement, followed by a CVaR method that assesses the safety risk and designs a constrained receding controller to track the path. The bio-inspired brain limbic system (BLS)-based control method is utilized to control the motion as per plan in an efficient manner. Another hybrid algorithm, membrane pseudo-bacterial potential field (MemPBPf), is proposed, combining pseudo-bacterial genetic algorithm, APF algorithm, and membrane computing for efficient path planning for autonomous mobile robots.^[163] Moreover, a hybrid algorithm is also presented in Ref. [164], incorporating artificial potential fields and enhanced GA for motion planning. The APF, initially using the time-efficient deterministic scheme, identifies all the feasible paths, after which the EGA analyses the path from the initial search and finds the optimal path.

A hybrid solution to the path planning problem is also proposed in Ref. [165]. A new path planning algorithm is proposed where the A* algorithm is linked to the adaptive window approach to generate cost path points to perform real-time path planning and obstacle avoidance. A hybrid differential dynamic programming framework is proposed for whole-body motion planning with legged robots.^[166]

Merits of ACO and FLA are combined to present a new algorithm known as the fuzzy logic ant colony optimization (FLACO) to find the optimal path for unmanned vehicles considering the dynamic factors like traffic flow, incident risk, and speed limit to plan the path.^[167] A fuzzy ant colony optimization (FACO) method was proposed in Ref. [168], which claims to tackle the iterative learning error of ACO by combining the fuzzy logic with ACO. A layered hybrid Probabilistic Roadmap (PRM) and the Artificial Potential Field (APF) method are used to solve the higher computation cost of the PRM. A hybrid control method is also proposed to avoid collision using DRL without having to plan for obstacles beforehand.^[169,170]

Similarly, Liu *et al.*^[171] have presented a novel fusion algorithm, a hybrid path planning technique, by integrating jump-A* with a dynamic windows approach. The combination of jump point search and dynamic windows approach plans the motion fast and avoids obstacles smoothly. A hybrid approach combined a simulated annealing algorithm and proportional-integral derivative control algorithm to plan and track the movement of a mobile robot in a static environment. The algorithm works well in a static, cluttered environment when the distance from the origin to the target is short.^[170] Rapidly exploring randomized trees and APF are combined to plan the path, avoiding 3-dimensional obstacles.^[173] In the same, an orientation potential field is also presented to control

the motion. A bio-inspired path planning algorithm is proposed in Ref. [174] that combines the new bio-inspired grey wolves algorithm and particle swarm optimization that connects the search for grey wolves and avoids local minima issues. Zhou *et al.*[175] proposed another bio-inspired algorithm, which follows the process of plant growth that proposes to solve the dynamic obstacle avoidance path planning problem in an unknown environment.

Furthermore, another hybrid motion planning and control algorithm is also proposed in Ref. [176], where the ant colony optimization is combined with the kinematic equation. In this research, the movement of the robot is planned from one point to another, and the movement is controlled to that planned step. The same happens for the second and the other consecutive steps before reaching the goal. A combination of APF and the fireworks algorithm from the improved swarm intelligent algorithm known as the dynamic enhanced firework algorithm (dynEFWA) - APF is proposed to tackle the trapping in the local minimum problem of APF.[177]

An improved Q-Learning coverage path planning algorithm, including a predator-prey model, is proposed.[178] Coverage path planning (CPP) introduces a comprehensive reward function consisting of a starting point prize, a linear reward, and the covering behavior reward to evolve the Q-learning algorithm to have the robot move straight.

Hybrid algorithms basically combine the advantages of different algorithms to reduce the drawbacks. Hybrids are efficient and cost-effective, work well in different situations, and reduce the waste of computational resources. A different combination of algorithms that forms a hybrid algorithm is summarized in Table 3.

9. Discussion

Based on the information in this study, the problem of motion planning and control is very challenging. Different algorithms based on different situations and environments have been presented to solve the motion planning and control problem. However, it has been identified that the growing environment with dynamic obstacles of different sizes and shapes with varying characteristics aggravates the problem. When planning the path, various factors like the environment boundary, obstacle information, and characteristics of the robot should be properly examined. The majority of the papers deal with objectives like path distance, smoothness of the path, collision-free path, safe path, traveling time, time complexity, and energy consumption. However, all the above objectives are not included in a single study.

Planning in a known environment where the obstacles are known and static is not a very complex task, but a partially known and unknown environment with dynamic obstacles is very challenging, as is evident in Table 4. The table tries to summarize the different articles and their limitations to provide the field with a clear understanding of the direction in which the robot path planning field is heading. Hence, many of the researchers are trying to capitalize on the strength of an

algorithm and improve on the drawbacks with the new variants (Table 4). The algorithms are mainly designed for static environments with known obstacles. Most of the proposed algorithms have not been tested in the real environment, which might not give a clear judgment of the algorithm. It can be visualized that as the complexity of the problem increases, the cost of getting the specialized resources increases.

Table 3: Different combinations of algorithms.

Algorithm Combinations to form the hybrid	Ref.
Probabilistic Roadmaps and Cell Decomposition	[158]
D* Lite, Enhanced D* Lite and A*	[159]
Deep RL and Force-based motion planning (FMP)	[160]
RRT* and bio-inspired brain limbic system (BLS)	[162]
Pseudo-bacterial genetic algorithm, APF algorithm	[163]
Artificial Potential Field and enhanced GA	[164]
A* algorithm and Adaptive Window Approach	[165]
Impact-aware DDP step, Augmented Lagrangian (AL) and Switching Time Optimization (STO) algorithm	[166]
ACO and FLA	[167]
ACO and FLA	[168]
PRM and APF	[169]
Collision Unaware Motion Planner and DRL policy	[170]
Jump-A* and dynamic windows approach	[171]
Simulated annealing algorithm and proportional-integral derivative control algorithm	[172]
RRT and APF	[173]
Bio-inspired grey wolf algorithm and PSO	[174]
ACO and Kinematic equation	[176]
APF and Fireworks algorithm	[177]
Improved Q-Learning and Coverage Path Planning (CPP)	[178]

Classical motion planning algorithms have been used by different robots in multiple environments and were computationally efficient, flexible, and very successful in static environments. It is easy to implement, cost-effective, and successfully applied in the real world. However, some of the classical algorithms find difficulties with dynamic obstacles in a dynamic environment. The classical algorithms that are further being analyzed and tested by different authors are A*, RRT, cell decomposition, APF, and the variants of each. One challenge with PRM and RRT is that the algorithms work fine in smaller configuration spaces. However, as the configuration space gets bigger, these algorithms fail to provide optimal paths.

According to researcher,[179,180] the heuristic approach is a dominant and commonly used method compared to the classical approach due to the nature of performing well in a complex dynamic environment. Heuristic approaches can learn and act based on environmental information. However, getting trapped in the local minimal is a common issue in heuristic algorithms. Moreover, the algorithm takes time to

Table 4: Summary of the articles with limitations and mechanical system used.

Mechanical System	Remarks	Limitations	Ref.
Seven Degrees of freedom manipulator	The algorithm is closely related to the optimization approach	The problem remains with the quantization of configuration parameters into intervals due to high cost.	[20]
Route selection for a mission-critical public safety network		Is well suited for 5G networks only. The research only focuses on 3 criteria (connectivity, delay, trust) however there can be other relevant criteria that are not considered.	[21]
Automated Guided vehicles in a warehouse	Multi-robot motion planning	Might not be suitable for other environments.	[22]
Transportation and logistic systems	Solve shortest path problem in an uncertain environment	Environmental data like weather and other unexpected factors can be important in determining the shortest path, which is not considered in this study.	[23]
single robot - single target and multi-robot - multitarget	surface path planning for wild ground, planet ground, and game space	Increased time cost	[28]
Omni-wheel mobile robot	combines A* and fuzzy analytic hierarchy process (FAHP)	An experiment in a real dynamic environment with relevant hardware is not tested.	[31]
Autonomous vehicle	Unknown Environment	Unable to handle traffic lights and not tested with diverse traffic like bicycles and pedestrians	[34]
Autonomous parking system	Under unstructured environment	Increased pre-processing work compared to hybrid A* which eventually increases the cost.	[36]
Industrial robot	Was tested in a static environment	Not tested in a dynamic environment with multiple robots	[41]
Differential drive mobile robot Husky	The robot is equipped with a 2-D SICK LMS100-10000 laser and 3-D Velodyne LiDAR	Becomes slow to reach the destination when obstacles are found	[47]
Mobile robot with multi-directional mobility capabilities	the experiment is done on simulated rough terrain indoors	The robot gets computationally inefficient as the number of obstacles increases which also leads to more hardware and software to perform the computation	[48]
Mobile robot	Comparison between A*, Vertical Cell Decomposition, and the proposed Radial Cell Decomposition is performed	The algorithm is tested on a static environment	[55]
XQ-4 Pro robot	An outline of the environment is given without any obstacle information	The algorithm is tested in a static environment only	(Kim, Han, Park, & Kim, 2020)
Six-DOF DENSO VP6242G robot	Can be used in a multi robot environment	The camera used have field of view only	[61]
Mobile robot	Claims to solve the APF minimum trap problem	Suitable for indoor path planning, tested on rounded shape obstacles	[63]
Mobile robot	Focus is on energy efficiency	Dynamic obstacles are not included	[59]
Autonomous mobile robot	Make safe and personalized decisions during autonomous driving	Algorithm is designed with measurement error and disturbance. Real-time experiment is not performed	[66]
Collaborative robot (KUKA LBR iiwa)		Real-time experiment is not performed	[67]
Autonomous underwater vehicle		Not tested in the real environment. The obstacle model used was very simple	[69]

Mechanical System	Remarks	Limitations	Ref.
Unmanned Aerial Vehicles		Collision is identified in the vertical direction only, horizontal needs attention	[70]
Hovercrafts and Satellites	The RRT to solve Kinodynamic planning problem	Algorithm parameters can be further altered to create optimal path. The metric function is not stable. The computation of the nearest neighbor needs attention. Configuration points might be affected while in motion, leading to poor collision detection.	[71]
Wheeled robot	Proposes a variant of RRT	Framework is not tested with a high dimensional system	[74]
6-DOF PUMA arm		The convergence rate analysis is not done	[76]
Wheeled robot (3X Khepera III)	Experiment is performed on multi-robot platform	The algorithm proposed claims to be centralized which can lead to problems like scalability, latency, security, and robustness	[77]
Warehouse mobile robot	Proposed algorithm is the black-hole potential field (BHPF)	Unable to adapt to a different environment. The algorithm is not optimized to be used in a multi-agent environment	[78]
7-DOF manipulator (Dual-arm system)	The algorithm in concern was the probabilistic roadmap	The article dealt with closed-chain constraints and not the optimal path	[81]
8-DOF modular manipulator	Is an improved PRM method	The algorithm is improved which works well with fixed-base manipulators in terms of obstacle-free path planning	[82]
6-DOF Universal robot and 7-DOF KUKA LWR robot	Hierarchical dynamic roadmap (HDRM)	Works in a static environment and does not adapt to runtime changes	[83]
Planar robot 3D workspace robot		Unable to solve narrow passage problems and is not capable of fine motion planning. Incremental methods are not updated where the representation changes dynamically when the environment frequently changes	[84]
reconfigurable robots	hTetro-GA algorithm is proposed	Real-time analysis is required to present comparison with traditional algorithms like A* and D*	[92]
Autonomous-guided vehicles	Application presented for indoor factory environment	The fleet of AGVs is not tested in a dynamic environment	[93]
Mobile robot	Domain Knowledge Based Genetic Algorithm for single and multi-target is proposed	The algorithm cannot be used in a dynamic environment. Parameters like safety, smoothness, and energy consumption are not considered.	[94]
Multi-unmanned aerial vehicle (UAV)	Genetic trajectory planning algorithm with variable population size (GTPA-VP)	Other parameters apart from energy consumption can be considered	[95]
non-omnidirectional mobile robot	Improved DWA is proposed	The algorithm is not tested in a dynamic environment	[104]
mobile robot	continuous high-degree Bezier curve with an improved PSO algorithm is proposed in study	Practical effectiveness is missing. High-order Bezier curve cannot be used in a dynamic environment	[109]
indoor mobile robot	Improvement adaptive ant colony algorithm (IAACO) is proposed in this study	Practical effectiveness is missing	[114]
USVs specifically marine transportation	Improved ACO with fuzzy logic is proposed	Modelling of dynamic obstacles is performed using elliptic trajectories, it can mean that the algorithm will not be able to easily adapt to the changing environment	[115]
AGV (Parking System)	An improved ACO algorithm is	The algorithm's efficiency reduces as nodes	[116]

Mechanical System	Remarks	Limitations	Ref.
	proposed	increase more	
Mobile robot (multiple)	Distributed multi-mobile robot obstacle-avoidance algorithm is proposed thus combining ACO and DWA	Not applicable in a dynamic obstacle environment	[118]
Mobile robot	Hybridization of Invasive weed optimization (IWO) and firefly (FA)	Not applicable with dynamic obstacles	(Yan, 2008)
Mobile robot	Firefly algorithm with self-adaptive population size	The running time of the proposed algorithm compared to the FA algorithm with fixed population time is unstable	[124]
Three-bar truss optimal design	Firefly algorithm 1 to 3 (FA1→3)	The proposed algorithm has a low convergence speed, and some functions are trapped in the local optimal solution trap	[125]
Mobile robot	Deep RL-based mobile robot path planning method	Not tested with a real mobile robot	[129]
Mobile robot		Not tested in a real environment as it is impossible to run the neural network learning method	[130]
Mobile robot	It is a reinforcement learning method. Each round of path planning is based on a Q-learning algorithm (EPRQL)	The algorithm is designed to work in a static environment only.	[132]
Turtlebot3 burger robot	The proposed algorithm is Experience-Memory Q-Learning (EMQL) algorithm	The algorithm can work in a single environment, but the robot needs to relearn in a new environment	[136]
A Four-wheeled ground differential robot is used as the mechanical system	The proposed algorithm is Lyapunov-based SAC with collision probability prediction (LSAC-CPP) algorithm	The algorithm is not tested in a real uncertain and dynamic environment	[138]
Guava harvesting robot	A recurrent Deep deterministic policy gradient algorithm (DDPG) is proposed	The algorithm is unable to plan the path when multiple obstacles surround the target. The controller fails to implement the plan. Some targets were not accurately located due to sensing power.	[141]
Mobile robot	LSTM-DDPG and MN-LSTM-DDPG algorithms are proposed	Dynamic obstacles are not considered	[142]
2-DOF and 3-DOF open manipulator	A policy gradient-based path planning algorithm that is DDPG with HER is proposed	Multiple robotic arms with obstacles in the same environment need to be tested. Testing the algorithm in a dynamic environment is also proposed to be done in the future	(Kim, Han, Park, & Kim, 2020)
Wheeled mobile robot	Model predictive control (MPC) algorithm is proposed	The algorithm is not efficient as a lot of calculation is required to be done, tracking accuracy of the algorithm is poor and the roller skidding of wheeled mobile robots requires more in-depth study	[145]
UGVs	Dynamic-window for dynamic obstacles tree (DW4DOT)	Computational requirements increase	[150]
Wheeled mobile robot	Quarter orbits particle swarm optimization (QOPSO)	Practical testing is not performed	[154]
3-DOF and 4-DOF robots	Planning is done with static obstacles	The algorithm is applicable to the low degree of freedom (DOF) robots as approximate cell	[158]

Mechanical System	Remarks	Limitations	Ref.
Multiple agents	DRL-FMP Hybrid algorithm	decomposition (ACD) is complex The algorithm is unable to generate continuous output, and it is also not tested on a real robot	[160]
Multiple mobile robots	Enhanced genetic algorithm (EGA)	Dynamic obstacles are not included in the testing of the algorithm	[164]
Mobile robots	Hybrid system differential Dynamic programming (HS-DDP)	The proposed algorithm is not been tested in a real environment	[166]
Autonomous vehicles	Fuzzy logic ant colony optimization (FLACO)	The algorithm is suited for a static environment	[167]
Mobile Robots	Simulated annealing and Proportional-integral-derivative control algorithms are combined	The algorithm is not tested in a dynamic environment	[172]
7-DOF Kinova JACO robotic arm and Kinova Gen3 robotic arm	Combines rapidly exploring randomized trees and artificial potential fields	The algorithm is unable to find the optimal path. some generated paths may be impractical due to mechanical limitations	[173]
Point-mass robot	ACO-Kinematic algorithm	The algorithm is not tested with dynamic obstacles and static obstacles with unknown obstacle location	[176]
Mobile robot	Predator-Prey reward-based Q-Learning Coverage Path Planning (PP-Q-Learning based CPP)	The algorithm is tested only with static obstacles	[178]
Unmanned ground vehicle (UGV)	Teaching-learning-based optimization (TLBO)	The experiment is not conducted in a dynamic environment with dynamic and irregular shape obstacles	[179]
Turtlebot2 mobile robot	Secure tunnel fast marching tree motion planning algorithm (ST-FMT*)	Is not tested in an unknown dynamic environment	[180]

learn before execution, and computation and calculation are quite challenging and costly.

Machine learning algorithms are being highly considered in motion planning and control problems of robots. Compared to classical and heuristic approaches, this approach is more robust, suitable for the dynamic environment, and works well with multiple robots compared to classical and heuristic approaches. However, the challenges it faces are complex computation and calculation, and it requires training data that might not be relevant in different environments, leading to inefficiency. The majority of the machine learning algorithms are not implemented to be used in the real environment, which might have a different output than the algorithm simulations, so the true worth of the algorithm application in the real world is not portrayed.

It is gathered from the literature that motion planning algorithms should be able to solve problems for any type of environment. It should be efficient, wherein the algorithm should not waste computational resources and load the memory with unnecessary data. Together with that, it should provide the best cost-effective path that should be fast, safe and energy efficient. It should also be noted that the robots now should be secured from intrusion attacks and be able to execute a function that keeps the robot under control in

situations where the robots get damaged with some functionalities not working and become unsafe for the environment and people. The summary on the merits and demerits of the of the different algorithms are summarized in [Table 5](#). Hybrid algorithms are another area that has attracted the attention of many researchers; it was identified that algorithms proposed for motion planning and control have their own benefits and drawbacks based on different environments. For instance, Hybrid A* is required to deal with the problem of collision-avoidance safety and searching efficiency. That means the algorithm is slow because it generates a lot of paths, and some paths are so close to the obstacles, which can lead to collision.

Nonetheless, the merits of combining two or more algorithms to create a new hybrid algorithm to solve the path planning problem is a promising area of research. Hybridization is very popular among researchers and is effective for path planning problems. Combining different algorithms to be applied in known, partially known, and unknown environments is an area that needs to be investigated for new hybrid algorithms. The hybrid of any combination of classical, heuristic and machine learning still requires to be thoroughly studied.

Table 5: Comparison of different algorithms.

Name of Algorithm	Advantages	Disadvantages
A*	Efficient, flexible, simple, have deterministic nature	Not applicable in dynamic environment
Dijkstra	simple, flexible, efficient, have deterministic nature, performance does not depend on initial solution	Creates multiple nodes, which increases computational time
D*	Efficient in path searching in dynamic environment	Not effective in complex environment
Artificial potential fields	Efficient implementation is simple	Mostly tested in point obstacle environment, do not consider orientation of the end-effector during motion planning, traps in local minimal
Reinforcement Learning	Can process unstructured data, relatively low computational requirements	Safety, difficult to predict the performance in real environment, Time consuming to design
RRT	Fast, easily implemented	Irregular path generation, not efficient in complex environment
Genetic Algorithm	Efficient	Not applicable for dynamic environment
Neural Network	Can learn, retraining capabilities	Lot of training required, computationally inefficient
Firefly Algorithm	Efficient in certain situation	Do not provide optimal guarantee in all application
Ant Colony Optimization	Efficient	Slow
Particle swarm optimization	Fast, robust, simple	Traps in local minimal

10. Conclusion and perspective

This paper presented popular motion planning and control algorithms under classic, heuristic, and machine learning approaches. The robot motion planning field is vast and dynamic, with different algorithms performing path planning and control in different environments. Over the past years, review articles were published based on similar areas of research. However, none of the papers consider path planning and motion control algorithms together. In this article, a holistic approach is taken to explore the recent advancements and the path planning problems. The study reviewed the different approaches analysing the advantages, limitations, and applications. It was noted in the research that:

- Different methods are developed for different environments, more specifically, the known, unknown and partially known environments.
- Algorithm attributes like computational complexity, optimality, cost, adaptability in dynamic environments, and real-time performance were the major focus, hence, suggested that:
- More emphasis should be given to algorithms that respond to changes in the environment. The study also presents recent developments in path planning problems using hybrid algorithms combining the strengths of classical, heuristic, and machine learning methods. Thus, it can be concluded that:

- Hybrid algorithms can have positive impacts in solving the path planning problem. This research is important as it provide insights on the historical development and recent advancement to the motion planning and control problems. The study will help other researchers to understand the current trend and provide directions for future research, motivating and encouraging further research in this field.

This perspective provides a summary of the key findings of this research and provides a prospective outlook on the potential of hybrid algorithms in addressing path planning and control challenges.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

- [1] J. Xia, Z. Jiang, H. Zhang, R. Zhu, H. Tian, Dual fast marching tree algorithm for human-like motion planning of anthropomorphic arms with task constraints, *IEEE/ASME Transactions on Mechatronics*, 2021, **26**, 2803-2813, doi: 10.1109/TMECH.2020.3047476.
- [2] M. Ford, Rise of the Robots, 2015.
- [3] J. Yang, J. Ni, Y. Li, J. Wen, D. Chen, The intelligent path

- planning system of agricultural robot *via* reinforcement learning, *Sensors*, 2022, **22**, 4316, doi: 10.3390/s22124316.
- [4] A. Zemmar, A. M. Lozano, B. J. Nelson, The rise of robots in surgical environments during COVID-19, *Nature Machine Intelligence*, 2020, **2**, 566-572, doi: 10.1038/s42256-020-00238-2.
- [5] Y. Tang, M. Chen, C. Wang, L. Luo, J. Li, G. Lian, X. Zou, Recognition and localization methods for vision-based fruit picking robots: a review, *Frontiers in Plant Science*, 2020, **11**, 510, doi: 10.3389/fpls.2020.00510.
- [6] J. Wirtz, P. G. Patterson, W. H. Kunz, T. Gruber, V. N. Lu, S. Paluch, A. Martins, Brave new world: service robots in the frontline, *Journal of Service Management*, 2018, **29**, 907-931, doi: 10.1108/josm-04-2018-0119.
- [7] J. Li, M. Ran, L. Xie, Efficient trajectory planning for multiple non-holonomic mobile robots *via* prioritized trajectory optimization, *IEEE Robotics and Automation Letters*, 2021, **6**, 405-412, doi: 10.1109/LRA.2020.3044834.
- [8] G. Lei, Y. Zheng, Research on cooperative trajectory planning algorithm based on tractor-trailer wheeled robot, *IEEE Access*, 2022, **10**, 64209-64221, doi: 10.1109/ACCESS.2021.3062392.
- [9] T. Mikolajczyk, E. Mikołajewska, H. F. N. Al-Shuka, T. Malinowski, A. Kłodowski, D. Y. Pimenov, T. Paczkowski, F. Hu, K. Giasin, D. Mikołajewski, M. Macko, Recent advances in bipedal walking robots: review of gait, drive, sensors and control systems, *Sensors*, 2022, **22**, 4440, doi: 10.3390/s22124440.
- [10] A. K. Varma, J. Karjee, H. K. Rath, A. Pal, Dynamic path selection for cloud-based multi-hop multi-robot wireless networks, *IETE Technical Review*, 2020, **37**, 98-107, doi: 10.1080/02564602.2019.1566031.
- [11] K. Bergman, O. Ljungqvist, D. Axehill, Improved path planning by tightly combining lattice-based path planning and optimal control, *IEEE Transactions on Intelligent Vehicles*, 2021, **6**, 57-66, doi: 10.1109/TIV.2020.2991951.
- [12] H. Hailong, A. V. Savkin, C. Huang, Wireless communication networks supported by autonomous UAVs and mobile ground robots, 2022.
- [13] X. Xiao, B. Liu, G. Warnell, P. Stone, Motion planning and control for mobile robot navigation using machine learning: a survey, *Autonomous Robots*, 2022, **46**, 569-597, doi: 10.1007/s10514-022-10039-8.
- [14] H.-Y. Zhang, W.-M. Lin, A.-X. Chen, Path planning for the mobile robot: a review, *Symmetry*, 2018, **10**, 450, doi: 10.3390/sym10100450.
- [15] J. J. M. Lunenburg, S. A. M. Coenen, G. J. L. Naus, M. J. G. van de Molengraft, M. Steinbuch, Motion planning for mobile robots: a method for the selection of a combination of motion-planning algorithms, *IEEE Robotics & Automation Magazine*, 2016, **23**, 107-117, doi: 10.1109/mra.2015.2510798.
- [16] M. Elbanhawi, M. Simic, Sampling-based robot motion planning: A review, *IEEE Access*, 2014, **2**, 56-77, doi: 10.1109/ACCESS.2014.2302442.
- [17] C. Zhou, B. Huang, P. Fränti, A review of motion planning algorithms for intelligent robots, *Journal of Intelligent Manufacturing*, 2022, **33**, 387-424, doi: 10.1007/s10845-021-01867-z.
- [18] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, A. Jagadeesh, A review: on path planning strategies for navigation of mobile robot, *Defence Technology*, 2019, **15**, 582-606, doi: 10.1016/j.dt.2019.04.011.
- [19] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik*, 1959, **1**, 269-271, doi: 10.1007/BF01386390.
- [20] T. Lozano-Pérez, M. A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, *Communications of the ACM*, 1979, **22**, 560-570, doi: 10.1145/359156.359164.
- [21] J. Tang, G. Chen, J. P. Coon, Route selection based on connectivity-delay-trust in public safety networks, *IEEE Systems Journal*, 2019, **13**, 1558-1567, doi: 10.1109/JSYST.2018.2813929.
- [22] Z. Zhang, Q. Guo, J. Chen, P. Yuan, Collision-free route planning for multiple AGVs in an automated warehouse based on collision classification, *IEEE Access*, 2018, **6**, 26022-26035, doi: 10.1109/ACCESS.2018.2819199.
- [23] Y. Deng, Y. Chen, Y. Zhang, S. Mahadevan, Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment, *Applied Soft Computing*, 2012, **12**, 1231-1237, doi: 10.1016/j.asoc.2011.11.011.
- [24] M. Luo, X. Hou, J. Yang, Surface optimal path planning using an extended dijkstra algorithm, *IEEE Access*, 2020, **8**, 147827-147838, doi: 10.1109/ACCESS.2020.3015976.
- [25] E. Gough, A. T. Conn, J. Rossiter, Planning for a tight squeeze: navigation of morphing soft robots in congested environments, *IEEE Robotics and Automation Letters*, 2021, **6**, 4752-4757, doi: 10.1109/LRA.2021.3067594.
- [26] R. Holladay, O. Salzman, S. Srinivasa, Minimizing task-space Fréchet error *via* efficient incremental graph search, *IEEE Robotics and Automation Letters*, 2019, **4**, 1999-2006, doi: 10.1109/LRA.2019.2899668.
- [27] K. Karur, N. Sharma, C. Dharmatti, J. E. Siegel, A survey of path planning algorithms for mobile robots, *Vehicles*, 2021, **3**, 448-468, doi: 10.3390/vehicles3030027.
- [28] P. E. Hart, N. J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics*, 1968, **4**, 100-107, doi: 10.1109/TSSC.1968.300136.
- [29] W. Y. H. Adoni, T. Nahhal, B. Aghezaf, A. Elbyed, The MapReduce-based approach to improve the shortest path computation in large-scale road networks: the case of A* algorithm, *Journal of Big Data*, 2018, **5**, 16, doi: 10.1186/s40537-018-0125-8.
- [30] A. K. Guruji, H. Agarwal, D. K. Parsediya, Time-efficient A* algorithm for robot path planning, *Procedia Technology*, 2016, **23**, 144-149, doi: 10.1016/j.protcy.2016.03.010.
- [31] C. Kim, J. Suh, J.-H. Han, Development of a hybrid path planning algorithm and a bio-inspired control for an omni-wheeled mobile robot, *Sensors*, 2020, **20**, 4258, doi: 10.3390/s20154258.
- [32] C. Sun, Q. Li, B. Li, L. Li, A successive linearization in feasible set algorithm for vehicle motion planning in unstructured

- and low-speed scenarios, *IEEE Transactions on Intelligent Transportation Systems*, 2022, **23**, 3724-3736, doi: 10.1109/TITS.2020.3041075.
- [33] P. E. Hart, N. J. Nilsson, B. Raphael, Correction to a formal basis for the heuristic determination of minimum cost paths, *ACM SIGART Bulletin*, 1972, **37**, 28-29, doi: 10.1145/1056777.1056779.
- [34] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, S. Thrun, Junior: the stanford entry in the urban challenge, *Journal of Field Robotics*, 2008, **25**, 569-597, doi: 10.1002/rob.20258.
- [35] D. Dolgov, S. Thrun, M. Montemerlo, J. Diebel, Path planning for autonomous vehicles in unknown semi-structured environments, *The International Journal of Robotics Research*, 2010, **29**, 485-501, doi: 10.1177/0278364909359210.
- [36] S. Sedighi, D.-V. Nguyen, K.-D. Kuhnert, Guided hybrid A-star path planning algorithm for valet parking applications, 2019 5th International Conference on Control, Automation and Robotics (ICCAR), April 19-22, 2019, Beijing, China. IEEE, 2019.
- [37] S. Koenig, M. Likhachev, D. Furcy, Lifelong planning A, *Artificial Intelligence*, 2004, **155**, 93-146, doi: 10.1016/j.artint.2003.12.001.
- [38] A. C. Coolen, R. Kühn, P. Sollich, Theory of neural information processing systems, 2005.
- [39] R. Holte, M. Perez, R. Zimmer, A. MacDonald, Hierarchical A*: searching abstraction hierarchies efficiently, 1996.
- [40] L. Cheng, C. Liu, B. Yan, Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot, 2014 IEEE International Conference on Information and Automation, July 28-30, 2014, Hailar, China, IEEE, 2014, 695-698, doi: 10.1109/ICInfA.2014.6932742.
- [41] B. Fu, L. Chen, Y. Zhou, D. Zheng, Z. Wei, J. Dai, H. Pan, An improved A* algorithm for the industrial robot path planning with high success rate and short length, *Robotics and Autonomous Systems*, 2018, **106**, 26-37, doi: 10.1016/j.robot.2018.04.007.
- [42] A. Stentz, Optimal and efficient path planning for partially-known environments, Proceedings of the 1994 IEEE International Conference on Robotics and Automation. May 8-13, 1994, San Diego, CA, USA. IEEE, 1994, 3310-3317, doi: 10.1109/ROBOT.1994.351061.
- [43] S. M. LaValle, Planning algorithms. Cambridge: Cambridge University Press, 2006.
- [44] Stentz A. The focussed d^{*} algorithm for real-time replanning, *International Joint Conference on Artificial Intelligence*, 1995, **95**, 1652-1659.
- [45] S. Koenig, M. Likhachev, Fast replanning for navigation in unknown terrain, *IEEE Transactions on Robotics*, 2005, **21**, 354-363, doi: 10.1109/TRO.2004.838026.
- [46] D. Ferguson, A. Stentz, Using interpolation to improve path planning: the field d^{*} algorithm, *Journal of Field Robotics*, 2006, **23**, 79-101, doi: 10.1002/rob.20109.
- [47] I. Maurović, M. Seder, K. Lenac, I. Petrović, Path planning for active SLAM based on the D* algorithm with negative edge weights, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, **48**, 1321-1331, doi: 10.1109/TSMC.2017.2668603.
- [48] B. Sebastian, P. Ben-Tzvi, Physics based path planning for autonomous tracked vehicle in challenging terrain, *Journal of Intelligent & Robotic Systems*, 2019, **95**, 511-526, doi: 10.1007/s10846-018-0851-3.
- [49] Choset H, Pignon P. Coverage path planning: the boustrophedon cellular decomposition, Field and Service Robotics, 1998.
- [50] S. K. Debnath, R. Omar, S. Bagchi, E. N. Sabudin, M. H. A. Shee Kandar, K. Foysol, T. K. Chakraborty, Different cell decomposition path planning methods for unmanned air vehicles-a review, Lecture Notes in Electrical Engineering, 2020.
- [51] R. A. Brooks, T. Lozano-Pérez, A subdivision algorithm in configuration space for findpath with rotation, *IEEE Transactions on Systems, Man, and Cybernetics*, 1985, **SMC-15**, 224-233, doi: 10.1109/TSMC.1985.6313352.
- [52] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, *Journal of Computational Physics*, 2006, **219**, 912-929, doi: 10.1016/j.jcp.2006.07.019.
- [53] L. S. C. Pun-Cheng, M. Y. F. Tang, I. K. L. Cheung, Exact cell decomposition on base map features for optimal path finding, *International Journal of Geographical Information Science*, 2007, **21**, 175-185, doi: 10.1080/13658810600852206.
- [54] B. Dugarjav, S.-G. Lee, D. Kim, J. H. Kim, N. Y. Chong, Scan matching online cell decomposition for coverage path planning in an unknown environment, *International Journal of Precision Engineering and Manufacturing*, 2013, **14**, 1551-1558, doi: 10.1007/s12541-013-0209-5.
- [55] O. A. A. Salama, M. E. H. Eltaib, H. A. Mohamed, O. Salah, RCD: radial cell decomposition algorithm for mobile robot path planning, *IEEE Access*, 2021, **9**, 149982-149992, doi: 10.1109/ACCESS.2021.3125105.
- [56] D. Wu, M. G. Ierapetritou, Decomposition approaches for the efficient solution of short-term scheduling problems, *Computers & Chemical Engineering*, 2003, **27**, 1261-1276, doi: 10.1016/S0098-1354(03)00051-6.
- [57] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *Autonomous Robot Vehicles*, 1986.
- [58] L. Chang, L. Shan, C. Jiang, Y. Dai, Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment, *Autonomous Robots*, 2021, **45**, 51-76, doi: 10.1007/s10514-020-09947-4.
- [59] R. Szczepanski, T. Tarczewski, K. Erwinski, Energy efficient local path planning algorithm based on predictive artificial potential field, *IEEE Access*, 2022, **10**, 39729-39742, doi: 10.1109/ACCESS.2022.3166632.
- [60] B. Sharma, J. Vanualailai, S. Singh, Lyapunov-based nonlinear controllers for obstacle avoidance with a planar n-link doubly nonholonomic manipulator, *Robotics and Autonomous Systems*, 2012, **60**, 1484-1497, doi: 10.1016/j.robot.2012.07.014.

- [61] B. Ahmadi, E. Zakeri, W.-F. Xie, Optimal image-based task-sequence/path planning and robust hybrid vision/force control of industrial robots, *IEEE Access*, 2022, **10**, 26347-26368, doi: 10.1109/ACCESS.2022.3156919.
- [62] S. M. H. Rostami, A. K. Sangaiah, J. Wang, X. Liu, Obstacle avoidance of mobile robots using modified artificial potential field algorithm, *EURASIP Journal on Wireless Communications and Networking*, 2019, **2019**, 70, doi: 10.1186/s13638-019-1396-2.
- [63] J. Luo, Z.-X. Wang, K.-L. Pan, Reliable path planning algorithm based on improved artificial potential field method, *IEEE Access*, 2022, **10**, 108276-108284, doi: 10.1109/ACCESS.2022.3212741.
- [64] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, C. Hu, A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach, *IEEE Transactions on Industrial Electronics*, 2020, **67**, 1376-1386, doi: 10.1109/TIE.2019.2898599.
- [65] W. Wang, M. Zhu, X. Wang, S. He, J. He, Z. Xu, An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators, *International Journal of Advanced Robotic Systems*, 2018, **15**, 172988141879956, doi: 10.1177/1729881418799562.
- [66] C. Huang, C. Lv, P. Hang, Y. Xing, Toward safe and personalized autonomous driving: decision-making and motion control with DPF and CDT techniques, *IEEE/ASME Transactions on Mechatronics*, 2021, **26**, 611-620, doi: 10.1109/TMECH.2021.3053248.
- [67] G. Palmieri, C. Scoccia, Motion planning and control of redundant manipulators for dynamical obstacle avoidance, *Machines*, 2021, **9**, 121, doi: 10.3390/machines9060121.
- [68] S. Xie, J. Hu, P. Bhowmick, Z. Ding, F. Arvin, Distributed motion planning for safe autonomous vehicle overtaking via artificial potential field, *IEEE Transactions on Intelligent Transportation Systems*, 2022, **23**, 21531-21547, doi: 10.1109/TITS.2022.3189741.
- [69] S. Sun, H. Guo, G. Wan, C. Dong, C. Zheng, Y. Wang, High-precision underwater acoustic localization of the black box utilizing an autonomous underwater vehicle based on the improved artificial potential field, *IEEE Transactions on Geoscience and Remote Sensing*, 2023, **61**, 4202510, doi: 10.1109/TGRS.2023.3249199.
- [70] H. M. Jayaweera, S. Hanoun, A dynamic artificial potential field (D-APF) UAV path planning technique for following ground moving targets, *IEEE Access*, 2020, **8**, 192760-192776, doi: 10.1109/ACCESS.2020.3032929.
- [71] S. M. LaValle, J. J. Kuffner, Randomized kinodynamic planning, Proceedings 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 1999.
- [72] H. Nichols, M. Jimenez, Z. Goddard, M. Sparapany, B. Boots, A. Mazumdar, Adversarial sampling-based motion planning, *IEEE Robotics and Automation Letters*, 2022, **7**, 4267-4274, doi: 10.1109/LRA.2022.3148464.
- [73] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, *The International Journal of Robotics Research*, 2011, **30**, 846-894, doi: 10.1177/0278364911406761.
- [74] B. Hu, Z. Cao, M. Zhou, An efficient RRT-based framework for planning short and smooth wheeled robot motion under kinodynamic constraints, *IEEE Transactions on Industrial Electronics*, 2021, **68**, 3292-3302, doi: 10.1109/TIE.2020.2978701.
- [75] C. Jiang, Z. Hu, Z. P. Mourelatos, D. Gorsich, P. Jayakumar, Y. Fu, M. Majcher, R2-RRT*: reliability-based robust mission planning of off-road autonomous ground vehicle under uncertain terrain environment, *IEEE Transactions on Automation Science and Engineering*, 2022, **19**, 1030-1046, doi: 10.1109/TASE.2021.3050762.
- [76] J. J. Kuffner, S. M. LaValle, RRT-connect: an efficient approach to single-query path planning, IEEE International Conference on Robotics and Automation, Symposia Proceedings, April 24-28, 2000, San Francisco, CA, USA. IEEE, 2000.
- [77] G. Zhao, M. Zhu, Pareto optimal multirobot motion planning, *IEEE Transactions on Automatic Control*, 2021, **66**, 3984-3999, doi: 10.1109/TAC.2020.3025870.
- [78] Q. Yao, Z. Zheng, L. Qi, H. Yuan, X. Guo, M. Zhao, Z. Liu, T. Yang, Path planning method with improved artificial potential field—a reinforcement learning perspective, *IEEE Access*, 2020, **8**, 135513-135523, doi: 10.1109/ACCESS.2020.3011211.
- [79] L. E. Kavraki, P. Svestka, J.-C. Latombe, M. H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation*, 1996, **12**, 566-580, doi: 10.1109/70.508439.
- [80] Z. Wang, J. Cai, Probabilistic roadmap method for path-planning in radioactive environment of nuclear facilities, *Progress in Nuclear Energy*, 2018, **109**, 113-120, doi: 10.1016/j.pnucene.2018.08.006.
- [81] K. Jang, J. Baek, S. Park, J. Park, Motion planning for closed-chain constraints based on probabilistic roadmap with improved connectivity, *IEEE/ASME Transactions on Mechatronics*, 2022, **27**, 2035-2043, doi: 10.1109/TMECH.2022.3175260.
- [82] G. Chen, N. Luo, D. Liu, Z. Zhao, C. Liang, Path planning for manipulators based on an improved probabilistic roadmap method, *Robotics and Computer-Integrated Manufacturing*, 2021, **72**, 102196, doi: 10.1016/j.rcim.2021.102196.
- [83] Y. Yang, W. Merkt, V. Ivan, Z. Li, S. Vijayakumar, HDRM: A resolution complete dynamic roadmap for real-time motion planning in complex scenes, *IEEE Robotics and Automation Letters*, 2018, **3**, 551-558, doi: 10.1109/LRA.2017.2773669.
- [84] P. Leven, S. Hutchinson, A framework for real-time path planning in changing environments, *The International Journal of Robotics Research*, 2002, **21**, 999-1030, doi: 10.1177/0278364902021012001.
- [85] Y. Du, X. Zhang, Z. Nie, A real-time collision avoidance strategy in dynamic airspace based on dynamic artificial potential field algorithm, *IEEE Access*, 2019, **7**, 169469-169479, doi: 10.1109/ACCESS.2019.2953946.
- [86] D. Berenson, S. Srinivasa, J. Kuffner, Task space regions, *The International Journal of Robotics Research*, 2011, **30**, 1435-1460, doi: 10.1177/0278364910396389.

- [87] L. Jaillet, J. M. Porta, Path planning under kinematic constraints by rapidly exploring manifolds, *IEEE Transactions on Robotics*, 2013, **29**, 105-117, doi: 10.1109/TRO.2012.2222272.
- [88] B. Kim, T. T. Um, C. Suh, F. C. Park, Tangent bundle RRT: A randomized algorithm for constrained motion planning, *Robotica*, 2016, **34**, 202-225, doi: 10.1017/s0263574714001234.
- [89] J. H. Holland, Adaptation in natural and artificial systems:: an introductory analysis with applications to biology, control, and artificial intelligence, 1992.
- [90] T. T. Mac, C. Copot, D. T. Tran, R. De Keyser, Heuristic approaches in robot path planning: a survey, *Robotics and Autonomous Systems*, 2016, **86**, 13-28, doi: 10.1016/j.robot.2016.08.001.
- [91] B. K. Patle, D. R. K. Parhi, A. Jagadeesh, S. K. Kashyap, Matrix-Binary Codes based Genetic Algorithm for path planning of mobile robot, *Computers & Electrical Engineering*, 2018, **67**, 708-728, doi: 10.1016/j.compeleceng.2017.12.011.
- [92] Y. Liu, C. Guo, Y. Weng, Online time-optimal trajectory planning for robotic manipulators using adaptive elite genetic algorithm with singularity avoidance, *IEEE Access*, 2019, **7**, 146301-146308, doi: 10.1109/ACCESS.2019.2945824.
- [93] K. P. Cheng, R. E. Mohan, N. H. Khanh Nhan, A. V. Le, Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot, *IEEE Access*, 2020, **8**, 121267-121284, doi: 10.1109/ACCESS.2020.3006579.
- [94] P. T. Zacharia, E. K. Xidias, AGV routing and motion planning in a flexible manufacturing system using a fuzzy-based genetic algorithm, *The International Journal of Advanced Manufacturing Technology*, 2020, **109**, 1801-1813, doi: 10.1007/s00170-020-05755-3.
- [95] R. Sarkar, D. Barman, N. Chowdhury, Domain knowledge based genetic algorithms for mobile robot path planning having single and multiple targets, *Journal of King Saud University - Computer and Information Sciences*, 2022, **34**, 4269-4283, doi: 10.1016/j.jksuci.2020.10.010.
- [96] M. Asim, W. K. Mashwani, S. B. Belhaouari, S. Hassan, A novel genetic trajectory planning algorithm with variable population size for multi-UAV-assisted mobile edge computing system, *IEEE Access*, 2021, **9**, 125569-125579, doi: 10.1109/ACCESS.2021.3111318.
- [97] K. S. Suresh, R. Venkatesan, S. Venugopal, Mobile robot path planning using multi-objective genetic algorithm in industrial automation, *Soft Computing*, 2022, **26**, 7387-7400, doi: 10.1007/s00500-022-07300-8.
- [98] L. A. Zadeh, Fuzzy sets, *Information and Control*, 1965, **8**, 338-353, doi: 10.1016/S0019-9958(65)90241-X.
- [99] L. Cherroun, M. Boumehraz, A. Kouzou, Mobile robot path planning based on optimized fuzzy logic controllers, *New Developments and Advances in Robot Control*, 2019, **175**, 255-283, doi: 10.1007/978-981-13-2212-9_12.
- [100] K. M. Passino, Fuzzy Control, 1998.
- [101] M. Nadour, M. Boumehraz, L. Cherroun, V. Puig, Mobile robot visual navigation based on fuzzy logic and optical flow approaches, *International Journal of System Assurance Engineering and Management*, 2019, **10**, 1654-1667, doi: 10.1007/s13198-019-00918-2.
- [102] Y. Chang, Y. Shi, A. Dostovalova, Z. Cao, J. Kim, D. Gibbons, C. Lin, Interpretable fuzzy logic control for multirobot coordination in a cluttered environment, *IEEE Transactions on Fuzzy Systems*, 2021, **29**, 3676-3685, doi: 10.1109/TFUZZ.2021.3111446.
- [103] M. Abdelwahab, V. Parque, A. M. R. Fath Elbab, A. A. Abouelsoud, S. Sugano, Trajectory tracking of wheeled mobile robots using Z-number based fuzzy logic, *IEEE Access*, 2020, **8**, 18426-18441, doi: 10.1109/ACCESS.2020.2968421.
- [104] S. Khesrani, A. Hassam, O. Boutalbi, M. Boubezoula, Motion planning and control of nonholonomic mobile robot using flatness and fuzzy logic concepts, *International Journal of Dynamics and Control*, 2021, **9**, 1660-1671, doi: 10.1007/s40435-020-00754-4.
- [105] L. Xiang, X. Li, H. Liu, P. Li, Parameter fuzzy self-adaptive dynamic window approach for local path planning of wheeled robot, *IEEE Open Journal of Intelligent Transportation Systems*, 2022, **3**, 1-6, doi: 10.1109/OJITS.2021.3137931.
- [106] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995.
- [107] P. Qu, F. Du, Improved particle swarm optimization for laser cutting path planning, *IEEE Access*, 2023, **11**, 4574-4588, doi: 10.1109/ACCESS.2023.3236006.
- [108] H. S. Dewang, P. K. Mohanty, S. Kundu, A robust path planning for mobile robot using smart particle swarm optimization, *Procedia Computer Science*, 2018, **133**, 290-297, doi: 10.1016/j.procs.2018.07.036.
- [109] L. Xu, M. Cao, B. Song, A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm, *Neurocomputing*, 2022, **473**, 98-106, doi: 10.1016/j.neucom.2021.12.016.
- [110] B. Song, Z. Wang, L. Zou, An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve, *Applied Soft Computing*, 2021, **100**, 106960, doi: 10.1016/j.asoc.2020.106960.
- [111] H. H. Shorey, Animal communication by pheromones, 2013.
- [112] M. Dorigo, V. Maniezzo, A. Colomi, Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 1996, **26**, 29-41, doi: 10.1109/3477.484436.
- [113] M. Dorigo, L. M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, 1997, **1**, 53-66, doi: 10.1109/4235.585892.
- [114] Q. Luo, H. Wang, Y. Zheng, J. He, Research on path planning of mobile robot based on improved ant colony algorithm, *Neural Computing and Applications*, 2020, **32**, 1555-1566, doi: 10.1007/s00521-019-04172-2.
- [115] C. Miao, G. Chen, C. Yan, Y. Wu, Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, *Computers & Industrial Engineering*, 2021, **156**, 107230, doi: 10.1016/j.cie.2021.107230.

- [116] D. V. Lyridis, An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints, *Ocean Engineering*, 2021, **241**, 109890, doi: 10.1016/j.oceaneng.2021.109890.
- [117] X. Wang, H. Shi, C. Zhang, Path planning for intelligent parking system based on improved ant colony optimization, *IEEE Access*, 2020, **8**, 65267-65273, doi: 10.1109/ACCESS.2020.2984802.
- [118] Q. Wang, J. Li, L. Yang, Z. Yang, P. Li, G. Xia, Distributed multi-mobile robot path planning and obstacle avoidance based on ACO-DWA in unknown complex terrain, *Electronics*, 2022, **11**, 2144, doi: 10.3390/electronics11142144.
- [119] X. Yan, Nature-inspired metaheuristic algorithm, 2008.
- [120] M. R. Panda, S. Dutta, S. Pradhan, Hybridizing invasive weed optimization with firefly algorithm for multi-robot motion planning, *Arabian Journal for Science and Engineering*, 2018, **43**, 4029-4039, doi: 10.1007/s13369-017-2794-6.
- [121] A. Yelghi, C. Köse, A modified firefly algorithm for global minimum optimization, *Applied Soft Computing*, 2018, **62**, 29-44, doi: 10.1016/j.asoc.2017.10.032.
- [122] H. Kundra, W. Khan, M. Malik, K. P. Rane, R. Neware, V. Jain, Quantum-inspired firefly algorithm integrated with cuckoo search for optimal path planning, *International Journal of Modern Physics C*, 2022, **33**, 2250018, doi: 10.1142/S0129183122500188.
- [123] B. K. Patle, A. Pandey, A. Jagadeesh, D. R. Parhi, Path planning in uncertain environment by using firefly algorithm, *Defence Technology*, 2018, **14**, 691-701, doi: 10.1016/j.dt.2018.06.004.
- [124] J. Henrio, T. Deligne, T. Nakashima, T. Watanabe, Route planning for multiple surveillance autonomous drones using a discrete firefly algorithm and a Bayesian optimization method, *Artificial Life and Robotics*, 2019, **24**, 100-105, doi: 10.1007/s10015-018-0454-x.
- [125] F. Li, X. Fan, Z. Hou, A firefly algorithm with self-adaptive population size for global path planning of mobile robot, *IEEE Access*, 2020, **8**, 168951-168964, doi: 10.1109/ACCESS.2020.3023999.
- [126] M. Ghasemi, S. K. Mohammadi, M. Zare, S. Mirjalili, M. Gil, R. Hemmati, A new firefly algorithm with improved global exploration and convergence with application to engineering optimization, *Decision Analytics Journal*, 2022, **5**, 100125, doi: 10.1016/j.dajour.2022.100125.
- [127] X. Chen, P. Gao, Path planning and control of soccer robot based on genetic algorithm, *Journal of Ambient Intelligence and Humanized Computing*, 2020, **11**, 6177-6186, doi: 10.1007/s12652-019-01635-1.
- [128] A. Konar, I. Goswami Chakraborty, S. J. Singh, L. C. Jain, A. K. Nagar, A deterministic improved Q-learning for path planning of a mobile robot, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2013, **43**, 1141-1153, doi: 10.1109/TSMCA.2012.2227719.
- [129] J. Xin, H. Zhao, D. Liu, M. Li, Application of deep reinforcement learning in mobile robot path planning, 2017 Chinese Automation Congress (CAC), October 20-22, 2017, Jinan, China, IEEE, 2017.
- [130] J. Yu, Y. Su, Y. Liao, The path planning of mobile robot by neural networks and hierarchical reinforcement learning, *Frontiers in Neurorobotics*, 2020, **14**, 63, doi: 10.3389/fnbot.2020.00063.
- [131] R. S. Sutton, Reinforcement learning: an introduction, 2018.
- [132] B. Wang, Z. Liu, Q. Li, A. Prorok, Mobile robot path planning in dynamic environments through globally guided reinforcement learning, *IEEE Robotics and Automation Letters*, 2020, **5**, 6932-6939, doi: 10.1109/LRA.2020.3026638.
- [133] J. Ha, B. An, S. Kim, Reinforcement learning heuristic A, *IEEE Transactions on Industrial Informatics*, 2023, **19**, 2307-2316, doi: 10.1109/TII.2022.3188359.
- [134] C. J. Watkins, Learning from delayed rewards, England: University of Cambridge, 1989.
- [135] G. Pan, Y. Xiang, X. Wang, Z. Yu, X. Zhou, Research on path planning algorithm of mobile robot based on reinforcement learning, *Soft Computing*, 2022, **26**, 8961-8970, doi: 10.1007/s00500-022-07293-4.
- [136] M. Zhao, H. Lu, S. Yang, F. Guo, The experience-memory Q-learning algorithm for robot path planning in unknown environment, *IEEE Access*, 2020, **8**, 47824-47844, doi: 10.1109/ACCESS.2020.2978077.
- [137] P. Hernandez-Leal, B. Kartal, M. E. Taylor, A survey and critique of multiagent deep reinforcement learning, *Autonomous Agents and Multi-Agent Systems*, 2019, **33**, 750-797, doi: 10.1007/s10458-019-09421-1.
- [138] L. Zhang, R. Zhang, T. Wu, R. Weng, M. Han, Y. Zhao, Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles, *IEEE Transactions on Neural Networks and Learning Systems*, 2021, **32**, 5435-5444, doi: 10.1109/TNNLS.2021.3084685.
- [139] Y. Wang, Z. He, D. Cao, L. Ma, K. Li, L. Jia, Y. Cui, Coverage path planning for kiwifruit picking robots based on deep reinforcement learning, *Computers and Electronics in Agriculture*, 2023, **205**, 107593, doi: 10.1016/j.compag.2022.107593.
- [140] T. Mikołajczyk, D. Mikołajewski, A. Kłodowski, A. Łukaszewicz, E. Mikołajewska, T. Paczkowski, M. Macko, M. Skornia, Energy sources of mobile robot power systems: a systematic review and comparison of efficiency, *Applied Sciences*, 2023, **13**, 7547, doi: 10.3390/app13137547.
- [141] G. Lin, L. Zhu, J. Li, X. Zou, Y. Tang, Collision-free path planning for a guava-harvesting robot based on recurrent deep reinforcement learning, *Computers and Electronics in Agriculture*, 2021, **188**, 106350, doi: 10.1016/j.compag.2021.106350.
- [142] H. Gong, P. Wang, C. Ni, N. Cheng, Efficient path planning for mobile robot based on deep deterministic policy gradient, *Sensors*, 2022, **22**, 3579, doi: 10.3390/s22093579.
- [143] M. Kim, D. K. Han, J. H. Park, J. S. Kim, Motion planning of robot manipulators for a smoother path using a twin delayed deep deterministic policy gradient with hindsight experience replay, *Applied Sciences*, 2020, **10**, 575, doi: 10.3390/app10020575.

- [144] H. Gong, P. Wang, C. Ni, N. Cheng, Efficient path planning for mobile robot based on deep deterministic policy gradient, *Sensors*, 2022, **22**, 3579, doi: 10.3390/s22093579.
- [145] Q. Liu, Q. Cong, Kinematic and dynamic control model of wheeled mobile robot under Internet of Things and neural network, *Journal of Supercomputing*, 2022, **78**, 8678-8707, doi: 10.1007/s11227-021-04160-1.
- [146] M. K. Muni, D. R. Parhi, P. B. Kumar, S. Kumar, Motion control of multiple humanoids using a hybridized prim's algorithm-fuzzy controller, *Soft Computing*, 2021, **25**, 1159-1180, doi: 10.1007/s00500-020-05212-z.
- [147] M. Dirik, O. Castillo, A. F. Kocamaz, Gaze-guided control of an autonomous mobile robot using type-2 fuzzy logic, *Applied System Innovation*, 2019, **2**, 14, doi: 10.3390/asi2020014.
- [148] C. C. Tsai, L. B. Jiang, T. Y. Wang, T. S. Wang, Kinematics control of an omnidirectional mobile robot, Proceedings of 2005 CACS Automatic Control Conference, Tainan, 2005.
- [149] M. Seder, I. Petrovic, Dynamic window based approach to mobile robot motion control in the presence of moving obstacles, Proceedings 2007 IEEE International Conference on Robotics and Automation, April 10-14, 2007, Rome, Italy, IEEE, 2007.
- [150] E. J. Molinos, Á. Llamazares, M. Ocaña, Dynamic window based approaches for avoiding obstacles in moving, *Robotics and Autonomous Systems*, 2019, **118**, 112-130, doi: 10.1016/j.robot.2019.05.003.
- [151] A. Hakobyan, I. Yang, Wasserstein distributionally robust motion control for collision avoidance using conditional value-at-risk, *IEEE Transactions on Robotics*, 2022, **38**, 939-957, doi: 10.1109/TRO.2021.3106827.
- [152] B. Lütjens, M. Everett, J. P. How, Safe reinforcement learning with model uncertainty estimates, 2019 International Conference on Robotics and Automation (ICRA). May 20-24, 2019, Montreal, QC, Canada. IEEE, 2019.
- [153] K. Wan, B. Li, X. Gao, Z. Hu, Z. Yang, A learning-based flexible autonomous motion control method for UAV in dynamic unknown environments, *Journal of Systems Engineering and Electronics*, 2021, **32**, 1490-1508, doi: 10.23919/JSEE.2021.000126.
- [154] Z. E. Kanoon, A. Sabah Al-Araji, M. N. Abdullah, An intelligent path planning algorithm and control strategy design for MultiMobile robots based on a modified Elman recurrent neural network, *International Journal of Intelligent Engineering and Systems*, 2022, **15**, 400-415, doi: 10.22266/ijies2022.1031.35.
- [155] C. Yang, C. Chen, W. He, R. Cui, Z. Li, Robot learning system based on adaptive neural control and dynamic movement primitives, *IEEE Transactions on Neural Networks and Learning Systems*, 2019, **30**, 777-787, doi: 10.1109/TNNLS.2018.2852711.
- [156] Z. Xie, L. Jin, X. Luo, B. Hu, S. Li, An acceleration-level data-driven repetitive motion planning scheme for kinematic control of robots with unknown structure, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022, **52**, 5679-5691, doi: 10.1109/TSMC.2021.3129794.
- [157] J. Haviland, P. Corke, NEO: A novel expeditious optimisation algorithm for reactive motion control of manipulators, *IEEE Robotics and Automation Letters*, 2021, **6**, 1043-1050, doi: 10.1109/LRA.2021.3056060.
- [158] L. Zhang, Y. J. Kim, D. Manocha, A hybrid approach for complete motion planning 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29-November 2, 2007. San Diego, CA, USA. IEEE, 2007.
- [159] S. G. Anavatti, S. Biswas, J. T. Colvin, M. Pratama, A hybrid algorithm for efficient path planning of autonomous ground vehicles, 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), November 13-15, 2016, Phuket, Thailand. IEEE, 2016.
- [160] S. H. Semnani, H. Liu, M. Everett, A. de Ruiter, J. P. How, Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning, *IEEE Robotics and Automation Letters*, 2020, **5**, 3221-3226, doi: 10.1109/LRA.2020.2974695.
- [161] S. H. Semnani, A. D. Ruiter, H. Liu, Force-based algorithm for motion planning of large agent teams, *arxiv preprint arxiv:1909.05415*, 2019.
- [162] A. Hakobyan, G. C. Kim, I. Yang, Risk-aware motion planning and control using CVaR-constrained optimization, *IEEE Robotics and Automation Letters*, 2019, **4**, 3924-3931, doi: 10.1109/LRA.2019.2929980.
- [163] U. Orozco-Rosas, K. Picos, O. Montiel, Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots, *IEEE Access*, 2019, **7**, 156787-156803, doi: 10.1109/ACCESS.2019.2949835.
- [164] M. Nazarahari, E. Khanmirza, S. Doostie, Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm, *Expert Systems with Applications*, 2019, **115**, 106-120, doi: 10.1016/j.eswa.2018.08.008.
- [165] X. Zhong, J. Tian, H. Hu, X. Peng, Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment, *Journal of Intelligent & Robotic Systems*, 2020, **99**, 65-77, doi: 10.1007/s10846-019-01112-z.
- [166] H. Li, P. M. Wensing, Hybrid systems differential dynamic programming for whole-body motion planning of legged robots, *IEEE Robotics and Automation Letters*, 2020, **5**, 5448-5455, doi: 10.1109/LRA.2020.3007475.
- [167] Q. Song, Q. Zhao, S. Wang, Q. Liu, X. Chen, Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization, *IEEE Access*, 2020, **8**, 62107-62115, doi: 10.1109/ACCESS.2020.2984695.
- [168] A. Ragmani, A. Elomri, N. Abghour, K. Moussaid, M. Rida, FACO: a hybrid fuzzy ant colony optimization algorithm for virtual machine scheduling in high-performance cloud computing, *Journal of Ambient Intelligence and Humanized Computing*, 2020, **11**, 3975-3987, doi: 10.1007/s12652-019-01631-5.
- [169] A. A. Ravankar, A. Ravankar, T. Emaru, Y. Kobayashi, HPPRM: hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots, *IEEE Access*, 2020, **8**, 221743-221766, doi: 10.1109/ACCESS.2020.3043333.
- [170] B. Sangiovanni, G. P. Incremona, M. Piastra, A. Ferrara, Self-configuring robot path planning with obstacle avoidance via

- deep reinforcement learning, *IEEE Control Systems Letters*, 2021, **5**, 397-402, doi: 10.1109/LCSYS.2020.3002852.
- [171] L. Liu, J. Yao, D. He, J. Chen, J. Huang, H. Xu, B. Wang, J. Guo, Global dynamic path planning fusion algorithm combining jump-A* algorithm and dynamic window approach, *IEEE Access*, 2021, **9**, 19632-19638, doi: 10.1109/ACCESS.2021.3052865.
- [172] T. Zahid, Z. Kausar, M. F. Shah, M. T. Saeed, J. Pan, An intelligent hybrid control to enhance applicability of mobile robots in cluttered environments, *IEEE Access*, 2021, **9**, 50151-50162, doi: 10.1109/ACCESS.2021.3068988.
- [173] A. Sepehri, A. M. Moghaddam, A motion planning algorithm for redundant manipulators using rapidly exploring randomized trees and artificial potential fields, *IEEE Access*, 2021, **9**, 26059-26070, doi: 10.1109/ACCESS.2021.3056397.
- [174] F. Gul, I. Mir, D. Alarabiat, H. M. Alabool, L. Abualigah, S. Mir, Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning, *Journal of Parallel and Distributed Computing*, 2022, **169**, 171-184, doi: 10.1016/j.jpdc.2022.06.014.
- [175] Y. Zhou, Y. Su, A. Xie, L. Kong, A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of UAV, *Chinese Journal of Aeronautics*, 2021, **34**, 199-209, doi: 10.1016/j.cja.2020.12.018.
- [176] K. Chaudhary, A. Prasad, V. Chand, B. Sharma, ACO-Kinematic: a hybrid first off the starting block, *PeerJ Computer Science*, 2022, **8**, e905, doi: 10.7717/peerj-cs.905.
- [177] H. Li, W. Liu, C. Yang, W. Wang, T. Qie, C. Xiang, An optimization-based path planning approach for autonomous vehicles using the DynEFWA-artificial potential field, *IEEE Transactions on Intelligent Vehicles*, 2022, **7**, 263-272, doi: 10.1109/TIV.2021.3123341.
- [178] M. Zhang, W. Cai, L. Pang, Predator-prey reward based Q-learning coverage path planning for mobile robot, *IEEE Access*, 2023, **11**, 29673-29683, doi: 10.1109/ACCESS.2023.3255007.
- [179] A. D. Sabiha, M. A. Kamel, E. Said, W. M. Hussein, Real-time path planning for autonomous vehicle based on teaching-learning-based optimization, *Intelligent Service Robotics*, 2022, **15**, 381-398, doi: 10.1007/s11370-022-00429-3.
- [180] Z. Wu, Y. Chen, J. Liang, B. He, Y. Wang, ST-FMT*: A Fast Optimal Global Motion, *IEEE Transactions on Industrial Electronics*, 2021, **69**, 3854-3864, doi: 10.1109/TIE.2021.3075852.
- Commons licence and changes need to be indicated if there are any. The images or other third-party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

©The Author(s) 2025

Publisher's Note: Engineered Science Publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits the use, sharing, adaptation, distribution and reproduction in any medium or format, as long as appropriate credit to the original author(s) and the source is given by providing a link to the Creative