Engineering Science and Technology, an International Journal 40 (2023) 101343

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Review

A systematic review on recent advances in autonomous mobile robot navigation

Anbalagan Loganathan, Nur Syazreen Ahmad *

School of Electrical & Electronic Engineering, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia

A R T I C L E I N F O

Article history:
Received 11 August 2022
Revised 3 November 2022
Accepted 18 January 2023
Available online 24 February 2023

Keywords:
Autonomous mobile robot
Dynamic environment
Heuristic
Navigation
Path planning
Systematic review

A B S T R A C T

Recent years have seen a dramatic rise in the popularity of autonomous mobile robots (AMRs) due to their practicality and potential uses in the modern world. Path planning is among the most important tasks in AMR navigation since it demands the robot to identify the best route based on desired performance criteria such as safety margin, shortest time, and energy consumption. The complexity of the problem can however become intractable when challenging scenarios are considered, which include navigation in a dynamic environment and solving multi-objective optimizations. Various classical and heuristic techniques have been proposed by researchers to mitigate such issues. The purpose of this paper is to provide a comprehensive and up-to-date literature review of the path planning strategies that have received a considerable attention over the past decade. A systematic analysis of the strengths, shortcomings, and scope of each method is presented. The trends as well as challenges in practical implementation of the strategies are also discussed at the end of this paper. The outcome of this survey provides useful guidance for future research into creating new strategies that can enhance the autonomy level of AMRs while preserving their robustness against unforeseen circumstances in practice.

Contents

* Corresponding author.
E-mail address: syazreen@usm.my (N.S. Ahmad).

https://doi.org/10.1016/j.jestch.2023.101343
2215-0986/

1. Introduction

Over the past few decades, the deployment of mobile robots has been a major contributor towards increasing productivity in a large variety of domains such as manufacturing, agriculture, military, and education. Recent years have also witnessed a dramatic rise in the adoption of autonomous mobile robots (AMRs) due to COVID-19 pandemic where most sectors particularly healthcare, security patrols and food industries are moving towards less human-to-human interaction and more towards human-to-machine to minimize the spread of the virus [1]. Similar to humans, AMRs are capable of independent decision-making and taking corrective actions. A robot that is completely autonomous is able to observe its surroundings, make judgments based on what it sees and/or has been trained to recognize, and then carry out an action or manipulation in that environment. As such, they have become increasingly more promising and effective compared to automated guided vehicles due to their ability to navigate in an environment without the aid of guidance devices or centralized control [2].

Among others, path planning is a crucial task in the AMR navigation which requires the robot to find an optimal path based on desired performance outcomes such as shortest time, shortest route and energy consumption. The path planning problem can be divided into two categories according to the type of environment information; namely global path planning and local path planning. The global path planning assumes complete knowledge of the environment and is best applied when the environment is static. The global path is typically generated offline before the robot begins to navigate. The local path planning on the other hand assumes incomplete knowledge of the environment, and is usually invoked online based on data from on-board sensors. It is thus more suited for navigation in unknown or dynamic environments. The most defining feature of an AMR is its ability to navigate without disruption by generating a new hazard-free local path when a previously unforeseen obstacle is detected, and re-route to ensure the change in the desired performance measure is minimized. Fig. 1 illustrates the scenario where a local path is generated to avoid an obstacle which coincidentally crosses the global path as the AMR moves towards the goal.

The applications of AMR in recent times have been shifting from industrial environments into a more challenging scenario such as surveillance, disaster response, [3,4] and domestic applications [5–7]. In these dynamic settings, it is not possible to locate all the obstacles in the environment a priori. Thus, it is necessary to equip the AMR with a robust navigational strategy that can simultaneously avoid oncoming obstacles in real-time and compute a new conflict-free path in a fast manner.

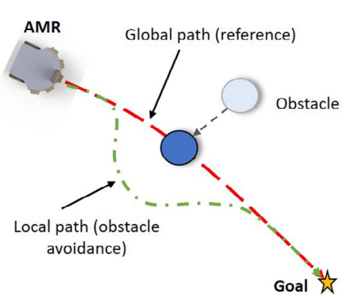


Fig. 1. Illustration on the scenario where a local path is generated to avoid an obstacle which crosses the global path as the AMR moves towards the goal.

The path planning, which is considered as a non-deterministic polynomial-time (“NP”) hard problem [8], becomes more complicated as the system’s degree of freedom rises such as navigation in a 3-dimensional (3D) environment. Other challenges include incorporation of non-holonomic or dynamic constraints in the optimization problem formulation particularly for wheeled mobile robots [9], and formulating a tractable solution for path planning of swarm or multiple mobile robots [10]. Although there is a growing number of survey papers focusing on path planning methodologies in the literature, many of these studies fall short of providing an in-depth examination of each navigational method and illustrations of state-of-the art techniques in solving common key challenges. The main contributions of this paper are as follows:

1. It provides a comprehensive and up-to-date literature review of the path planning strategies of AMRs that have received a considerable attention over the past decade.
2. It presents a systematic analysis of the strengths and shortcomings of each technique, as well as trends in practical implementation of the navigation strategies.
3. It identifies state-of-the art techniques that can solve various path planning problems including the key challenges in AMR navigation.

The rest of the paper is organized as follows: 2 provides a comprehensive review on the AMR navigation techniques which are classified into classical and heuristic approaches. 3 presents a systematic analysis on the studied literature from 2. The last section, which is 4, concludes the analysis and highlights the key take-aways from this study including future research directions.

2. Navigation techniques

Path planning techniques for AMR navigation can be categorized into classical and heuristic approaches as depicted in Fig. 2. Classical methods are typically more computationally intensive and are found to be less effective in dynamic or uncertain environments in their basic formulations. Heuristic methods, on the other hand, are relatively more robust against uncertainties in the search space. While reports on heuristic approaches have been mushrooming in the literature over the last twenty years, there are a number of classical methods that are still in common use. The following subsections will review the classical and heuristic methods as brought out in Fig. 2 which have received a great deal of attention among researchers in the past decade. For clarity purposes, the variables defined in the description of each method are only applicable locally within the section, unless otherwise stated (i.e. they are being referred to in another section).

2.1. Classical methods

2.1.1. Cell Decomposition (CD)

The essential principle behind the Cell Decomposition (CD) method is that a path between the initial and goal positions may

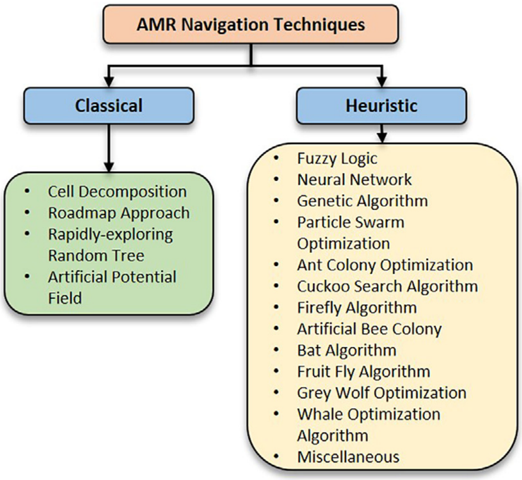
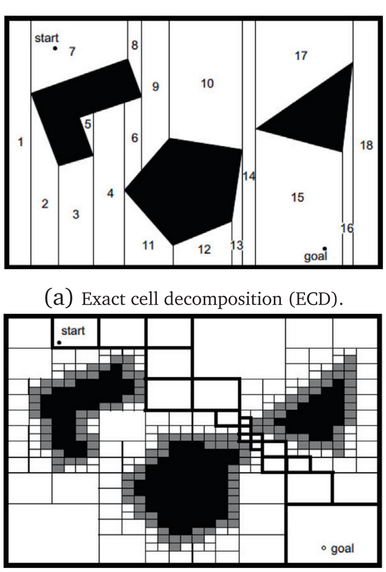


Fig. 2. Overview of classical and reactive methods commonly used for AMR navigation.

be established by separating the free space of the robot's configuration into smaller non-overlapping portions called cells. The cells can be of different shapes such as vertical strip, regular and irregular rectangular grids. To achieve path planning from the initial point to the goal position, pure cells (cells without obstacles or barriers) are considered during traversing. The start and end cells indicate the beginning and target positions, respectively. The CD has the advantage of ensuring collision avoidance between a robot with any discrete geometry and obstacles of any shape, that are not necessarily convex. Two well-known categories of the CD approach are exact cell decomposition (ECD), and approximate cell decomposition (ACD) as shown in Fig. 3.

The initial step in the ECD method is to divide the free space, which is limited by polygonal obstacles both externally and internally, into trapezoidal and triangular cells by drawing parallel line segments from each vertex of each interior polygon in the configuration space to the outside boundary. Then each cell, represented as a node, is numbered in the connectivity graph. Nodes that are adjacent in the configuration space are linked in the graph. By simply following adjacent free cells from the initial location to the goal point, a continuous path may be found from this connectivity graph [12]. Nonetheless, this method may result in unnecessary small sub-regions. Techniques that have been proposed to treat this issue include trapezoidal and boustrophedon decomposition approaches[13,14]. The Boustrophedon decomposition is an extension to the trapezoidal decomposition which reduces total number of regions by merging all the intermediate cells between two critical points into one cell [15]. In [16], a more flexible version of Boustrophedon decomposition method is proposed by taking into account both contour-following paths in cells, and the corners of cells as start and end points of cell coverage.

ACD method on the other hand employs a recursive way to keep subdividing the cells until either (i) each cell is either entirely in the free space or entirely within the obstacle zone; or (ii) it is possible to obtain an arbitrary limit resolution [17]. When a cell meets one of these requirements, it stops disintegrating. This process is also called "quadtree" decomposition as each time a cell is decomposed, it is divided into four smaller cells of the same shape [18]. The free path can then be easily determined by following the nearby, decomposed cells across free space after the decomposition process. This strategy is demonstrated in Fig. 3(b). This system has the shortcoming of not being able to update the software when new data (such as the position of an obstacle) is received, and hence fails in dynamic settings. Moreover, it often does not allow the user to include other motion restrictions, such as nonholonomic dynamics or communication constraints [19]. For high-dimensional static configuration spaces, the probabilistic cell decomposition (PCD) method is introduced in which combines the CD method with probabilistic sampling [20]. It has also been demonstrated that this method can be promising in various situations such as maze-like problems [21] and complex 3D environments [22]. Extensions to the CD method can be found in various other literature such as [23] which proposes a sensor-based CD algorithm that guarantees the complete path planning coverage of an unknown rectilinear workspace, [24] which introduces Expanded Douglas–Peucker polygonal approximation for path-planning problems with curvilinear obstacles, and [25] which designs a radial CD algorithm to further reduce the path length as well as processing time.



(a) Exact cell decomposition (ECD).

(b) Approximate cell decomposition (ACD).

Fig. 3. Cell decomposition techniques [11].

2.1.2. Roadmap Approach (RA)

Roadmap Approach (RA) is also termed as highway approach due to its nature that resembles a highway. A pre-requisite for using this approach is that the environment map must be completely known to the robot. Via this approach, multiples nodes are formed within the configuration space. If the straight line between two nodes is not obstructed by any obstacles, the nodes will be connected with a line called the edge. This approach can be classified into three different techniques, which are Visibility graph (VG), Voronoi diagram (VD), and probabilistic roadmap (PRM).

In the VG and VD methods, obstacles are considered as polygons, and their vertices are considered as nodes [26]. A network graph is then generated which may consist of multiple paths that connect the robot to the goal position. The VG generation is illustrated in Fig. 4 where the edges are represented by the blue lines while the optimal path is represented by the red line. One major downside of this method is it becomes costly and more time-consuming as the number of obstacles increases. To minimize the computation time, [27] proposes a simultaneous VG construction and path optimization which constructs the graph and searches for an optimal path at the same time. In [28], the computation time is reduced by grouping small obstacles and splitting the operating environment for parallel processing. Another drawback of the VG approach is the probability of collision tends to increase

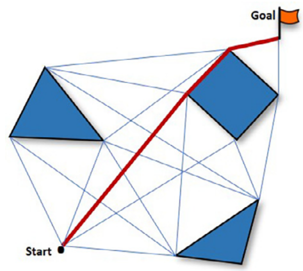


Fig. 4. Visibility graph.

as the path gets closer to the obstacles. The VD technique was initially introduced to treat this issue by virtually expanding the obstacles' size in the configuration space [29]. The concept behind the creation of the VD is to increase the distance between the robot and the obstacles to ensure the safest path is found. In this method, the space is divided into regions formed by edges which are constructed by calculating the perpendicular bisector between two adjacent sites. Fig. 5 illustrates the generation of the VD with a smoothed path between the starting point to the goal position. Nonetheless, the diagrams will be relatively harder to construct as the distance between the start and goal positions gets longer. In [30], the optimal path is found by combining the information from both the VD and sensor to construct a more reliable motion plan for the robot. The work in [31] extends the VD method by generating an Eulerian graph after the diagram construction to avoid the robot from having to perform multiple u-turns during the navigation, which is undesirable in most AMR applications. Elbanhawi et al. introduce an adaptive RA that generates waypoints, through which the robot can reach the goal point without collision with obstacles [32]. In [33,34], the VD and VG, are integrated to solve the path planning problem of Unmanned Surface Vehicles (USVs).

Unlike the VG and VD methods, the nodes via the PRM approach are be randomly generated in the configuration space. The system analyses whether the node lies in a free space, i.e. whether this configuration intersects with any obstacles. If the node is in a free space, it is added to the graph. The newly generated node is then connected to the closest node via a straight line. Next, the system checks whether the connection between the two nodes lies in a free space. If it does, the connection is added to the graph, and the process repeats for several times. As the number of nodes increases, the efficiency of the path also increases as there will be more feasible paths, but the computation time will be longer due to the increased complexity. Another drawback is its tendency to generate non-smooth paths with increased number of nodes. Fig. 6 visualizes the network graph as the number of nodes increases (from top to bottom). Nonetheless, a notable disadvantage of the classical PRM approach is that it has problems in finding paths through narrow passages due to the random sampling nodes in the free space.

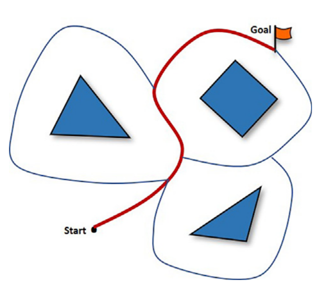


Fig. 5. Voronoi diagram.

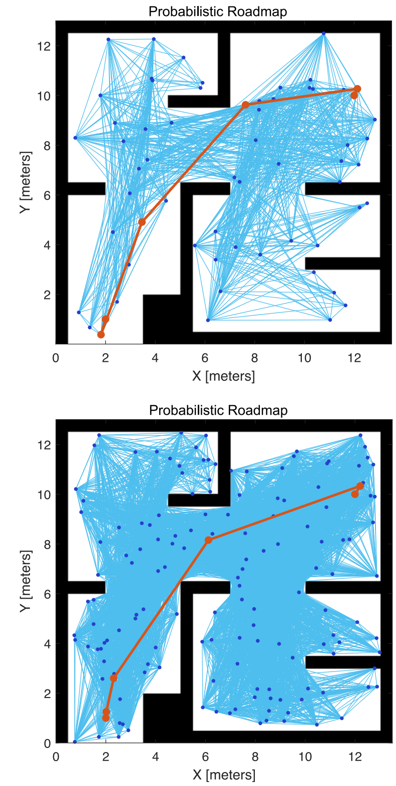


Fig. 6. Roadmaps with 60 nodes (top) and with 150 nodes (bottom) via the PRM approach. Optimal path from the start point $(2, 1)$ to the goal position, $(12, 10)$, for each case is represented by the orange line.

Within the RA and CD approaches, the shortest route between the initial and final positions are usually determined via various graph search algorithms such as the Dijkstra's algorithm (DA) [35]. DA, which stores and queries partial solutions sorted by distance from the start using a data structure, is commonly used in traffic routing systems such as Google Maps [36]. The study in [37] highlights that the A* [38] and D* Lite [39] algorithms, which are variations of DA are the most common graph search methods for finding the shortest path.

Unlike DA which emphasizes directions (favouring lower-cost paths to encourage travelling along straight lines or higher-cost paths to avoid U-turns) to explore and identify the shortest path, the A* algorithm prioritizes paths that appear to lead closer to the goal. The algorithm selects the path that minimizes the function $f(n)$ as follows:

$$f(n) = g(n) + h(n) \tag{1}$$

where $h(n)$ is the shortest distance (Euclidean, Manhattan or Chebyshev) of the current node n to the goal, and $g(n)$ is the length of the

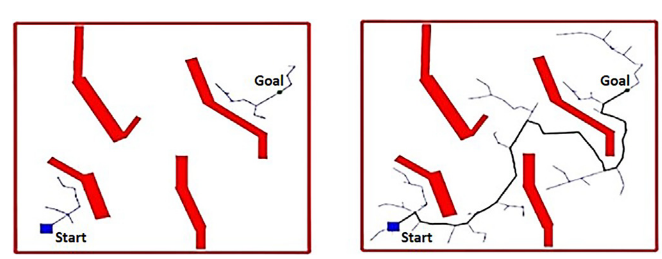


Fig. 7. RRT-Connect [45]: (Left) Two trees from start and goal points growing towards each other; (Right) The optimal path is found after both trees are connected.

path from the starting node to the goal node n. In [40], A* algorithm is used as the local path planner, and a new sampling strategy based on a virtual force field is introduced to increase the sampling density at narrow passages and reduce the redundancy of the samples in wide-open regions. Yan et al. [41] combine the RA and A* algorithm to plan a feasible path for an unmanned aerial vehicle (UAV) in a complex 3D environment.

In contrast, the D* Lite algorithm works in the opposite direction, from the objective to the beginning, and is particularly beneficial for locating the shortest path in huge and complex environments. A combination of the PRM and the D* Lite algorithm to solve the UAV path planning problems in complex 2D and 3D environments is proposed in [42]. In their work, the PRM is applied for local obstacle avoidance while the D* lite algorithm is used to plan a global trajectory.

2.1.3. Rapidly-exploring Random Tree (RRT)

Unlike DA, A*, D* Lite algorithms and their extensions which require a network graph to be constructed a priori based on RA, the Rapidly-exploring Random Tree (RRT) method does not require a graph to be specified upfront. RRTs were created to deal with a wide range of path planning issues including non-holonomic constraints of wheeled mobile robots [43]. Both RRT and PRM have similar desirable properties and were created using minimal heuristics and arbitrary inputs. This results in improved performance and consistency in the outcomes. To obtain a solution, PRM may require thousands of connections between configurations or states, whereas RRTs may not require any connections between states. This facilitates the use of RRT in non-holonomic and kinodynamic planning [44].

RRTs grow by rapidly sampling the space, then expanding from the starting point until the tree is close enough to the goal point. The tree expands to the nearest vertex of the randomly produced vertex with each iteration. This closest vertex is chosen based on a distance metric such as Euclidean and Manhattan. RRT algorithms can be used to nearly any wheeled system since they can deal with non-holonomic restrictions. Nonetheless, the classical RRT methods suffer from a number of drawbacks. One is that in order to find the parent point in the process of new point formation, the entire random tree must be walked, which is time-consuming. Another issue is the enormous number of redundant points created during path building. Thus, many researchers have proposed variations of the RRT methods to address these issues such as RRT Connect, RRT*, RRT*-Smart, Quick-RRT*, and Sensor-based random tree (SRT).

RRT Connect [45,46], also known as bi-directional RRTs, is a method that connects two RRTs, one at the starting point and the other at the goal point as depicted in Fig. 7. This method is suitable for issues with no differential restrictions. One tree is enlarged with each iteration, and the new vertex is linked to the nearest vertex of the other tree. After that, the roles of each tree are flipped, with both trees exploring the open configuration space. In [47], the bi-directional RRT is combined intelligent sample insertion heuristic for fast convergence in complex cluttered environments. Tahir et al. [48] proposed a potentially guided bi-directional RRT* to improve the convergence rate and memory utilization.

RRT* is an extension of RRT that uses triangle equality to discover the best path from a start to a goal node [49]. It has grown in popularity as a result of its ability to solve high-dimensional difficult problems [50]. The RRT*-Smart [51,52] includes intelligent sampling and path optimization techniques to produce a path that is close to the optimal at a much faster rate and at a reduced execution time. Quick-RRT* [53] on the other hand increases the number of possible parent vertices by taking into account not only a set of vertices included in a hypersphere, as in RRT*, but also their ancestry up to a user-defined parameter, resulting in lower-cost paths than those of RRT*. Many other variants have also been proposed that extend the traditional RRT which include membrane-based RRT [54], Efficient Bias-Goal Factor RRT [55], RRT*N [56], and Exploration-RRT [57].

In the SRT approach, which is considered as a sensor-based version of the RRT method, mobile robots equipped with range finders are used to explore unexpected surroundings in a randomised manner [58]. In this approach, a roadmap of the explored area is constructed with an associated Safe Region (SR) [59], and the sensors detect the Local Safe Area (LSA). Each SRT node is made up of a free configuration with a LSA linked with it. The safe region is made up by all LSAs which is a calculation of how much open space there is around the robot in a certain arrangement. The shape of the LSA is determined by the robot's sensor features (such as angular resolution). The LSA can take the form of a ball or a star. Different perception approaches can be utilized depending on the robot's sensors. Experiments have shown that the star-shaped LSR exploration approach is more accurate [58].

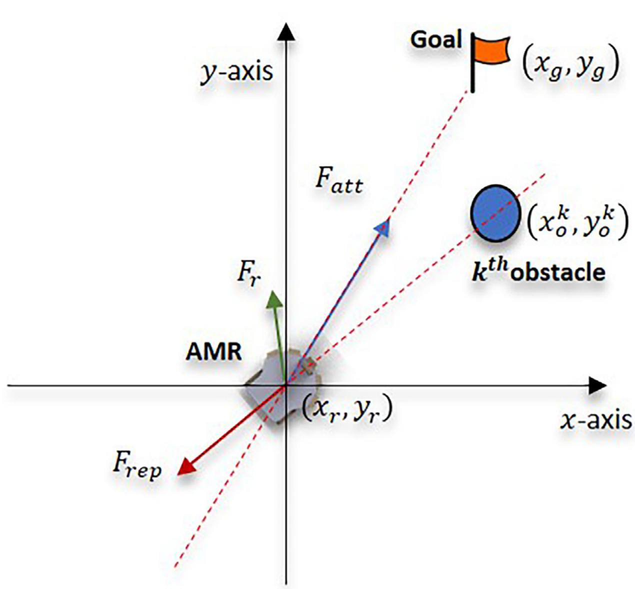


Fig. 8. Overview of APF-based path planning in the presence of an obstacle.

2.1.4. Artificial Potential Field (APF)

Artificial Potential Field (APF) method was first introduced by Khatib [60] in 1986 that worked based on the concept of attractive and repulsive forces as shown in Fig. 8. As illustrated in the figure, force fields are exerted on the AMR from the goal point at (x_g, y_g) and the obstacle at (x_o^k, y_o^k). The attraction force, $\overline{F}_{att}$, is generated

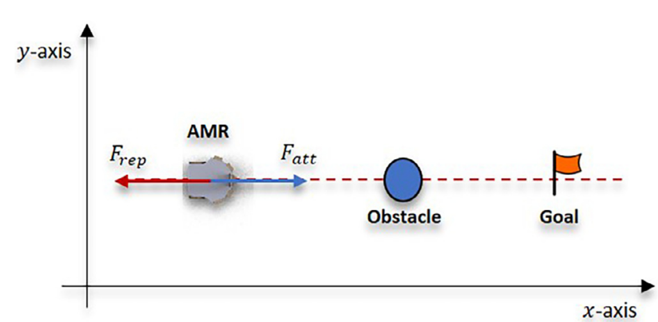


Fig. 9. A scenario showing an AMR that is trapped at a local minimum in an APF-based path planning.

from goal while the repulsive force, $\overline{F}_{rep}$, is generated from the obstacle. To simplify the path planning problem, the AMR is considered as a point mass, and $\overline{F}_{att}$ is expressed using the Gaussian function as follows

$$\overline{F}_{att} = a_g\left[1 - e^{\left(-b_g.d_g^2\right)}\right].e_g \tag{2}$$

where a_g is the maximum value of $\overline{F}_{att}$ at any instance, b_g is a constant representing the width of the distribution, and d_g is the Euclidean distance between the AMR and the goal, i.e. $d_g = \sqrt{(\Delta x_g)^2 + (\Delta y_g)^2}$ where $\Delta x_g = x_g - x_r$, and $\Delta y_g = y_g - y_r$. The parameter e_g is defined as the unit vector towards the goal point which can be expressed as

$$e_g = \frac{1}{d_g}(\Delta x_g + \Delta y_g) \tag{3}$$

$\overline{F}_{rep}$ generated by the k-th obstacle can be described by

$$\overline{F}^k_{rep} = \begin{cases} a_o\left[1 - e^{\left(-b_o.{}^kd_o^2\right)}\right].e_o^k & \text{if}\,{}^kd_o \leqslant d_d \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where a_o represents the maximum value of $\overline{F}^k_{rep}$, b_o refers to a constant representing the width of distribution, ${}^kd_o$ is the Euclidean distance between the robot and the k-th obstacle, and d_d is the influence distance around the robot. The parameter e_o^k can be computed similar to e_g, i.e.

$$e_o^k = \frac{1}{{}^kd_o}\left(\Delta^k x_o + \Delta^k y_o\right) \tag{5}$$

where $\Delta^k x_o = x_r - x_o^k, \Delta^k y_o = y_r - y_o^k$. The resultant force, $\overline{F}_r$, can then be computed via superposition theorem as follows:

$$\overline{F}_r = \overline{F}_{att} + \sum_k \overline{F}^k_{rep} \tag{6}$$

which will be the main reference for the AMR for its subsequent local path generation and motion control. Thus, APF is also regarded as a reactive navigation method since it is suitable for both offline and online path formation in unknown environments.

Despite its capabilities of simultaneous path planning and obstacle avoidance, as well as its appealing mathematical representation and conceptual simplicity, a notable downside of this approach is the local minima issue which occurs when $\overline{F}_r = 0$ before the AMR reaches the goal position. An example is shown in Fig. 9 where the robot, the goal and the centre of the obstacle are aligned which engenders a resultant force equal to zero. To address this issue, researchers have resorted to various techniques such as utilization of multi-state agents [61], exploitation of input-to-state stability property of multistable system [62], activation of a virtual escaping force [63], application of decision tree-based solution [64], and several other methods involving modifications of the resultant force [65–69].

Other inherent issues of the classical APF are the goal nonreachable with obstacles nearby (GNRON) and obstacle avoidance in a dynamic environment. A recent work in [70] adds a distance correction factor to the repulsive potential field function to solve the GNRON problem and a relative velocity detection method to avoid moving obstacles. In [71], a discrete APF method that takes into account the dynamic obstacles, path length and run time is constructed to solve the path planning problems of a differentially driven mobile robot in both static and dynamic environments. To solve the path planning problems of autonomous underwater vehicles (AUVs), Cheng et al. [72] combines the APF with a velocity synthesis algorithm to minimize the impact of the ocean current while avoiding obstacles. In [73], a predictive APF is proposed by modifying the angle limit, velocity and predictive potential to improve the feasibility and smoothness of the generated path for a high-speed USV. A recent work in [74] introduces a modified APF with a 3D vortex field for local minima elimination and obstacle avoidance of aircraft swarms in 3D dynamic environments.

2.2. Heuristic methods

2.2.1. Fuzzy Logic (FL)

The exceptional capability of a human being to reason with perception-based information and draw appropriate decisions became the inspiration behind the FL approach, which was first proposed by Zadeh in 1965 [75]. In AMR navigation, the Fuzzy Logic (FL) approach is preferred by many as it can be used not only to plan the path, but to also explicitly control the orientation and position of the robot [76,77]. Thus, the constraints on the robot's kinematic are usually taken into account during the design phase. To illustrate, consider a differential drive mobile robot with a discretized kinematic model as written below [78]:

$$x_{k+1} = x_k + \tau_s v_k \cos\theta_k \tag{7}$$
$$y_{k+1} = y_k + \tau_s v_k \sin\theta_k \tag{8}$$
$$\theta_{k+1} = \theta_k + \tau_s \omega_k \tag{9}$$

where τ_s is the sampling time, v_k is the robot's linear velocity, ω_k is the robot's angular velocity, and (x_k, y_k) and θ_k are the position and orientation of the robot respectively at time step k. Let the distance between the current position of the robot and goal position, (x_g, y_g), be written as

$$d = \sqrt{(x_g - x)^2 + (y_g - y)^2} \tag{10}$$

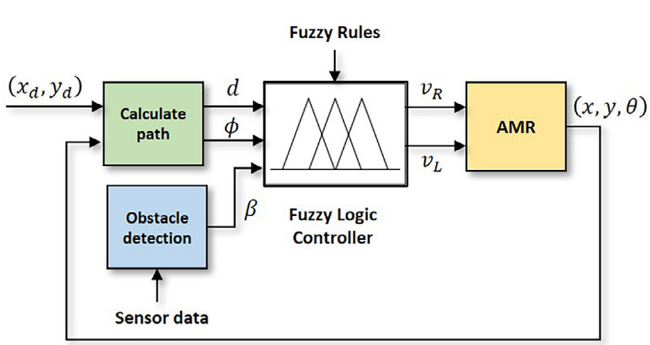


Fig. 10. An example of FL approach for AMR path planning with obstacle avoidance.

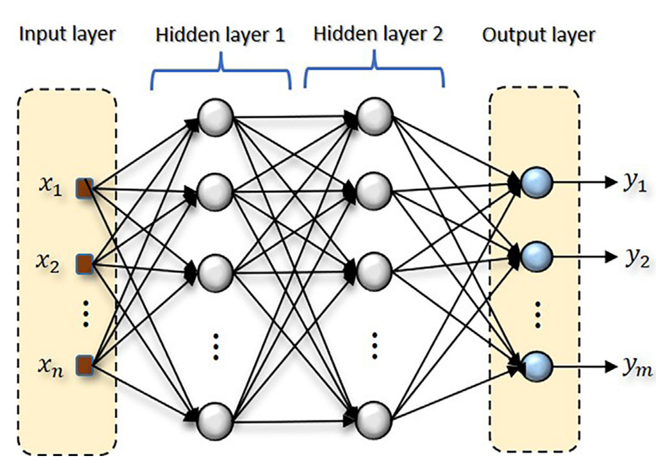


Fig. 11. An ANN architecture with n inputs, two hidden layers, and m outputs.

and the required rotation be written as $\phi = \theta_g - \theta$, where

$$\theta_g = \arctan\left(\frac{y_g - y}{x_g - x}\right). \quad (11)$$

As depicted in Fig. 10, the FL controller can be designed by setting d, ϕ, and β which is the sensor output as inputs, and the linear velocities of left and right wheels (i.e. v_L and v_R) as the outputs which relate to v and ω as follows:

$$v = \frac{v_L + v_R}{2}; \quad \omega = \frac{v_L - v_R}{2}. \quad (12)$$

FL is typically employed in AMR path planning to avoid issues like backtracking, looping, dead-end traps (concave, convex, maze, snails)[79], steering from confined passages and curved trajectories [80]. An FL controller can be designed by defining or establishing input variables, output variables, FL membership functions, rules, and defuzzification methods. The FL approach is also found to be suitable for avoiding obstacles of different shapes as presented in [81].

In [82], an adaptive neuro-fuzzy inference system (ANFIS) is used to compensate for changing dynamics within the robot such as wheel slips, resulting in a higher position accuracy. Al-Mutib et al. [83] and Dirik et al. [84] provide a vision-based mechanism with FL to boost the robot's capabilities when in motion. Techniques to optimize the FL parameters via various meta-heuristic approaches have also attracted many researchers to further improve the robot navigation. For instance, in [85], the GA is applied to optimize the membership function of input and output variables as well as the rule base of the FL controllers. To cope with parametric and nonparametric uncertainties in the robot's dynamic model, a Mamdani-type FL controller optimized using a PSO is introduced in [86]. Another tuning approach based on a wind-driven mechanism is proposed in [87] to optimize and set the antecedent and consequent parameters of the FL control parameters, which is applicable in both unknown static and dynamic environments.

In [88,89], a multi–input–multi–output FL is introduced to vary the robot's left and right wheel speeds dynamically to avoid obstacles and replan the path. Al-Jarrah et al. [90] uses a probabilistic fuzzy controller with the NN to offer a path planning approach for multiple mobile robot systems and active motion coordination amongst them.

2.2.2. Neural network (NN)

Since its inception in 1944, neural network (NN) which is also known as artificial neural network (ANN) has found a wide range of engineering applications which include feature recognition [91], predictions [92–94], and control [95,96]. An ANN is basically a system of distributed parallel processing units (neurons) organized in a graph structure. These units transmit data due to the ability to respond to external inputs in a dynamic state. Fig. 11 illustrates an example of an ANN architecture with n inputs, two hidden layers, and m outputs. This structure is also known as multilayer perceptron neural network.

In mobile robot navigation, the inputs $x_1, x_2, \ldots, x_n$ usually represent the raw or processed signals from range finders or proximity sensors such as distance from obstacles and heading angle [97–99], while the outputs, $y_1, y_2, \ldots, y_m$ represent the velocities of wheels and/or steering angle that will be applied for the robot's subsequent motion [100]. Thus, if the structure of the environment is known a priori, a global path can be generated by simulating the responses from the equipped sensors and iteratively computing the desired robot motions and orientations [101]. Navigation in unknown environments using the NN approach has also been introduced in [102] which employs a padding mean neural dynamic model to allow navigation in challenging scenarios such as the boundary and narrow pathway maps. In [103], the NN is combined with hierarchical reinforcement learning to reduce the planning and convergence time, decrease the number of path steps, and enhance the recognition and movement functions of the mobile robots. Zhu et al. [104] propose a Glasius bio-inspired NN for an autonomous underwater vehicle (AUV) which can reduce the path planning time and avoid the robots from deadlocks without imposing delays. An extension of the work to multi-AUVs is presented in [105]. Bae et al. [106] introduce Deep Q learning combined with a Convolution Neural Network (CNN) algorithm which makes it suitable for navigations of multi-robots in 2D dynamic environments.

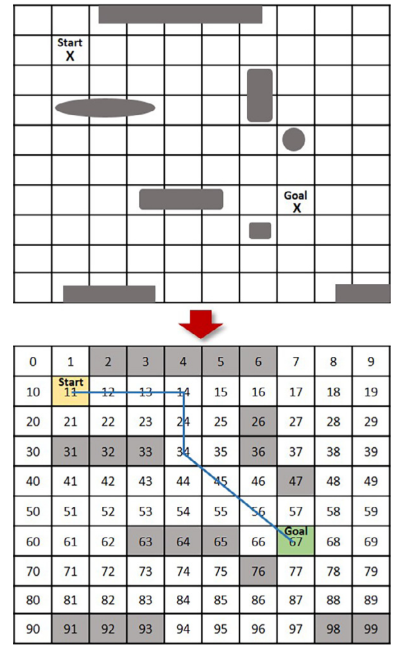


Fig. 12. Gridization of the environment. The grayed cells represent the obstacles while the white cells represent the unobstructed paths. The blue line denotes a path that can be converted into a chromosome.

Recent works have shown an increasing interest in integrating the ANN and the FL methods to form an adaptive neuro fuzzy inference system (ANFIS) to improve the robot navigation [107–110]. The ANFIS architecture has also been demonstrated to be suitable for navigations in highly cluttered environments [111,112] as well as in the presence of dynamic obstacles [113]. Aouf et al. [114] improved the ANFIS performance by introducing a Teaching–Learning-based optimization to train the ANFIS structure's setting for optimal trajectory and minimum navigation time.

2.2.3. Genetic Algorithm (GA)

Genetic Algorithm (GA) is an optimization method developed by John Holland [115] which abides the principle of genetics and natural selection. It starts with a set of solutions called generation or population, and each individual solution is termed as chromosome. In path planning, each path is represented by a chromosome which typically consists of genes that denote the start, goal and intermediate positions traversed by the robot in the search space. To construct successive generations, evolution normally begins with a population of randomly generated chromosomes, followed by fitness evaluation, selection, crossover, and mutation.

GA is among the earliest heuristic methods established for solving constrained and unconstrained optimization problems, and thus, many automated parameter tuning softwares have been developed to ease various applications. To solve the path planning problems, gridization of the environment map is usually performed prior to application of the GA. Several techniques have been proposed in the literature such as VD [116] and hexagonal cells [117], but the most common one is gridization into cells of equal size as the distance computation and obstacle representation are relatively easier. In the grid model, each cell can be represented by either a coordinate in an x-y plane, or a unique number in sequence starting from 0 as depicted in Fig. 12. The latter is found to be more widely used in the literature compared to the former. Algorithm 1 shows the pseudo-code of the standard GA path planner in a grid-based environment which usually starts with a random population generation. In [118], however, a new initialization process is proposed by creating a directed acyclic graph, which helps to search for the shortest path in a fast manner.

Algorithm 1: Pseudo-code of the standard GA path planner in a grid map

Require: Environment map and GA parameter setting.
Ensure: Best chromosome
Convert the environment into a grid map with numbered cells
Specify the starting and goal positions
Initialize arbitrarily the initial population
while no termination **do**
    Evaluation of each chromosome using fitness function
    Selection
    Crossover
    Mutation
end while

In the search of the best chromosome, the fitness function which is typically the total length of the path is applied. It can be written as [119]:

$$f = \begin{cases} \sum_{i=1}^{n-1} d(c_i, c_{i+1}) & \text{for feasible paths} \\ \sum_{i=1}^{n-1} d(c_i, c_{i+1}) + \text{penalty} & \text{for infeasible paths} \end{cases} \tag{13}$$

where $d(c_i, c_{i+1})$ is the distance between 2 adjacent cells that the robot goes through, c_i is the i-th gene of chromosome, and n is the length of the chromosome. There will be eight possible moves if we assume that the robot can move vertically, horizontally or diagonally from one cell to another. Thus, $d = 1$ if it travels vertically or horizontally, or 1.4 if it travels diagonally. If there is an obstacle in the path, a penalty which is greater than the maximum path length on the space is added. The optimal path is found by searching for a chromosome whose penalty is eliminated.

Next is the selection process which consists of three steps: (1) the objective function values of all chromosomes are found; (2) fitness values are assigned to chromosomes according to their objective function values; (3) chromosomes are selected according to fitness values and then placed in a mating pool to generate new chromosomes. In the crossover stage, features of two parent chromosomes are combined to produce two offsprings. Fig. 13 illustrates an example of a classical one-point crossover where the features after the crossover point from the parents are swapped in the offsprings. Different crossover operators have been proposed in the literature as they have a major impact on the path planning performance. In [120], the GA has been demonstrated to be applicable to large-scale static environment by using three different crossover operators applied at the same points. A new crossover technique called same-adjacency crossover has been proposed in [121] to obtain a faster convergence rate and a better search ability for global optimal path. The performance comparison between different crossover operators may be found in [122]. After the crossover process, all candidate chromosomes in the population are subjected to random mutation which is a minor alteration in a gene that is dependent on chromosome coding, and is applied uniformly to all genes of all individuals in the population with a predefined mutation rate. The mutation plays a significant role by ensuring the diversity of the population.

While the GA approach is randomized to some extent, it performs better than a random local search because it can take advantage of the historical data. A comparative study in [123] shows that the GA will result in smoother paths compared to the paths generated by the PRM approach. Nevertheless, classical GA methods are usually very slow in real-time due to the inherent shortcomings such as low convergence speed, long process in parameterizing the population size and mutation rate, and population premature [124]. Thus, many strategies have been proposed in the literature such as modifying the mutation operator [125,119], implementation of co-evaluation mechanism among the population [126] and multi-population migration GA [127], and modifying the chromosome length [128]. In [129], a multi-domain inversion technique is used to increase the number of offsprings followed by repeating the fitness evaluation stage to remove the undesirable offspring [130]. The GA approach has also been shown to be applicable in multi-robot scenario [131] as well as in a search space consisting of a moving target [132]. Nonetheless, the notable shortcoming of the GA approach is its high processing time which makes it unsuitable for real-time path planning in dynamic or unknown environments. Thus, to increase its adaptability of the GA in the mentioned scenarios, many recent studies have proposed combinations with other approaches such as ACO [133], PSO [134], FL [135] and NN [136].

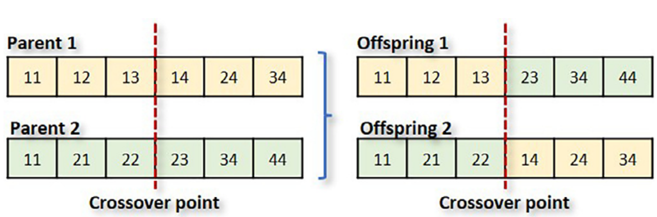


Fig. 13. An example of the classical same one-point crossover method.

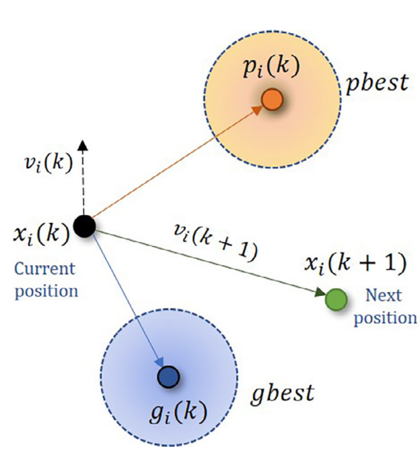


Fig. 14. Illustration on how the position of the i-th particle is updated via the classical PSO scheme.

2.2.4. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a nature-inspired meta-heuristic algorithm that mimics the social behaviour of animals like fish schools and bird flocks. It was created in 1995 by Eberhart and Kennedy [137] and has since been a rapidly expanding optimization tool for solving diverse engineering and science challenges. The PSO imitates the behaviour of a social animal, but it does not require a group leader to reach the goal. When the flocks of birds or schools of fish travel in search of food, they follow one of the individuals who is closest to the food instead of following the leaders. The PSO algorithm has been widely used to solve various problems such as mobile sinks in wireless sensor networks [138,139], and integrated process planning and scheduling [140].

In the PSO scheme, a particle is a point in the D-dimension search space, and the swarm refers to the collection of particles in a given iteration which moves around at a certain velocity. In the searching process, the position of the i-th particle, denoted by $x_i(k) = (x_{i1}(k), x_{i2}(k), \dots, x_{iD}(k))$, will be updated to reach the global optimum based on the velocity vector,

$$v_i(k) = (v_{i1}(k), v_{i2}(k), \dots, v_{iD}(k))$$

The velocity vector of the i-th particle will be updated at the k-th iteration based on

1. the historical best position of the i-th particle, known as local best particle (*pbest*), denoted as
$$p_i(k) = (p_{i1}(k), p_{i2}(k), \dots, p_{iD}(k)), \quad \text{and}$$
2. the historical best position of the entire swarm, which is also named as global best particle (*gbest*), denoted as
$$g_i(k) = (g_{i1}(k), g_{i2}(k), \dots, g_{iD}(k))$$

The updates on the velocity and position of the i-th particle can be written as follows:

$$v_i(k+1) = wv_i(k) + c_1r_1(p_i(k) - x_i(k)) + c_2r_2(g_i(k) - x_i(k)) \tag{14}$$
$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{15}$$

where $w \in [0,1]$ refers to the inertia weight that determines how much should the particle keep on with its previous velocity; c_1, c_2 are respectively the cognitive and the social parameters; and $r_1, r_2 \in [0,1]$ are two uniformly distributed random numbers. Fig. 14 illustrates on how the position of the i-th particle is updated via the classical PSO scheme which depends on three velocity components as described in the right hand side of (14). The corresponding pseudo code for the PSO is summarized in Algorithm 2 with N representing the size of the swarm.

Algorithm 2: Pseudo-code of the classical PSO

```
for i = 1,...,N do
    Randomly initialize x_i
    Randomly initialize v_i (or set to 0)
    Set the constants w, c_1, c_2, r_1, r_2
    Set p_i = x_i
end for
while no termination do
    for i = 1,...,N do
        Evaluate the fitness of particle i, f(x_i)
        if f(x_i(k)) ⩾ f(p_i(k-1)) then p_i(k) = p_i(k-1)
        else
        p_i(k) = x_i(k)
        end if
        g_i(k) = min{f(p_0(k)), f(p_1(k)), ..., f(p_N(k))}
        Update v_i according to (14)
        Update x_i according to (15)
    end for
end while
```

When utilized in the robot's path planning, the PSO is simple to construct as it only has a few adjustment parameters. Nevertheless, the classical PSO algorithm easily falls into local optima, and has poor searchability, robustness, path smoothness, and particle diversity. In [141], the path planning problem in static environments has been tackled with a new method called biogeography PSO (BPSO), which combines the biogeography-based optimization and PSO algorithms. The BPSO algorithm is used to optimize the network of paths through approximate voronoi boundary network modelling. It has also been demonstrated in [142] that the PSO can result in a better convergence than the GA approach in a dynamic environment. The work in [143] introduces a constrained multi-objective optimization by employing a fuzzy membership function to evaluate the risk degree of a path in order to avoid uncertain danger sources or obstacles. A parallel PSO scheme for global path planning for mobile robots is proposed in [144], in which three parallel PSO algorithms are paired with a communication operator to construct the viable line path that is subsequently smoothed by a cubic B-spline smoother. Compared to traditional PSOs, the parallel computing architecture has the advantage of preserving better population variety and preventing early convergence. In [145], PSO scheme is combined with an improved gravitational search algorithm to minimize the path length, energy consumption, total number of turns and arrival time for path planning of multi-robots in cluttered environments.

Li et al. reduced the probability of the classical PSO from being trapped in local optima by improving the random initialization of the particles and introducing an exponential decay for the inertia weight [146]. To attain a smooth path planning while avoiding the local trapping phenomenon that frequently occurs during the global search, the work in [147] proposes a unique multimodal delayed PSO algorithm that is combined with a Bezier curve. Another approach combining the PSO with the Bezier curve and an adaptive delayed velocity was recently introduced in [148] for the same purpose. In [149], a self-adaptive learning PSO is introduced to adaptively select the most appropriate search strategies at various optimization process stages, which can eventually increase the PSO's search ability. Zhang et al. developed a hybrid bare bones PSO to generate feasible paths by combining infeasible paths blocked by obstacles with feasible paths using improved mutation strategies of the differential evolution method [150]. A Pareto domination with collision constraints is then developed to determine the personal best position of a particle based on the collision degree of a path. Another adaptive PSO technique is introduced in [151] to reduce the path length by framing the objective function based on the distance between the robot and the goal, as well as the distance between the robot and the obsta-

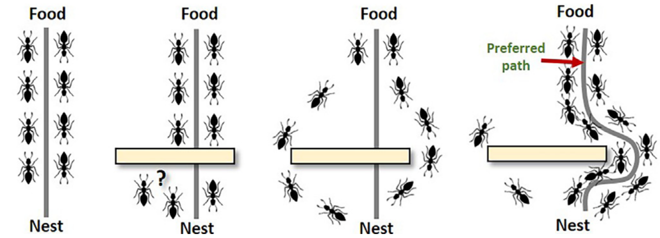


Fig. 15. Ants behavior while foraging for food.

cle. Similar to [150], the Pareto method is employed to balance conflicting constraints. In [152], an improved PSO is hybridized with a Q-learning technique to optimize path trajectories of multiple robots in a cluttered environment.

By considering the restrictions of the robot's field-of-view, relative positioning, communication, local sensing, anti-collisions issue, and kinematic limitations, Yang et al. [153] propose a constrained PSO-based collaborative searching method suitable for robotic swarms. In [154], a distributed PSO algorithm is introduced to guide swarm robots to perform target searches while taking into account energy consumption and communication limit of each robot. The PSO has also been hybridized with another heuristic method such as FFO [155], BA [156], ANFIS [157], GWO [158], and both APF and FL [159] to further improve the performance.

2.2.5. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a population-based approach proposed by Marco Dorigo in 1997 [160] which uses a form of past performance memory inspired by the real-world ant foraging behaviour. Fig. 15 illustrates the ant colony's movement to find the shortest path between the nest and the food source. While commuting between the two locations, they would leave pheromones on the ground for others to pick up and follow the trail.

Owing to its advantages, including good feedback information, strong robustness, and better distributed computing, the ACO is widely used in path planning for mobile robots. Nonetheless, due to the lack of initial pheromones and unnoticed changes in heuristic values across neighbouring grids in the search space, the traditional ACO usually takes a longer time to search, resulting in a slow convergence rate and prematurity. Thus, many researchers have introduced various strategies to improve the classical ACO mechanism. One way to increase the convergence speed is by combining the pheromone diffusion and geometric local optimization in the process of searching for the globally optimal path [161]. During the searching process, the present path pheromone diffuses in the direction of the potential field force, causing ants to seek for a larger fitness subspace and a smaller search space for the test pattern. Tao et al. present a new pheromone update rule using a FL control to change the value of pheromone and expectation heuristic factors, adjusting the evaporation rate in stages [162].

In [163], a new pheromone updating rule and a dynamic adjustment of the evaporation rate are adopted to accelerate the convergence speed and to enlarge the search space. Kumar et al. [164] introduce a hybridization scheme for humanoid robot navigation in a cluttered environment which combined the ACO with a regression controller to estimate the distance from obstacles. In [165], the bending suppression operator and the evaluation function of the classical A* algorithm are introduced to improve the heuristic information of the ACO, which can speed up the convergence and improve the smoothness of the global path. Such a method has been shown effective in complex tunnel, trough and baffle maps. To cope with narrow aisles in which the ACO method is bound to fail, Wang et al. [166] revamp the ants' walking rules and introduce a new weighted adjacency matrix to decide the walking direction. In addition, the best ant and the worst ant are introduced to help with the pheromone modification and the search process.

Other methods to mitigate the local minima or deadlock issues include adding a time-varying pheromone updating rule [167], building an unequal allocation initial pheromone to avoid the blindness search during early planning [168], and using an adaptive heuristic function based on the Euclidean distance between the ant location and the target destination [169]. To further optimize the initial path and obtain a high convergence rate, a local optimization method based on path geometry features is developed in [170]. For navigations in a complex environment, the ACO algorithm can be enhanced by redesigning the pheromone diffusion gradient formula and backtracking strategy as proposed in [171]. To navigate in a dynamic environment, Ajeil et al. [172] introduce a variation to traditional ACO that was dependent on the ant's age. The resulting algorithm was shown effective when implemented in association with the grid-based maps. In [173], the ACO is combined with a FL planner to solve the ACO drawbacks of a single evaluation factor and low path quality in path planning. Song et al. [174] introduce hybrid costs denoted as the grid weights to create a new workspace model to improve the robot navigation in a coal mine environment. Chen and Liu [175] combine the ACO and APF to increase the convergence rate of the ACO where the resulting the method has been shown to have stronger stability and environmental adaptability. In [176], the ACO is applied for unmanned surface vehicles (USVs) navigation by boosting the initial search efficiency where the initial pheromone was dispersed unevenly by creating a distance relationship between the intermediate node, the starting point, and the terminating location.

2.2.6. Cuckoo Search Algorithm (CSA)

Some cuckoos lay their fertilized eggs in the nests of another species, allowing host birds to brood and hatch the young cuckoo chicks. When the host birds realize the eggs aren't theirs, they either destroy them or abandon the nest and develop their own broods elsewhere. This scenario along with typical characteristics of Levy flights has led to the development of Cuckoo Search Algorithm (CSA) by Yang and Deb in 2009 to tackle nonlinear optimization problems [177]. The CSA relies on three basic rules; (i) each cuckoo is only allowed to lay one egg per iteration in a randomly selected nest; (ii) the best nests with high-quality eggs are passed down to the next generation; and (iii) the number of available host nests, N, is fixed, and a host bird uses a probability $p_a \in [0,1]$ to find the cuckoo's egg. In other words, the host has the option of discarding the egg or abandoning the nest, and start building a new nest by the fraction p_a (typically set to 10%) of N nests at a new location.

In the context of path planning problems, each egg in a nest is considered as a collision-free location or a solution in the search space, and a cuckoo's egg represents a new solution. The quality of a cuckoo's egg refers to the optimized path length of the robot. The goal is to use the new and potentially better solutions (or cuckoos) to replace a not-so-good solution in the nests. The following equation is used to generate a new solution for the i-th cuckoo [177],

$$x_i(k+1) = x_i(k) + \alpha \oplus levy(\lambda) \quad (16)$$

where $\alpha > 0$ (usually set to one) is the step size. The term $\oplus$ refers to entry-by-entry multiplications. This entry-wise product is comparable to those used in PSO, but because its step length is much longer in the long run, the random walk via Levy flight (i.e.

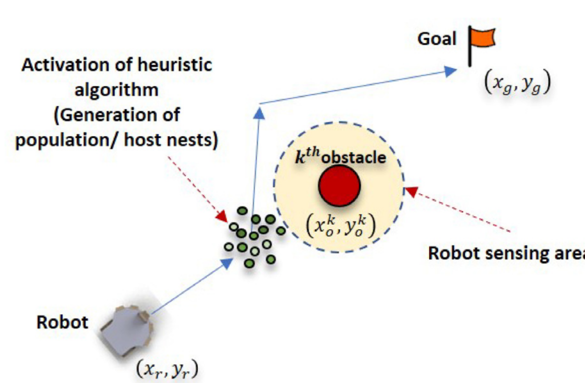


Fig. 16. Activation of the CSA to generate a local path.

$y = x^{-\lambda}, \lambda \in (1,3]$) is more efficient in traversing the search space [178]. Algorithm 3 [177] describes the pseudo code for the CSA where F_i refers to the objective or fitness function.

Algorithm 3: Pseudo-code of the CSA

```
Initialize population of host nests, x_i (i = 1, 2, ..., N)
while no termination do
    Update cuckoo x_i by Levy flights, and evaluate its fitness F_i
    Choose a nest among N nests (say, x_j), randomly
    if F_i > F_j then Replace x_j by a new solution using (16)
    end if
    A fraction (p_a) of worse nests are abandoned and new ones are built
    Keep the best solutions (or nests with quality solutions)
    Rank the nests and find the current best
end while
```

The application of CSA for AMR navigation in a partially unknown static environment is proposed in [179] with the following objective function for each nest

$$f_i(x) = \frac{C_1}{\min \|D_{N-O}\|} + C_2\|D_{N-G}\| \tag{17}$$

where $\|D_{N-O}\|$ and $\|D_{N-G}\|$ are the Euclidean distance between the nest and obstacle, and between the nest and goal respectively. In their work, the robot would initially move towards the goal, and the CSA would be activated only when it encounters an obstacle within its sensing area as depicted in Fig. 16. To avoid premature convergence which may lead to the local minima issues, the work in [180] replaced the random selection mechanism of the classical CSA with a tournament selection function. The proposed method has demonstrated a better output in terms of path time and path length when compared against the classical CSA as well as the PSO method. In [181], a smart CSA is proposed by redesigning the fitness function based on obstacle avoidance and target seeking behavior. By including the path length, safety and smoothness in the fitness function, the CSA is also able to solve the path planning problem in a dynamic environment as shown in [56]. In [182], a chaotic map for generating chaotic sequences is used to dynamically change the parameters α and p_a in the classical CSA which can eventually improve the global best searching performance. Song et al. introduce a compact CSA and a new parallel communication strategy to increase the accuracy and achieve a faster convergence in a 3D path planning problem of underwater unmanned submersibles [183].

Similar to other heuristic methods, the CSA can perform better when combined with other techniques for a more complex terrain. In [184] which focuses on navigation of multiple mobile robots in an unknown clustered environment, the CSA is employed to train the premise part and the least square estimation method was used to train the ANFIS subsequent parameters. The consistency between the simulation and experimental results is also demonstrated. Wang et al. propose a hybrid path planning strategy for an unknown 3D environment by merging the differential evolution algorithm with CSA to accelerate global convergence speed [185]. In their approach, the DE is applied to optimize the process of selecting cuckoos of the improved CSA model during the nest updating process. A combination with BA is introduced in [186] where the CS is employed to find the local best solution prior to utilization of the BA to find the global optimum. In [187,188], the GA is employed to enhance the global search ability of the CSA scheme to improve the path planning in a 3D environment.

2.2.7. Firefly Algorithm (FA)

The Firefly Algorithm (FA) was introduced in 2010 by Yang [189] which was inspired by the fireflies flashing behavior. This algorithm mimics the behavior of the firefly that produces bioluminescence; the process by which a firefly emits light to attract other fireflies and warn predators. The brightness of an individual firefly is based on the intensity of light emitted by it. Fireflies with low luminescent will be attracted by fireflies with high light intensity. Exploration and exploitation are the two crucial balancing factors in the firefly algorithm. Exploration in this algorithm is the process of determining a diverse range of solutions by randomly searching within the search space, while exploitation gives importance to the direction of search within the best available coverage using the information gathered in exploration. The FA is subject to three ideal rules, i.e. (i) All fireflies are unisex, one firefly will attract another regardless of the gender; (ii) The attractiveness is proportional to the brightness; i.e. one firefly will move towards the brighter one, and the brightest fireflies will move randomly; and (iii) The brightness of a firefly is influenced by the objective function.

The essential idea of FA is that the absolute brightness of fireflies indicates the value of the objective function, and the position of the firefly represents the solution to the problem. Both the attractiveness and brightness decrease as the distance between two fireflies increases. The brightest fireflies will move randomly. Comparing two fireflies yields the relative brightness, which is linked to attraction. Let β_{ij} and r_{ij} denote the attractiveness and distance between fireflies i and j respectively, d denote the dimension, and $x_i(k), x_j(k)$ denote the positions of fireflies i, j. Via FA, the next position of firefly i moving towards the brightest firefly j can be written as follows:

$$x_i(k+1) = x_i(k) + \beta_{ij}(r_{ij})\big(x_i(k) - x_j(k)\big) + \alpha\left(\epsilon - \frac{1}{2}\right) \tag{18}$$

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2}, \quad r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d}\big(x_i(k) - x_j(k)\big)^2} \tag{19}$$

where β_0 refers to the biggest attraction, i.e. when $r = 0$; $\alpha \in [0,1]$ is the random coefficient that controls the movement range; γ is the absorption coefficient that controls the light intensity changes and convergence speed; and $\epsilon \in [0,1]$ is a uniformly generated random number [189]. Given an objective function, $f(x), x = (x_1, x_2, \ldots, x_d)^T$, the pseudo code for FA can be stated as in Algorithm 4 [189].

Algorithm 4: Pseudo-code of the FA

```
Initialize population of fireflies, x_i (i = 1, 2, ..., N)
Light intensity I_i of x_i is determined by f(x_i)
Define γ
while no termination do
    for i = 1:n do
        for j = 1:n do
            if I_i < I_j then Move firefly i towards j
            end if
            Vary attractiveness with distance r via e^(-γr^2)
              Evaluate new solution and update light intensity
          end for
    end for
    Rank the fireflies and find the current global best
end while
```

The flashing behaviour of a firefly can be employed in the robot navigation problem to determine the best path planning when the robot is surrounded by static and dynamic obstacles. The parameters α and γ are typically designed to be adaptive to enhance the convergence speed [190,191]. Patle et al. introduce a multivariable correlation function to present the mathematical modelling for the FA, co-in-centric sphere, and robot navigation [192]. The proposed multivariable correlation function allows the robot to make appropriate decisions in complicated surroundings within a reasonable amount of time. Similar to the approach in Fig. 16, the work in [193] introduce activation of the FA to generate a local path with an objective function as in (17) when the robot encounters unanticipated obstacles while moving towards the goal. The approach has been validated by both simulations and real-time experiments for single and multiple AMRs where the percentage deviation was no greater than 5.7%. Liu et al. [194] introduced a modified FA for autonomous underwater vehicles by modifying α and γ in (18) as well as the objective function. The simulation results demonstrated an effective path planning in a 3D environment with a fast convergence speed. In [195], the objective function was formulated based on path safety, path length, and path smoothness which was related to the energy consumption. The effectiveness was demonstrated via eight realistic scenarios in path planning problems.

Compared to GA and ACO, the FA method is easier to be trapped into the local optima solutions. To address this issue, the work in [196] proposes combining the FA with GA whereby when the FA reaches the local optimal solution, the local optimal fireflies would be considered a group, and the group was subjected to the GA's selection, crossover, and mutation procedures. The genetic operations would then be used to identify the optimal firefly individual. Other methods such as adding a Sobol sequence to initialize the firefly population and adding a dynamic disturbance coefficient to enhance the global search ability have also been employed in a recent work in [197]. To treat the problems of slow convergence and low solution precision, a multi-objective FA based on archive learning s introduced in [198] which also results in shorter and smoother paths. Garip et al. combined the FA with PSO and CS to minimize the path length, which has been verified both via simulations and real-time experiments [199]. In [200], the FA is hybridized with FL to optimize the path and travel time of a UAV. The effectiveness has been validated via both simulation and real-time experiments in both static and dynamic environments.

2.2.8. Artificial Bee Colony (ABC)

A comparative study in [201] has demonstrated that the ABC method can outperform other algorithms such as GA and PSO on a large range of unconstrained test functions. The ABC algorithm, which was introduced by Karaboga in 2009, mimics the behaviour of a honey bee colony in its foraging procedure. In ABC, three types of bees, namely employed bees, onlooker bees, and scout bees, work together to find the best nectar sources in a particular area. The nectar amount of a food source refers to the quality (fitness) of the related solution, and the position of a food source reflects a feasible solution to the optimization problem.

Similar to other population-based approaches, the first step of ABC is generation of initial population which is made up of N number of D-dimensional real-valued vectors, representing the food sources. In the classical ABC method, both number of employed bees and number of onlooker bees are set to N. Let $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ denote the i-th food source position vector in the population, where each is initialized as

$$x_{ij} \leftarrow x_{min,j} + r(x_{max,j} - x_{min,j}) \tag{20}$$

with $r \in (0, 1)$ being a random number, and $x_{max,j}$ and $x_{min,j}$ being respectively the upper and lower bound constraints of the decision variable. These food sources are assigned to N employed bees whose duty is to determine a new food source, v_{ij}, by executing crossover and mutation using the following equation

$$v_{ij} \leftarrow x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \quad j \neq i \tag{21}$$

where $j \in [1, D]$ and $k \in [1, N]$ are randomly generated. The control parameter $\phi_{ij} \in [-1, 1]$ refers to the step length that reflects the bees movement. After the new food source has been generated, a greedy selection strategy is held between v_{ij} and x_{ij}, if v_{ij} has more nectar amount than x_{ij}, then x_{ij} is replaced by v_{ij} and its trial counter is reset to zero, otherwise x_{ij} remains and its trial counter increases by one.

After that, an onlooker bee selects a food position x_{ij} based on the probability

$$P_i = \frac{\mathrm{fit(i)}}{um_{j=1}^{N}\mathrm{fit(j)}} \tag{22}$$

$$\mathrm{fit(i)} = \begin{cases} \frac{1}{1+\mathrm{obj(i)}} & \mathrm{ifobj(i)} \geqslant 0 \\ 1 + \mathrm{abs(obj(i))} & \mathrm{ifobj(i)} < 0 \end{cases} \tag{23}$$

where fit(i) is the fitness value of the solution, obj(i) is the objective function value of the food position x_{ij}. Once the food positions have been selected, new food positions are then generated using (21), evaluated, and selected using the same greedy selection strategy. If a food position has not improved after a number of trials no greater than $N \times D$, the employed bees are replaced by scout bees where the solutions are also replaced by a random food position x_{ij} according to (20).

The ABC algorithm has the advantages of being simple, flexible, and having very few control parameters to tune [201]. Nonetheless, like other heuristic methods, it too suffers from low accuracy of the solution and convergence speed. In [202], the accuracy of the ABC is improved by using a balance-evolution strategy where the convergence information during the iteration is fully utilized to manipulate the exploration–exploitation accuracy and to pursue a balance between global exploration and local exploitation capabilities. The work in [203] proposes the search equation based on differential evolution into ABC's employed bees' phase to speed up its convergence and bring the global best position to guide onlooker bees' search.

Another improved ABC algorithm has been successfully implemented for multiple mobile robot navigation as presented in [204] by using elite individuals for preserving good evolution, solution sharing for a proper searching direction, and an instant update strategy for attaining the newest information of solution. In [205], the ABC is used to generate a feasible path to perform local searches that incrementally build a collision-free path connecting the start and target positions of the mobile robot. An evolutionary programming (EP) along with a set of mutation operators are then employed to refine and smoothen the feasible path, where the resulting solutions have demonstrated a significant improvement over the classical PRM and Dijkstra algorithm. A quite similar approach has been proposed in [206] with a slight modification on the criteria on which the best food position is selected to reduce the navigation time. In [207], a modified ABC is combined with the EP approach to smoothen the resulting intermediate feasible path, which has shown to be more effective compared to PSO, RRT and GA for solving the multi-robot navigation problems in unknown dynamic environments.

Nayyar et al. [208] introduce an improvement using an Arrhenius equation to balance between the exploration and exploitation process of the classical ABC for navigation in a static environment where the result has shown to be better than the PSO approach. The work in [209] focuses on improving the navigation strategy in an environment with circular-shaped static obstacles where the ABC was modified using a dynamic control limit in the exploitation phase. Liu et al. [210] propose a combination of an adaptive GA with an improved ABC algorithm to solve the path planning problems of multiple UAVs for a disaster rescue mission.

2.2.9. Bat Algorithm (BA)

Inspired by how fast bats send and receive ultrasonic signals from their prey, Bat Algorithm (BA) was formulated by Xin-She Yang in 2010 as an optimization technique to search for the global optimal solution [211]. The BA basically follows three idealized rules; (i) All bats can sense distance from the prey using the echolocation; (ii) Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min}. The wavelength λ, loudness A_0, and pulse transmission frequency $r \in [0,1]$ are varied to search for prey; (iii) The loudness A_0 varies between the smallest value A_{min} and A_0. The frequency, velocity and position of the bats are updated as follows:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{24}$$

$$v_i(k+1) = v_i(k) + (x_i(k) - x^*(k))f_i \tag{25}$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{26}$$

where f_{max} is the maximum frequency, $\beta \in [0,1]$ is a random number drawn from a uniform distribution, x^* is the current global best location (solution) based on all soultions from n bats which are randomly assigned a frequency $f_i \in [0,100]$ in the beginning. For the local search, once a solution is obtained among the current best solutions, a new solution for each bat will be generated locally using random walk:

$$x_{new} = x^* + \epsilon A(k) \tag{27}$$

where $\epsilon \in [-1,1]$ is a random number, while $A(k) =< A_i(k) >$ refers to the average loudness of all bats at time step k. The loudness A_i and the pulse emission rate γ_i are updated based on the following iterative process. When the bat finds the prey, A_i will decrease, r_i will increase, and then A_i will automatically change with any convenient value as follows:

$$A_i(k+1) = \alpha A_i(k), \quad r_i(k+1) = r_i^0(1 - e^{-\gamma k}) \tag{28}$$

where α, γ, and r_i^0 are constants. For any $\alpha \in (0,1)$ and $r_i > 0$, there exist

$$A_i(k) \to 0, \quad \text{and} \quad r_i(k+1) \to r_i^0 \quad \text{as} \quad k \to \infty. \tag{29}$$

While the BA has the capability of auto-zooming into a region where promising solutions have been found during the global search, it suffers from low convergence rate and precision during the local search. Thus, a number of recent works have proposed hybridization with other heuristic methods to treat this issue [212]. In [213] for instance, the Dijkstra algorithm is employed to find the local best solution prior to utilization of the BA to find the global optimum. Zhou et al. [214] adopt the ABC to reduce the local search time by 50% and increase the quality of the optimal solution by 14%. In [215], the BA is modified and hybridized with a PSO to balance between the exploration and exploitation processes in the path planning for static and dynamic environments. The work proposes variations of A_i and r_i during the auto-zooming process, and dynamic adjustment of α and γ during the local search. A new bat algorithm with mutation is proposed in [216] a new (BAM) is proposed to solve the path planning problem of an uninhabited combat air vehicle revolves around optimizing the flight route that was subject to different types of constraints under complicated battle field environments. Another approach is proposed in [217] where the differential evolution and B-Spline curves were employed to select the most suitable individual in the bat population and smoothen the path respectively. To achieve global optimal solutions with dynamic obstacle avoidance in the AMR path planning, the combination of BA with the dynamic window approach is able to significantly reduce the path length compared to the PSO approach and the classical BA as shown by the simulation results in [218]. In [219], an improved APF method was adopted to accelerate the convergence process of the bat's position update. The resulting algorithm has significantly increased the success rate of finding suitable path and decrease the convergence time in both 2D and 3D UAV path planning problems.

2.2.10. Fruit Fly Optimization (FFO)

Fruit Fly Optimization (FFO), which was first introduced by Pan in 2012 [220], is based on the idea that fruit flies have superior olfactory and visual senses compared to other species. The fruit fly may use its olfactory organ to detect various aromas floating in the air while also using its visual organ to locate the food and other fruit flies. The food foraging behaviour of the fruit flies is illustrated in Fig. 17. Using its osphresis, the fruit fly first investigates the direction of the food source and flies towards the spot it has smelled. Following that, the fruit fly utilizes its sensitive vision to locate the food as well as other fruit flies, and then flies in the direction it has discovered.

Similar to other population-based techniques, the location of the fruit fly swarm in FFO will be randomly initialized as described in (20). During the osphresis-based search process, the swarm generates Mospnew locations of the food source that are randomly surrounding its current location. This random searching strategy to find the new location of the food source is given by

$$x_{ij}(k+1) = x_{ij}(k) + \epsilon, \quad i = 1,2,\ldots,N; \quad j = 1,2,\ldots,D \tag{30}$$

where $\epsilon \in [-1,1]$ is a random number. For each location generated, its smell concentration judgement, denoted by S_i, is inversely proportional to the distance between the location of the $i$the fruit fly and the origin, D_i, i.e.

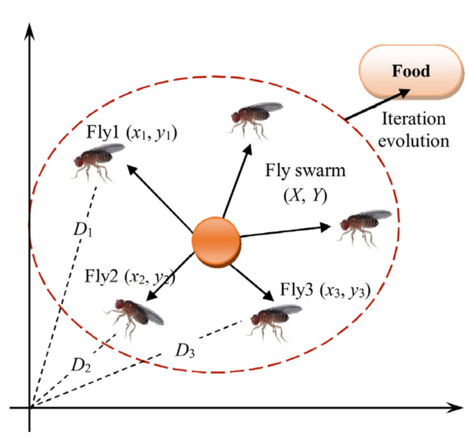


Fig. 17. Food foraging behaviour of fruit flies [221].

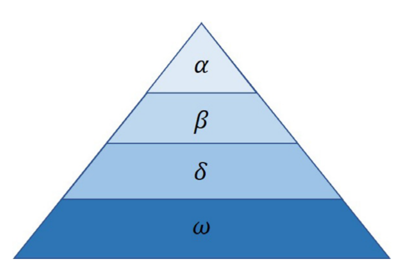


Fig. 18. Hierarchy of grey wolf (fitness decreases from top down)[227].

$$S_i = \frac{1}{D_i}, \tag{31}$$

In this case, the objective function for the candidate solution, which is also the smell concentration judgement function, simply relates to S_i as $f_i = f(S_i)$. The swarm will then perform a visual-based search which is essentially a greedy selection process to find out the best location that has the minimum smell concentration, i.e.

$$\text{index} = \arg\min(f), \quad f = (f_1, f_2, \ldots, f_N). \tag{32}$$

The best smell concentration value with their position information (i.e. based on index) will be kept, and the swarm will use the vision to fly towards that location. The osphresis-based and vision-based searching processes will be repeated until the specified stopping criterion is satisfied.

The notable advantages of the FFO algorithm are the nonlinear search process which can lead to a powerful search capability, and it only has a few parameters to optimize. Nonetheless the classical FFO has several inherent shortcomings such as low convergence speed and local optima. For instance, the S_i will be undefined when the coordinates of two points are similar, which can eventually cause errors or termination when run. To address this, many researchers resort to modifying (31) such as the work in [222,223]. In [224], the entire fruit fly swarm is divided into several sub-swarms with multi-tasks in order to expand the searching space and improve the searching ability, and the offspring competition strategy is introduced to improve the utilization degree of each calculation result and realise information exchange among different fruit fly sub-swarms. Zhang et al. employs a mutation adaptation mechanism to balance the exploitation and exploration abilities of the classical FFO, and a phase angle-based encoded strategy to expedite the convergence [221]. An extension of the work is presented in [225] which introduces a quantum-based FFO that takes advantage of the probability and uncertainty of the quantum theory to overcome the weakness in premature convergence and local optima issues in the UAV 3D path planning problems. For multiple heterogeneous UAVs scenario, a dual strategy switching is introduced in [226] to simultaneously minimize the makespan and the total mission time.

2.2.11. Grey Wolf Optimization (GWO)

Grey Wolf Optimization (GWO) was modelled after the inherent leadership structure and hunting mechanism of grey wolves by Mirjalili et al. in 2014 [227]. Similar to other heuristic methods, the GWO initializes by generation a set of random candidate solutions. The grey wolf population is divided into four categories as depicted in the hierarchy in Fig. 18. α is considered as the fittest solution, followed by β and δ. The rest of the candidate solutions are represented by ω. During hunting, the first three best wolves, i.e. α, β, and γ, lead other wolves towards promising zones of the search space. The remaining wolves, i.e. ω, update their positions as follows:

$$D = |Cx_p(k) - x_i(k)| \tag{33}$$

$$x_i(k+1) = x_p(k) - A \cdot D \tag{34}$$

where D is the Euclidean distance between the individual and the prey, $x_p(k)$ is the prey's current location, $x_i(k+1)$ is the ith wolf position update with $i = 1, 2, \ldots, N$, and N being the population size. A and C are coefficient vectors which can be expressed as

$$A = 2a \cdot r_1 - a; \quad C = 2r_2; \tag{35}$$

where $r_1, r_2 \in [0, 1]$ being a random number, and a being the convergence factor that decreases from 2 to 0 over the course of iterations. Exploration and exploitation are guaranteed by the adaptive values of a and A. Half of the iterations are devoted to exploitation ($A < 1$) while the other half are dedicated to exploration ($A \geqslant 1$). The grey wolves finish the hunt by attacking the prey when it is no longer moving.

During the optimization, the ω wolves update their positions encircling α, β and δ concurrently as follows:

$$D_\alpha = |C_1 \cdot x_\alpha(k) - x_i(k)|, \quad D_\beta = |C_2 \cdot x_\beta(k) - x_i(k)|, \tag{36}$$

$$D_\delta = |C_3 \cdot x_\delta(k) - x_i(k)| \tag{37}$$

$$x_{i\alpha} = x_\alpha - A_1 \cdot D_\alpha \quad x_{i\beta} = x_\beta - A_2 \cdot D_\beta \quad x_{i\delta} = x_\delta -_3 \cdot D_\delta \tag{38}$$

$$x_i(k+1) = \frac{x_{i\alpha} + x_{i\beta} + x_{i\delta}}{3}. \tag{39}$$

where A_1, A_2 and A_3 are adaptive vectors, $x_{i\alpha}, x_{i\beta}$ and $x_{i\delta}$ are respectively the current position vectors of the α, β and δ wolves, and $x_i(k+1)$ is the position update for the ith ω wolf. Fig. 19 illustrates on how a search agent, i.e. ω wolf, updates its position based on the locations of α, β and δ wolves in a 2D search space.

The GWO was firstly applied in a UAV path planning problem by Zhang et al. [228] in 2016. The simulation results show good performances in terms of quality, speed, and stability of final solutions in a 2D search space. A comparison against CSA, BA and ABC also demonstrates that the GWO is better at the exploration stage. In [229], both the path distance and smoothness are included in the fitness function before applying the GWO for navigation in 2D static environments. Application of the GWO for multi-UAV path planning problem in a 3D space is presented in [230] which shows a better performance compared to PSO, BA and WOA approaches. The work in [231] proposes an improvement for the UAV path planning by hybridizing the GWO with a FFO algorithm whereby the latter is used to perform a local optimization following the initial path is generated by the GWO. Another improvement tech-

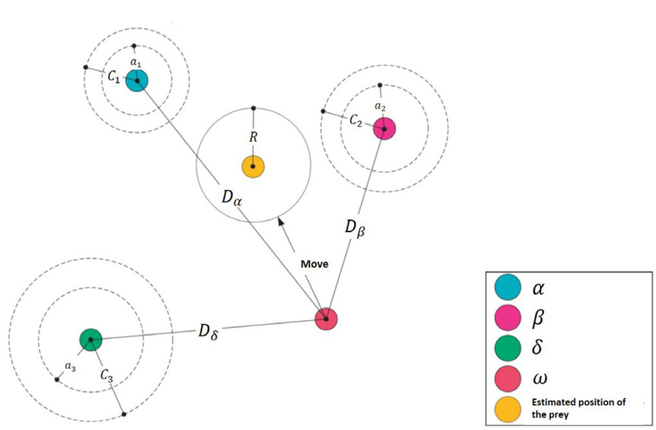


Fig. 19. Illustration on how the ω wolf updates its position after the prey is encircled [227].

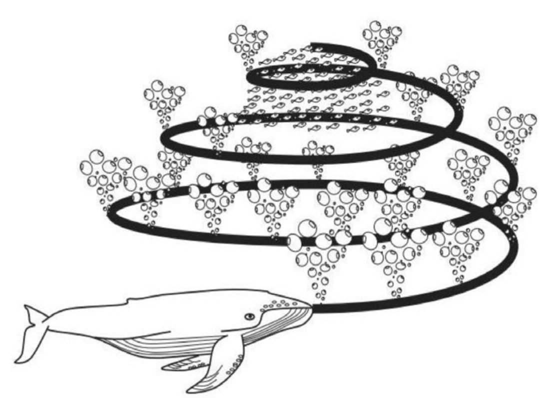


Fig. 20. Spiral bubble-net behavior of humpback whales [238].

nique is proposed by Qu et al. where the GWO is simplified and then combined with a modified symbiotic organisms algorithm to accelerate the convergence rate and retain the exploration ability of the population UAV [232]. A cubic B-spline curve is also used to smooth the generated UAV route. In [233], another new parameter is added to the convergence factor a in (35) to dynamically adjust its speed and effectively coordinate both global and local searching processes. The simulation results show that the modified GWO outperforms the classical GWO in terms of convergence precision, speed, and stability, and that it can be used to optimize a complicated 3D path planning.

In [234], the convergence speed is increased by modifying the parameter a into a nonlinear convergence factor based on a Gaussian distribution curve, and including a dynamically changing weights for $x_{i\alpha}, x_{i\beta}$ and $x_{i\delta}$ in the position update of the ω wolves in (39). Another approach to escalate the convergence speed was proposed in [235] where the ω wolves are treated equally as those of δ wolves in the exploration process. Liu and Li introduce a population initialization strategy based on logistic mapping along with an adaptive adjustment strategy for the parameter a to solve the path planning problem in a dynamic environment [236]. In [237], apart from applying the GWO to generate the global path, the GWO is also employed to tune the FL controller to further improve the trajectory of nonholonomic wheeled mobile robots.

2.2.12. Whale Optimization Algorithm (WOA)

Whale Optimization Algorithm (WOA) was introduced by Mirjalili and Lewis in 2016 to solve optimization problems based on the social behaviour of humpback whales [238]. The most interesting thing about the humpback whales is their special hunting strategy which is called bubble-net feeding. They hunt schools of krills or small fishes close to the surface by creating distinctive bubbles along a spiral path as illustrated in Fig. 20, which resembles the GWO approach of encircling the prey prior to attacking. The only difference is the use of the spiral path in place of the α, β and δ populations to imitate the bubble-net assault behaviour.

In the WOA algorithm, the whales will swim around the prey to hunt for food once the position information of the prey is available. Otherwise, the current individual whale is deemed to be the best candidate solution, and the remaining whales regulate their locations based on the best solution's position information. Let x^* denote the position vector of the best solution obtained so far, and x_i denote the current position vector of the ith whale. The position update of the ith whale can be expressed as:

$$D = |Cx^*(k) - x_i(k)| \tag{40}$$

$$x_i(k+1) = x^*(k) - A \cdot D \tag{41}$$

where A and C are coefficient vectors similar to (35) but with $r_1 = r_2$. When the whales search for the prey in the exploration phase, the same approach in (40)–(41) is used with $|A| > 1$ to force search agent to move far away from a reference whale, and $x^* = x_{rand}$ which refers to a random position vector (a random whale) selected from the current population.

During the bubble-net attack, which is regarded as the exploitation phase, the spiral position updating strategy is employed. This method first calculates the distance between the ith whale at x_i and the prey at x^*, i.e. $D = |x^*(k) - x_i(k)|$. A spiral equation is then created between the two to mimic the helix-shaped movement of the whales as follows:

$$x_i(k+1) = D \cdot e^{bl} \cdot \cos(2\pi l) + x^*(k) \tag{42}$$

where b is a constant defining the shape of the logarithmic spiral, and $l \in [-1, 1]$ is a random number.

Dao et al. is among the first to apply the WOA to solve the AMR path planning problems with path length and path smoothness as

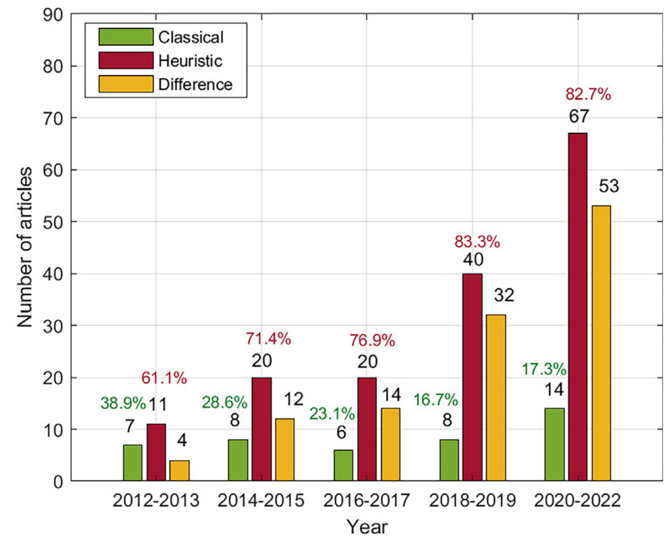


Fig. 21. Total number of articles across publication years since the past decade. The percentage values indicate the proportion with respect to all articles published within the specified period.

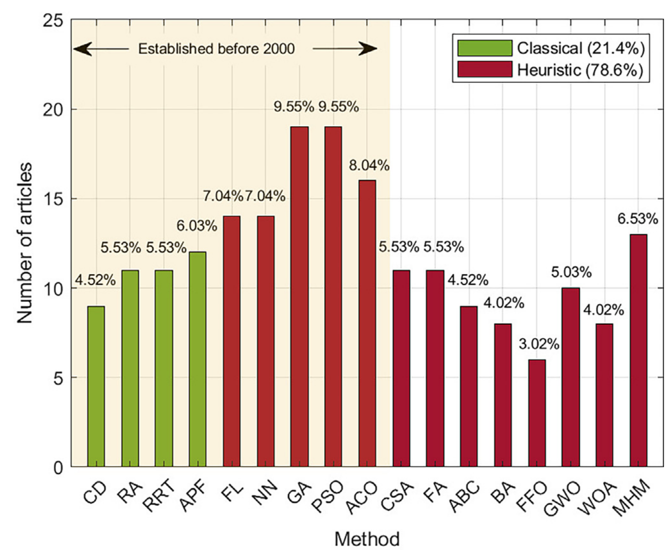


Fig. 22. Number of articles for each method. The percentage values represent the contributions to the total number.

the criteria in static environments [239]. A comparative study is conducted in [240] showing that the WOA outperforms other methods such as ABC, BA, CSA, and PSO in several 2D path planning problems due to the balanced exploration and exploitation processes in the WOA which can reduce the probability of falling into a local optimum. The WOA approach has also demonstrated superiority over GWO for a 3D path planning problem of an AUV where fuel consumption and execution time are typically the major concerns [241]. While the WOA can avoid the local optima and reach a global optimal solution, it suffers from low convergence precision and rate. In [242], an adaptive chaos-Gaussian switching solving strategy and a coordinated decision-making mechanism are introduced to enhance the global search ability of the classical WOA, which result in higher accuracy and convergence rate in 2D static environments. Liu et al. combine the WOA with an inverse initial coding optimization with Levy flight to maximally avoid the local optima during the navigation [243]. A quite similar approach can be found in [244] where apart from the Levy flight, a chaotic map-

Table 1
A summary of strengths/advantages and inherent shortcoming/challenges among classical and heuristic methods for AMR navigation.

Method	Strengths/Advantages	Inherent shortcomings/Challenges
CD	Collision avoidance between a robot with any discrete geometry and obstacles of any shape is guaranteed	Not suitable for dynamic settings particularly in high-dimensional space; does not allow inclusion of motion or communication constraints
RA	Generates path in a short time; suitable for reactive navigations	Not suitable for a large number of obstacles situated very close to each other; tend to generate non-smooth paths
RRT	Suitable for wheeled systems, can deal with non-holonomic restrictions	Time consuming, enormous number of redundant points created during path building
APF	Suitable for reactive navigations, easy implementation on hardware	Does not perform well in environments with narrow passages, GNRON, and dynamic obstacles; Easily trapped in local minima
FL	Suitable for local path planning and reactive navigations; The commands for speed and rotation are included in the design	Heavily relies on the equipped sensors. Does not usually focus on shortest path/time; multiobjective optimization is not straightforward
NN	Suitable for local path planning and reactive navigations; The commands for speed and rotation can be included in the design	Heavily relies on the equipped sensors; may require memory to improve performance
GA	Easier to generate smooth path; among the earliest heuristic methods established for optimization problems; availability of many automated parameter tuning softwares	High processing time, low convergence speed,not suitable for unknown environments and real-time path planning in dynamic environments
PSO	Simple to construct; simple concept; few control parameters	Poor searchability, robustness, and smoothness; low convergence accuracy; prone to local optima particularly in cluttered environments
ACO	Good feedback information; strong robustness; better distributed computing	Low convergence rate and prematurity due to long search process
CSA	Fewer parameters to tune compared to GA and PSO	Heavily relies on random movement to find new solutions (new host nests to lay), low convergence rate
FA	Low computation cost; flexible; easier to combine with other heuristic methods	Low exploration capability as the search is only in one direction, prone to local optima
ABC	Simple and flexible, very few control parameters, minimal processing time; better convergence speed particularly in solving multi-objective optimization problems	The demand for additional fitness tests with the inclusion of new parameters; if the nectar amount of the new food source is greater than the prior source, it can lose information.
BA	Automatic zooming into a region with promising solutions, and automatic switch from explorative moves to local intensive exploitation.	Low convergence rate and precision during the local search
FFO	Simple mechanism; fewer control parameters	No communications between search agents, low convergence speed at later stage
GWO	Better in exploration compared to CSA, BA, and ABC in a 2D search space	Low convergence speed; low solution accuracy
WOA	Ability to balance exploration and exploitation; better than GWO in a 3D path planning problem	Low convergence precision and rate
DE	Simpler implementation	Poor at exploitation; prone to local optima
BFO	Robust against the area of the workspace and nonlinearity	Stuck in the local minima whenever non-circular obstacles are encountered
SFLO	A higher success rate to achieve optimal solution in uncertain environments	Premature and slow convergence; prone to local optima particularly in multi-objective optimization problems
GHO	Can obtain shorter path length and time for navigations in dynamic environments compared to PSO, RRT, GA and NN	Low convergence rate; lack of randomness; high probability of falling into local optima
SMO	Maintains the balance between convergence speed, decision-making ability and accuracy	Prone to local optima particularly when solving high-dimension problems
SCA	Suitable for dynamic environments with multi-robots when combined with kidney-inspired algorithm	Unbalanced exploration and exploitation phases
AFS	Large tolerance of parameter setting; insensitivity to initial values	Some parameters need to be self-tuned in due time to improve the convergence accuracy
BSO	Continuous convergence and divergence process in the solution space	Premature convergence and high computational complexity

ping is additionally introduced to disturb the solutions of each generation which can indirectly enhance the diversity of solutions. Hybridizations of the WOA with FL [245] and WOA with GWO [246] have also been introduced to improve the path planning and control over multiple mobile robots in static and dynamic environments.

2.2.13. Miscellaneous heuristic methods (MHM)

Other miscellaneous heuristic methods that have been employed to solve the path planning problems in the past decade include Differential Evolution (DE) [247]248249, Bacterial Foraging Optimization (BFO) [250], Harmony Search Algorithm (HSA) [251], Shuffled Frog Leaping Algorithm (SFLA) [252], Grasshopper Optimization (GHO) [253], Slime Mould Optimization (SMO) [254], Sine–Cosine Algorithm (SCA)[255], Artificial Fish Swarm (AFS) [256]257, and Brain Storm Optimization (BSO) [258,259].

3. A systematic analysis

Based on the studied literature in the previous section, the total numbers of articles on AMR navigation using classical and heuristic approaches across publication years in the last decade is visualized in Fig. 21. Evidently, the number of articles on heuristic approach increases more rapidly than the classical approach, and the difference between them escalates almost exponentially. This does not only imply that the heuristic approach is more effective in solving various path planning problems, it also signals that there are many rooms for improvements within the heuristic approach that interest researchers to investigate further. Fig. 22 breaks down the number of articles according to the actual navigation technique where the green and maroon bars represent the classical and heuristic methods respectively, and the percentage values represent the contributions to the total number. Comparing the first nine methods from the left, which are among the earliest navigation techniques introduced before the year 2000, GA and PSO are found to be the most popular techniques. One reason might be due to availability of many automated parameter tuning software that does not require users to have extensive knowledge on its underlying model [260], thus allowing more advanced studies to be conducted. Within the classical approach, the APF, RRT and RA appear to be competitive in terms of popularity due to their suitability for reactive navigations and ease of implementation on hardware compared to CD.

The strengths/advantages and inherent shortcoming/challenges among classical and heuristic AMR path planning methods are

Table 2
Analysis of the reviewed AMR navigation techniques based on (i) environment, i.e. static (SE) or dynamic (DE); (ii) evaluation type which is either by simulation (SI) or real-time experiment (RT); (iii) search space, i.e. 2D or 3D; (iv) objective function, i.e. single-objective (SO) or multi-objective (MO); (v) consideration of the robot's dynamic constraint (DC); and (vi) application to swarm or multi-robot (SM).

			Environment		Evaluation		Space		Objective		Others	
	Hybrid	**Article**	**SE**	**DE**	**SI**	**RT**	**2D**	**3D**	**SO**	**MO**	**DC**	**SM**
CD		[13,17,16,25]	✓		✓		✓			✓		
	RRT	[14]	✓	✓	✓		✓			✓		
		[15,24]	✓		✓		✓		✓			
		[23]	✓		✓	✓	✓		✓			
		[22]	✓		✓			✓				
RA		[27,30,31,37,32,34]	✓		✓		✓		✓			
		[28]	✓			✓	✓		✓			
		[33]	✓		✓		✓			✓		
		[40]	✓		✓	✓		✓	✓			
		[41]	✓			✓		✓	✓			
		[42]	✓		✓		✓	✓	✓			
RRT		[46,51,52,53,54,59]	✓		✓		✓		✓			
		[47,48]	✓		✓		✓	✓	✓			
		[55]		✓	✓		✓			✓		
		[56]	✓	✓	✓		✓		✓			
		[57]	✓		✓		✓			✓		
APF		[62]	✓		✓	✓	✓			✓	✓	
		[63]	✓		✓		✓			✓		
		[64,67,69]	✓		✓		✓		✓			
		[66,68]	✓		✓	✓	✓		✓			
		[70,71]	✓	✓	✓	✓	✓			✓		
		[72]	✓		✓		✓		✓			
		[73]	✓		✓		✓			✓	✓	
		[74]	✓		✓	✓		✓				✓
FL		[77]	✓		✓	✓	✓		✓			
		[76,82]	✓		✓	✓	✓		✓		✓	
	GA	[89]	✓		✓	✓	✓		✓		✓	
		[84]	✓			✓	✓		✓			
	GA	[85]	✓		✓		✓		✓			
	PSO	[86]	✓		✓		✓		✓		✓	
		[87]	✓	✓	✓	✓	✓		✓		✓	
	NN	[90]	✓	✓	✓		✓		✓			
		[88]	✓		✓		✓		✓		✓	
		[77]	✓		✓	✓	✓		✓			
		[81]	✓		✓		✓		✓			
		[79]	✓	✓	✓		✓		✓		✓	
NN		[100,107]	✓		✓	✓	✓		✓			
		[102]	✓	✓	✓		✓		✓			
		[103]	✓	✓	✓		✓			✓		
		[104,108,110]	✓		✓		✓		✓			
		[105]	✓		✓		✓		✓			✓
		[106]	✓	✓	✓		✓		✓			✓
	ACO	[109]	✓		✓		✓		✓			
		[111]	✓		✓	✓	✓		✓			✓
		[112]	✓		✓	✓	✓			✓		✓
		[113]	✓	✓	✓	✓	✓		✓			
		[114]	✓		✓	✓	✓			✓		
GA	RA	[116]	✓		✓			✓		✓		
		[117,118]	✓		✓		✓			✓		
		[119]		✓	✓		✓		✓			
		[120,121]	✓		✓		✓		✓			
		[122,127]	✓		✓		✓			✓		
		[125]	✓	✓	✓	✓	✓		✓			
		[126]	✓	✓	✓		✓			✓		✓
		[128]	✓	✓	✓	✓	✓			✓		
		[129]	✓		✓	✓	✓		✓			
		[130]	✓		✓	✓	✓			✓		✓
		[131]	✓		✓		✓			✓		✓
		[132]	✓	✓	✓	✓	✓		✓			✓
	ACO	[133]	✓		✓	✓	✓		✓			
	PSO	[134]	✓		✓			✓		✓		✓
	FL	[135]	✓		✓	✓	✓			✓	✓	
	FL	[136]	✓		✓	✓	✓		✓		✓	
PSO	RA	[141]	✓		✓		✓		✓			
		[142]	✓	✓	✓		✓		✓			
		[143]	✓	✓	✓		✓			✓		
		[144]	✓			✓	✓			✓		
		[145,152]	✓		✓	✓	✓			✓		✓

(continued on next page)

Table 2 (*continued*)

	Hybrid	Article	Environment		Evaluation		Space		Objective		Others	
			SE	DE	SI	RT	2D	3D	SO	MO	DC	SM
		[146]	✓		✓		✓			✓	✓	
		[147,148,150,151]	✓		✓		✓			✓		
		[149]	✓		✓	✓	✓			✓		
		[153]	✓		✓		✓			✓	✓	✓
		[154]	✓		✓		✓			✓		✓
	FFA	[155]	✓		✓		✓			✓		✓
	BA	[156]	✓	✓	✓		✓			✓		
	NN, FL	[157]	✓		✓		✓			✓		
	GWO	[158]	✓		✓		✓		✓			
ACO		[161,163,170,171]	✓		✓		✓		✓			
	FL	[162]	✓		✓		✓		✓			
		[164]	✓		✓	✓	✓			✓		
		[165,166,167,168]										
		..[169,173,174,176]	✓		✓		✓			✓		
	APF	[172]	✓	✓	✓		✓			✓		
	FL	[175]	✓		✓	✓	✓			✓		
CSA		[179,181]	✓		✓	✓	✓			✓		
		[180]	✓		✓		✓			✓		
		[56]	✓	✓	✓					✓		
		[182]	✓		✓		✓		✓			
		[183]	✓		✓			✓		✓		
	FL	[184]	✓		✓	✓	✓		✓			✓
	DE	[185]	✓		✓			✓		✓		
	BA	[261]	✓		✓		✓		✓			
	GA	[187]	✓		✓			✓		✓		
	GA	[188]	✓		✓			✓		✓		
FA		[190,191,197]	✓		✓		✓		✓			
		[192,195]	✓		✓		✓			✓		
		[193]	✓	✓	✓	✓	✓			✓		
		[194]	✓		✓			✓		✓		
	GA	[196]	✓		✓		✓		✓			
	PSO,CS	[199]	✓		✓	✓	✓		✓			
	FL	[200]	✓	✓	✓	✓				✓		
ABC		[202,206]	✓		✓		✓			✓		
		[203,208,209]	✓		✓		✓		✓			
		[204]	✓	✓	✓		✓			✓		✓
		[205]	✓		✓	✓	✓			✓		
		[207]	✓	✓	✓		✓			✓		
	GA	[210]	✓		✓			✓		✓		✓
BA		[213]	✓	✓	✓		✓		✓		✓	
		[214]	✓		✓			✓		✓		
	PSO	[215]	✓	✓	✓		✓			✓		
		[216]	✓		✓			✓		✓		
	DE	[217]	✓		✓			✓		✓		
		[218]	✓	✓	✓		✓			✓	✓	
	APF	[219]	✓		✓		✓	✓		✓		
FFO		[222,221,225]	✓		✓			✓		✓		
		[223,224,226]	✓	✓	✓			✓		✓		✓
GWO		[228,229]	✓		✓		✓			✓		
		[230]	✓		✓			✓		✓		✓
		[231]		✓	✓		✓	✓		✓		
		[232]		✓	✓		✓			✓		
		[233]	✓		✓			✓		✓		
		[234]	✓		✓		✓		✓			
		[235]	✓		✓		✓			✓		✓
		[236]	✓	✓	✓		✓		✓			
	FL	[237]	✓			✓	✓			✓	✓	✓
WOA		[239]	✓		✓		✓			✓		
		[240]	✓		✓		✓	✓	✓			
		[241,242]	✓		✓			✓		✓		
		[243]	✓		✓		✓		✓			
		[244]	✓		✓		✓			✓		
		[245]	✓	✓	✓		✓			✓		✓
	GWO	[246]	✓	✓	✓		✓			✓		✓
MHM		DE[247]	✓		✓		✓			✓	✓	
		DE[248],BSO[258]	✓		✓		✓		✓			
		DE[249]	✓		✓		✓	✓		✓		
		BFO[250],HSA[251]	✓	✓	✓		✓			✓		
		SFLA[252], SM[254],										
		. . .AFS[257],BSO[259]	✓		✓		✓			✓		

Table 2 (*continued*)

Hybrid	Article	Environment		Evaluation		Space		Objective		Others	
		SE	DE	SI	RT	2D	3D	SO	MO	DC	SM
	GHO[253]		✓	✓		✓			✓		
	SCA[255]	✓	✓	✓	✓	✓			✓		✓
	AFS[256]	✓			✓	✓		✓			

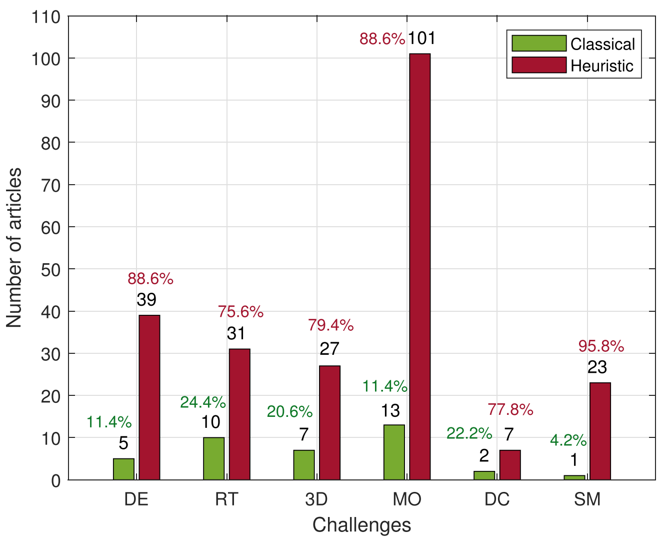


Fig. 23. A comparison between the number of articles published with classical and heuristic methods across challenges. The percentage values indicate the corresponding contributions to the total number for each category.

summarized in Table 1. In general, classical methods are mainly employed for navigation in a known environment as they typically require the initial information of the working space. Numerous researchers have worked to identify flaws in the classical approach and created a number of novel strategies, but these workarounds do not perform well in challenging scenarios such as navigation in a cluttered or dynamic environment. In contrast, the heuristic methods particularly the population-based algorithms offer a more robust technique that can deal with a higher level of environment uncertainties as they are able to generate a set of temporary paths at each iteration that will get them one algorithmic step closer to the final destination. This also indicates that the heuristic approach is able to enhance the autonomy level of the AMRs due to their efficient decision making while navigating towards the goal. Nevertheless, nearly all of them share several common challenges such as the balancing strategy between the exploration and exploitation phases to avoid local minima issues, premature convergence and low accuracy.

In order to analyse further, the scope of each method based on the reviewed articles is rigorously judged in Table 2 in terms of the (i) environment, i.e. static (SE) or dynamic (DE); (ii) evaluation type which is either by simulation (SI) or real-time experiment (RT); (iii) search space, i.e. 2D or 3D; (iv) objective function, i.e. single-objective (SO) or multi-objective (MO); (v) consideration of the robot's dynamic constraint (DC); and (vi) application to swarm or multi-robot (SM). A quick scan over the comparison made shows that over 90% of the published work are focused on SE, SI, 2D and SO which are relatively less onerous than their counterparts. Hybridization of algorithms is also more common within the heuristic approach compared to the classical method due to their mathematical formulations that allow fusion of different search techniques particularly in the exploration and exploitation stages.

AMR navigations are undeniably more challenging in practice as the surroundings are most of the time uncertain particularly in human-robot co-existence environments, underwater and in the air. Plus, the complexity increases when the robot's motion limitations are taken into account such as energy consumption and dynamic constraint. These identified key challenges are highlighted in 2 by the greyed columns. To observe the trends, a comparison between the number of articles using classical and heuristic methods focusing on each challenge is visualized in Fig. 23. The breakdown of the numbers according to the actual navigation technique for each setting is depicted in Fig. 24. From the Fig. 23, it can be seen that at least 75% of the published work have employed the heuristic methods, implying that they are more effective in solving those challenges. Another striking observation is the number of articles focusing on the MO navigation problems where the heuristic approach is found to have attracted a lot more attention from researchers compared to the classical approach. As seen in the middle plot of Fig. 23, this number is hugely contributed by the population-based heuristic techniques which allow integration of multiple objectives in a single cost function. The most common MO problem for a 2D search space is optimization of path length, path safety and path smoothness, which can have a trade-off between each other. For a 3D space, such as navigations of AUV and UAV particularly those that are solar-powered, the execution time, turning control, and fuel consumption are three other constraints that are typically incorporated in the cost function apart from the path safety [262].

The applicability of the heuristic approach for the MO problem has also led to the increase in number of articles under the 3D category in Fig. 23. In this regard, FFO, CSA, BA, GWO and WOA which were introduced much later than the GA and PSO are among the highest contributors as illustrated in the middle-left plot of Fig. 23. This is due to the nature of the algorithms themselves which have fewer tuning parameters that can ease the computations in a higher dimensional space.

Another superiority of the heuristic approach is its suitability to be applied to navigation in a DE which is defined as an environment in the presence of dynamic obstacles and/or a dynamic goal. In a DE setting, the algorithms must adapt to any unexpected changes such as the advent of new moving obstacles in the preplanned path or when the goal or target is continuously moving. In this regard, the heuristic approach has contributed to 88.6% of the total number of articles published as illustrated in Fig. 23. As can be seen on the top left plot of Fig. 24, the GA and PSO have contributed the highest number of articles, followed closely by NN, BA, MHM, FL, FFO and GWO. The classical methods, on the other hand, are not performing well in such a setting due to their basic formulations that typically require a full knowledge of the environment. The path planning problem in a DE setting becomes more critical to solve when both obstacles and targets are moving [263], or when the target can only be sensed based on the received signal strength that varies with time such as radio frequency [264] or audio signals [265,266]. Solving such a problem is of paramount importance in emergency response tasks such as search and rescue missions in uncertain disaster environments [267].

With regard to RT evaluation, high processing power, speed, and large memory are typically required in order to implement the heuristic algorithms on hardware for RT applications. Thus, the heuristic approach will be able to outperform the classical

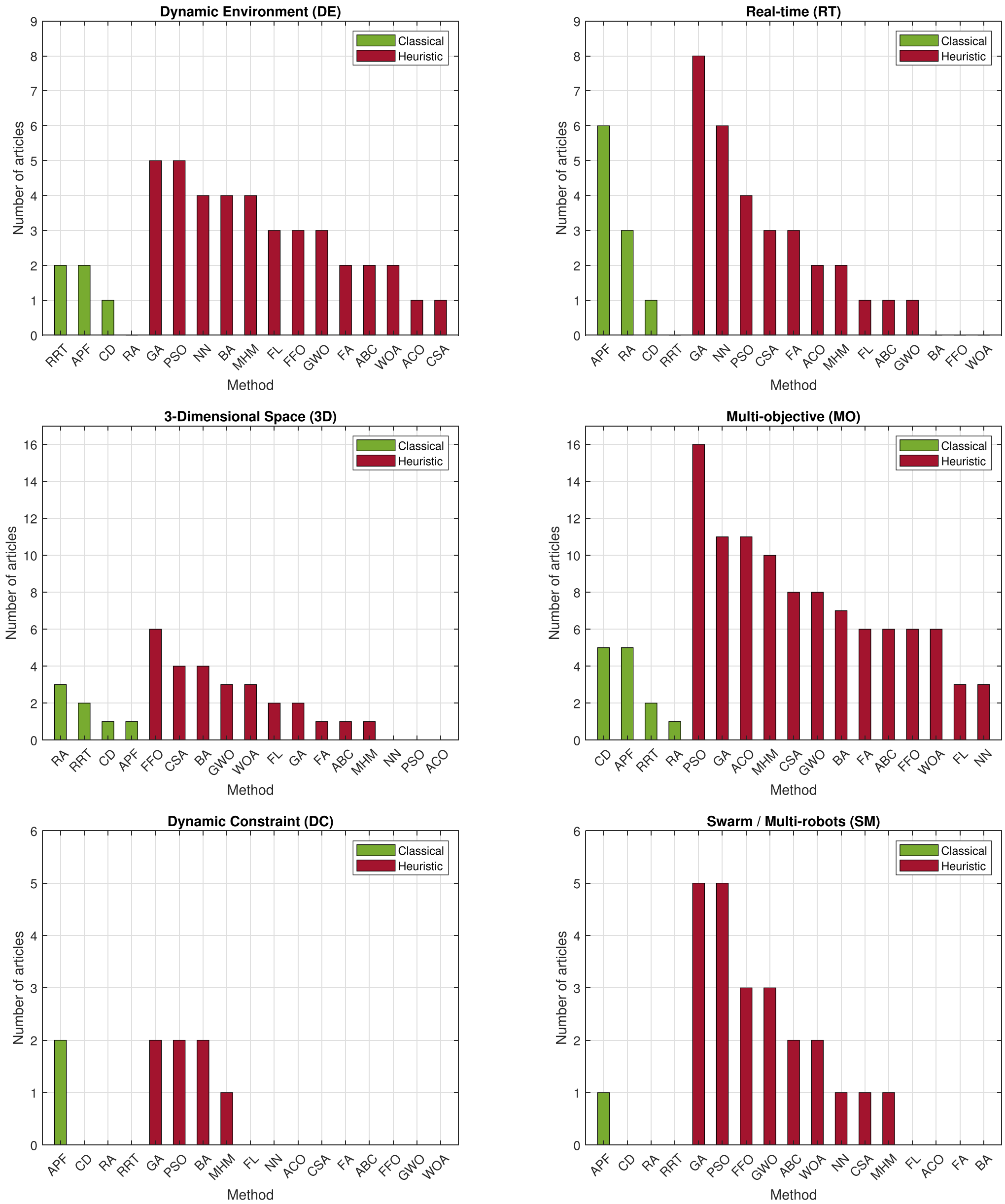
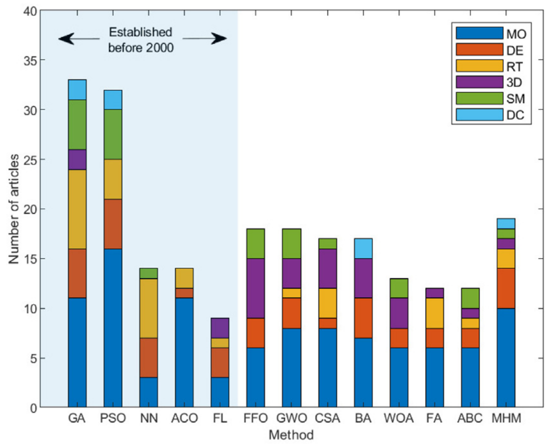


Fig. 24. Breakdown of the number of articles according to the actual path planning method for each key challenge. The green and maroon bars, which represent the classical and heuristic approaches respectively, are arranged in descending order for each category.

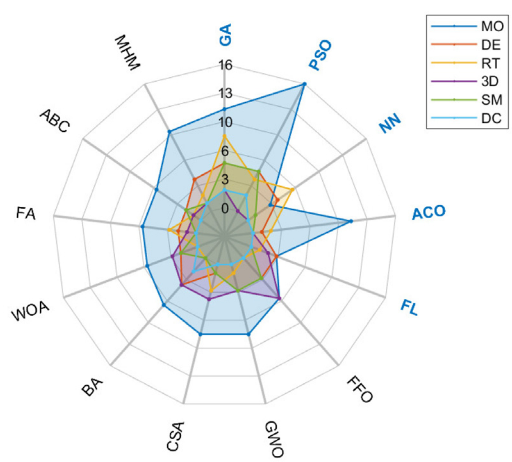
approach especially when such requirements are no longer a limitation. The difference between the two approaches as can be seen in Fig. 23 indicates that the heuristic approach is more preferable for RT applications, implying that mobile robotic technologies have been rapidly evolving in the past decade. Within the classical approach, the APF method has gained more attention than the rest as can be seen on the top right plot of Fig. 24. This is due to its simpler implementation on hardware without needing the mentioned requirements.

The DC as seen in the second last plot of Fig. 23 has received the least attention as it is relatively more challenging. Nonetheless, the DC is equally crucial in navigation as it represents the motion limitation of the robot in practice, which will eventually affect the actual performance measures such as path length, smoothness, and execution time in RT applications [268,269]. The bottom left plot of Fig. 24 shows that while the popularity of heuristic approach at handling this constraint is cumulatively higher, it is quite competitive to APF when compared individually.

The largest percentage difference between the heuristic and classical approaches in Fig. 23 is 91.6% which corresponds to the SM setting as shown in the last plot. The SM navigation is undoubtedly the most challenging setting compared to the rest as the complexity is not only similar to DE, but can exacerbate further when the navigation problem additionally includes the 3D and RT settings. Nonetheless, observing the statistical analysis in Fig. 23, the SM navigation has also received a considerable attention in the past decade which may be caused by the rising popularity of SM in various applications such as search and rescue missions [270], military, as well as service and maintenance of large areas [271]. In the bottom right plot of Fig. 24, it is clear that the heuristic approach has been used more compared to the classical approach in the SM setting. This is mainly due to the formulation of most



(a) Total number of articles for each heuristic method. The original algorithms of the first five methods from the left were established before the year 2000, while those of the rest were published later.



(b) A visualization on the strengths of each method in solving multiple challenges.

Fig. 25. Comparisons among heuristic methods on solving the identified key challenges in AMR navigation.

heuristic algorithms that allow integration of multiple constraints and objective functions which are typically imposed in the SM navigation problems.

A further analysis among the heuristic methods addressing the identified key challenges in AMR navigation is illustrated in Fig. 25. The bar chart in Fig. 25(a) compares the total number of publications between each method. The original algorithms of the first five methods from the left have been established before the year 2000, while those of the rest were published later. The radar chart in Fig. 25(b) visualizes the strengths of each method in solving those challenges. What stands out in both charts is that, all methods are capable in solving the MO optimization problems and navigations in DE settings. Fig. 25(a) also shows that, contrary to NN, ACO and FL, the solutions via PSO and GA remain relevant despite the emergence of the new navigation methods after the year 2000. On the other hand, a competitive performance among the new methods can be observed when we compare the total number of publications. A possible explanation for this might be that even though some of the algorithms are inherently incapable of solving certain navigation problems, they have other unique features that are more suited to solving other challenges, thus compensating each other.

4. Conclusion and future directions

In this literature study, over 200 articles on AMR navigation methods that were published within the last ten years have been reviewed. The key challenges and state-of-the-art techniques that can solve various path planning problems in the AMR navigation have been systematically identified. The main takeaways from this study are as follows:

- The trend in employing heuristic methods has been found to increase dramatically since the last decade, signifying their superiority over classical approaches.
- Addressing multiple objectives simultaneously such as minimizing path length and energy consumption, has become the major focus of many researches, and this can be accomplished via the heuristic approach.
- Navigation in dynamic environments is crucial as it represents the real world situation. Nonetheless, it has received less attention compared to navigation in static environments.
- Incorporating the AMR's dynamic constraints in the cost function may pose a significant challenge to the path planning strategy. However, it can be a good research direction as it reflects the robot's actual mobility restriction, which will eventually have an impact on the overall performance of the AMR in practice.

The outcome of this survey particularly the analysis of the strengths and shortcomings of each method, along with the popularity of certain techniques in solving the identified key challenges can provide useful guidance for future research into creating new strategies that can enhance the autonomy level of AMRs. Nevertheless, focusing on the path planning alone may not guarantee no failure in practical implementation. To incorporate robustness into the navigation techniques against unforeseen circumstances when deployed to the real environment, the future research agenda should include the following:

- Trajectory planning: Trajectory planning, which takes a given geometric path and endow it with the time information, is essential to optimize the robot's dynamic capabilities and motion constraints which can improve the navigation technique particularly in terms of the travel time [272,273]. In most existing techniques, it is often oversimplified to minimize the complexity of the path planning. For instance, the robot is only assumed to have a constant speed despite being able to move faster or accelerate in real implementation.
- Odometry errors: These errors which often affect the stability particularly when a closed-loop speed control is employed, can be another research direction to focus on if one is to minimize the mismatch between the simulation and experimental performance [274,275].
- Robot's sensory or perception: In most path planning techniques, sensing is often decoupled from planning. Sensors such as cameras and LiDARS can help in overcoming uncertainties in the environment, but incorporating them requires effective use of sensor feedback. The robot path planning should take into account the limitations of the sensor system as well as mechanisms that can improve the sensor feedback.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank Malaysia Ministry of Higher Education for the financial support under Fundamental Research Grant Scheme with Project Code: FRGS/1/2021/TK0/USM/02/18.

References

[1] X.V. Wang, L. Wang, A literature survey of the robotic technologies during the COVID-19 pandemic, J. Manuf. Syst. 60 (2021) 823–836.
[2] G. Fragapane, R. De Koster, F. Sgarbossa, J.O. Strandhagen, Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda, Eur. J. Oper. Res. 294 (2) (2021) 405–426.
[3] D.D. Paola, A. Milella, G. Cicirelli, A. Distante, An autonomous mobile robotic system for surveillance of indoor environments, Int. J. Adv. Rob. Syst. 7 (1) (2010) 8.
[4] A.J. Sathyamoorthy, U. Patel, M. Paul, Y. Savle, D. Manocha, Covid surveillance robot: Monitoring social distancing constraints in indoor scenarios, PLOS ONE 16 (12) (2021) 1–20.
[5] K. Berns, S.A. Mehdi, Use of an autonomous mobile robot for elderly care, 2010 Advanced Technologies for Enhancing Quality of Life (2010) 121–126.
[6] N.S. Ahmad, N.L. Boon, P. Goh, Multi-sensor obstacle detection system via model-based state-feedback control in smart cane design for the visually challenged, IEEE Access 6 (2018) 64182–64192.
[7] J. Zhong, C. Ling, A. Cangelosi, A. Lotfi, X. Liu, On the gap between domestic robotic applications and computational intelligence, Electronics 10 (2021) 793.
[8] L. Erickson, S. Valle, A simple, but np-hard, motion planning problem, in: Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013, 2013, pp. 1388–1393.
[9] T. Fraichard, Dynamic trajectory planning with dynamic constraints: A 'state-time space' approach, in: Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, 1993, pp. 1393–1400.
[10] J. Yu, S. LaValle, Structure and intractability of optimal multi-robot path planning on graphs, in: Proceedings of the 27th AAAI Conference on Artificial Intelligence, vol. 27, 2013, pp. 1443–1449.
[11] R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza, Introduction to Autonomous Mobile Robots, second ed., The MIT Press, Cambridge, MA, USA, 2011.
[12] J.C. Latombe, Exact Cell Decomposition, Springer, US, Boston, MA, 1991, pp. 200–247.
[13] R. Gonzalez, M. Kloetzer, C. Mahulea, Comparative study of trajectories resulted from cell decomposition path planning approaches, in: 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), 2017, pp. 49–54.
[14] A. Abbadi, R. Matousek, Hybrid rule-based motion planner for mobile robot in cluttered workspace, Soft. Comput. 22 (2018) 1815–1831.
[15] P. Zhou, Z. min Wang, Z. nan Li, Y. Li, Complete coverage path planning of mobile robot based on dynamic programming algorithm, in: Proceedings of the 2nd International Conference on Electronic & Mechanical Engineering and Information Technology, Atlantis Press, 2012, pp. 1837–1841.
[16] B. Nasirian, M. Mehrandezh, F. Janabi-Sharifi, Efficient coverage path planning for mobile disinfecting robots using graph-based representation of environment, Front. Robot. AI 8 (2021) 1–19.
[17] L. Zuo, Q. Guo, X. Xu, H. Fu, A hierarchical path planning approach based on a* and least-squares policy iteration for mobile robots, Neurocomputing 170 (2015) 257–266.
[18] H. Noborio, T. Naniwa, S. Arimoto, A quadtree-based path-planning algorithm for a mobile robot, J. Robot. Syst. 7 (4) (1990) 555–574.
[19] A. Swingler, S. Ferrari, A cell decomposition approach to cooperative path planning and collision avoidance via disjunctive programming, in: 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 6329–6336.
[20] L. Kavraki, P. Svestka, J.C. Latombe, M. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE Trans. Robot. Autom. 12 (4) (1996) 566–580.
[21] F. Lingelbach, Path planning using probabilistic cell decomposition, in: IEEE International Conference on Robotics and Automation, 2004. Proceedings, vol. 1, 2004, pp. 467–472.
[22] F. Yan, Y.S. Liu, J. Xiao, Path planning in complex 3d environments using a probabilistic roadmap method, Int. J. Autom. Comput. 10 (2013) 525–533.
[23] B. Dugarjav, S. Lee, D.H. Kim, J.H. Kim, N.Y. Chong, Scan matching online cell decomposition for coverage path planning in an unknown environment, Int. J. Precis. Eng. Manuf. 14 (2013) 1551–1558.
[24] J.W. Jung, B.C. So, J.G. Kang, D.W. Lim, Y. Son, Expanded douglas–peucker polygonal approximation and opposite angle-based exact cell decomposition for path planning with curvilinear obstacles, Appl. Sci. 9 (2019) 638.
[25] 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 9 (2021) 149982–149992.
[26] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, J.C. Nuño, From time series to complex networks: The visibility graph, Proc. Natl. Acad. Sci. 105 (13) (2008) 4972–4975.
[27] T. Lv, C. Zhao, J. Bao, A Global Path Planning Algorithm Based on Bidirectional SVGA, J. Robot. 2017 (2017) 1–11.
[28] T.Q. Toan, A.A. Sorokin, V.T.H. Trang, Using modification of visibility-graph in solving the problem of finding shortest path for robot, 2017 International Siberian Conference on Control and Communications (2017) 1–6.
[29] D.T. Lee, R.L.S. Drysdale, Generalization of voronoi diagrams in the plane, SIAM J. Comput. 10 (1981) 73–87.
[30] S. Garrido, L. Moreno, Mobile Robot Path Planning using Voronoi Diagram and Fast Marching (2015) 92–108.
[31] M. Özcan, U. Yaman, A continuous path planning approach on voronoi diagrams for robotics and manufacturing applications, Proc. Manuf. 38 (2019) 1–8.
[32] M. Elbanhawi, M. Simic, R. Jazar, Autonomous robot path planning: An adaptive roadmap approach, Appl. Mech. Mater. 373–375 (2013) 9.
[33] H. Niu, Y. Lu, A. Savvaris, A. Tsourdos, Efficient path planning algorithms for unmanned surface vehicle, in: 10th IFAC Conference on Control Applications in Marine SystemsCAMS, vol. 49, 2016, pp. 121–126.
[34] H. Niu, A. Savvaris, A. Tsourdos, Z. Ji, Voronoi-visibility roadmap-based path planning algorithm for unmanned surface vehicles, J. Navigation 72 (4) (2019) 850–874.
[35] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271.
[36] D.R. Lanning, G.K. Harrell, J. Wang, Dijkstra's algorithm and google maps, in: Proceedings of the 2014 ACM Southeast Regional Conference, no. 30, Association for Computing Machinery, New York, NY, USA, 2014, pp. 1–3.
[37] M. De Ryck, M. Versteyhe, K. Shariatmadar, Resource management in decentralized industrial automated guided vehicle systems, J. Manuf. Syst. 54 (2020) 204–214.
[38] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107.
[39] S. Koenig, M. Likhachev, D* lite, in: Proceedings of 18th AAAI Conference on Artificial Intelligence (AAAI '02), 2002, pp. 476–483.
[40] G. Chen, N. Luo, D. Liu, Z. Zhao, C. Liang, Path planning for manipulators based on an improved probabilistic roadmap method, Robot. Comput.-Integr. Manuf. 72 (2021) 102196.
[41] F. Yan, Y.S. Liu, J. Xiao, Path planning in complex 3d environments using a probabilistic roadmap method, Int. J. Autom. Comput. 10 (2013) 525–533.
[42] Q. Xue, P. Cheng, N. Cheng, Offline path planning and online replanning of uavs in complex terrain, in: Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference, 2014, pp. 2287–2292.
[43] S.M. Lavalle, Rapidly-exploring random trees: A new tool for path planning (Tech. rep.), Department of Computer Science, Iowa State University, 1998.
[44] S.M. LaValle, J. James, J. Kuffner, Randomized kinodynamic planning, Int. J. Robot. Res. 20 (5) (2001) 378–400.
[45] J. Kuffner, S. LaValle, RRT-Connect: An Efficient Approach to Single-Query Path Planning, in: Proceedings - IEEE International Conference on Robotics and Automation, vol. 2, 2000, pp. 995–1001.
[46] J.G. Kang, D.W. Lim, Y. Choi, W.J. Jang, J.W. Jung, Improved RRT-Connect Algorithm Based on Triangular Inequality for Robot Path Planning, Sensors 21 (2) (2021) 333.
[47] A.H. Qureshi, Y. Ayaz, Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments, Robot. Autonomous Syst. 68 (2015) 1–11.
[48] Z. Tahir, A.H. Qureshi, Y. Ayaz, R. Nawaz, Potentially guided bidirectionalized rrt* for fast optimal path planning in cluttered environments, Robot. Autonomous Syst. 108 (2018) 13–27.
[49] S. Karaman, M. Walter, A. Perez, E. Frazzoli, S. Teller, Anytime motion planning using the rrt*, in: Proceedings - IEEE International Conference on Robotics and Automation, 2011, pp. 1478–1483.
[50] I. Noreen, A. Khan, Z. Habib, Optimal Path Planning using RRT* based Approaches: A Survey and Future Directions, Int. J. Adv. Comput. Sci. Appl. 7 (11) (2016) 97–107.
[51] F. Islam, J. Nasir, U.A. Malik, Y. Ayaz, O. Hasan, RRT*-Smart: Rapid convergence implementation of RRT* towards optimal solution, 2012 IEEE International Conference on Mechatronics and Automation (2012) 1651–1656.
[52] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, M.S. Muhammad, Rrt*-smart: A rapid convergence implementation of rrt*, Int. J. Adv. Rob. Syst. 10 (7) (2013) 299.
[53] I.B. Jeong, S.J. Lee, J.H. Kim, Quick-rrt*: Triangular inequality-based implementation of rrt* with improved initial solution and convergence rate, Expert Syst. Appl. 123 (2019) 82–90.
[54] I. Pérez-Hurtado, M.J. Pérez-Jiménez, G. Zhang, D. Orellana-Martín, Robot path planning using rapidly-exploring random trees: A membrane computing approach, in: 2018 7th International Conference on Computers Communications and Control, 2018, pp. 37–46.
[55] C. Yuan, G. Liu, W. Zhang, X. Pan, An efficient rrt cache method in dynamic environments for path planning, Robot. Autonomous Syst. 131 (2020) 103595.
[56] H. Mohammed, L. Romdhane, M.A. Jaradat, Rrt*n: an efficient approach to path planning in 3d for static and dynamic environments, Adv. Robot. 35 (3–4) (2021) 168–180.
[57] B. Lindqvist, A.A. Agha-Mohammadi, G. Nikolakopoulos, Exploration-rrt: A multi-objective path planning and exploration framework for unknown and unstructured environments, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE Press, 2021, pp. 3429–3435.

[58] G. Oriolo, M. Vendittelli, L. Freda, G. Troso, The srt method: Randomized strategies for exploration, in: IEEE International Conference on Robotics and Automation, vol. 5, 2004, pp. 4688–4694.
[59] Z. Yiping, G. Jian, Z. Rui-lei, C. Qingwei, A srt-based path planning algorithm in unknown complex environment, in: The 26th Chinese Control and Decision Conference, 2014, pp. 3857–3862.
[60] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, Int. J. Robot. Res. 5 (1) (1986) 90–98.
[61] M. Mabrouk, C. McInnes, Solving the potential field local minimum problem using internal agent states, Robot. Autonomous Syst. 56 (12) (2008) 1050–1060.
[62] M. Guerra, D. Efimov, G. Zheng, W. Perruquetti, Avoiding local minima in the potential field method using input-to-state stability, Control Eng. Practice 55 (2016) 174–184.
[63] A. Azzabi, K. Nouri, An advanced potential field method proposed for mobile robot path planning, Trans. Inst. Meas. Control 41 (11) (2019) 3132–3144.
[64] X. Lin, Z.Q. Wang, X.Y. Chen, Path planning with improved artificial potential field method based on decision tree, in: 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), 2020, pp. 1–5.
[65] Q. Li, L. Wang, B. Chen, Z. Zhou, An improved artificial potential field method for solving local minimum problem, in: Proceedings of the 2nd International Conference on Intelligent Control and Information Processing, vol. 1, 2011, pp. 420–424.
[66] W. Chen, X. Wu, Y. Lu, An improved path planning method based on artificial potential field for a mobile robot, Cybern. Inf. Technol. 15 (2015) 181–191.
[67] T. Weerakoon, K. Ishii, A.A.F. Nassiraei, An artificial potential field based mobile robot navigation method to prevent from deadlock, J. Artif. Intell. Soft Comput. Res. 5 (2015) 189–203.
[68] P. Hou, H. Pan, C. Guo, Simulation research for mobile robot path planning based on improved artificial potential field method recommended by the asiasim, Int. J. Model. Simul. Scientific Comput. 08 (02) (2017) 1750046.
[69] M. Zha, Z. Wang, J. Feng, X. Cao, Unmanned vehicle route planning based on improved artificial potential field method, J. Phys: Conf. Ser. 1453 (2020) 012059.
[70] X. Fan, Y. Guo, H. Liu, B. Wei, W. Lyu, Improved artificial potential field method applied for auv path planning, Math. Problems Eng. 2020 (6523158) (2020) 1–17.
[71] A. Lazarowska, Discrete artificial potential field approach to mobile robot path planning, in: 10th IFAC Symposium on Intelligent Autonomous Vehicles, vol. 52, 2019, pp. 277–282.
[72] C. Cheng, D. Zhu, B. Sun, Z. Chu, J. Nie, S. Zhang, Path planning for autonomous underwater vehicle based on artificial potential field and velocity synthesis, in: IEEE 28th Canadian Conference on Electrical and Computer Engineering, 2015, pp. 717–721.
[73] J. Song, C. Hao, J. Su, Path planning for unmanned surface vehicle based on predictive artificial potential field, Int. J. Adv. Robot. Syst. 17 (2) (2020) 1–13.
[74] R.M.J.A. Souza, G.V. Lima, A.S. Morais, L.C. Oliveira-Lopes, D.C. Ramos, F.L. Tofoli, Modified artificial potential field for the path planning of aircraft swarms in three-dimensional environments, Sensors 22 (4) (2022) 1558.
[75] L. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353.
[76] N.S. Ahmad, Robust $H\infty$Fuzzy Logic Control for Enhanced Tracking Performance of a Wheeled Mobile Robot in the Presence of Uncertain Nonlinear Perturbations, Sensors 20 (13) (2020) 7673.
[77] N.H. Singh, K. Thongam, Mobile robot navigation using fuzzy logic in static environments, in: The 6th International Conference on Smart Computing and Communications, vol. 125, 2018, pp. 11–17.
[78] A. Loganathan, N.S. Ahmad, P. Goh, Self-adaptive filtering approach for improved indoor localization of a mobile node with zigbee-based rssi and odometry, Sensors 19 (21) (2019) 4748.
[79] F. Kamil, M.Y. Moghrabiah, Multilayer decision-based fuzzy logic model to navigate mobile robot in unknown dynamic environments, Fuzzy Inf. Eng. 14 (1) (2022) 51–73.
[80] B. Patle, G. Loganathan, D.A. Pandey, D. Parhi, J. Anne, A review: On path planning strategies for navigation of mobile robot, Defence Technol. 15 (2019) 582–606.
[81] A. Pandey, R.K. Sonkar, K.K. Pandey, D.R. Parhi, Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller, in: 2014 IEEE 8th International Conference on Intelligent Systems and Control, 2014, pp. 39–41.
[82] H. Lee, J. Jung, K. Choi, J. Park, H. Myung, Fuzzy-logic-assisted interacting multiple model (flaimm) for mobile robot localization, Robot. Autonomous Syst. 60 (2012) 1592–1606.
[83] K. Al-mutib, E. Mattar, M. Alsulaiman, Implementation of fuzzy decision based mobile robot navigation using stereo vision, Proc. Comput. Sci. 62 (2015) 143–150.
[84] M. Dirik, A. Kocamaz, O. Castillo, Global path planning and path-following for wheeled mobile robot using a novel control structure based on a vision sensor, Int. J. Fuzzy Syst. 22 (2020) 1880–1891.
[85] R. Zhao, D. Lee, H. Lee, Mobile robot navigation using optimized fuzzy controller by genetic algorithm, Int. J. Fuzzy Logic Intell. Syst. 15 (2015) 12–19.
[86] D.N.M. Abadi, M.H. Khooban, Design of optimal mamdani-type fuzzy controller for nonholonomic wheeled mobile robots, J. King Saud Univ. Eng. Sci. 27 (2015) 92–100.
[87] A. Pandey, D.R. Parhi, Optimum path planning of mobile robot in unknown static and dynamic environments using fuzzy-wind driven optimization algorithm, Defence Technol. 13 (1) (2017) 47–58.
[88] H. Omrane, M. Masmoudi, M. Masmoudi, Fuzzy logic based control for autonomous mobile robot navigation, Computational Intelligence and Neuroscience 2016 (2016) 1–10.
[89] A. Rath, D. Parhi, H. Das, P. Kumar, M. Muni, K. Salony, Path optimization for navigation of a humanoid robot using hybridized fuzzy-genetic algorithm, Int. J. Intell. Unmanned Syst. 7 (2019) 112–119.
[90] R. Al-Jarrah, A. Shahzad, H. Roth, Path planning and motion coordination for multi-robots system using probabilistic neuro-fuzzy, IFAC-PapersOnLine 48 (2015) 46–51.
[91] N. Öztürk, F. Öztürk, Neural network based non-standard feature recognition to integrate cad and cam, Comput. Ind. 45 (2) (2001) 123–135.
[92] C.H. Goay, P. Goh, N.S. Ahmad, M. Ain, Eye-height/width prediction using artificial neural networks from s-parameters with vector fitting, J. Eng. Sci. Technol. 13 (2018) 625–639.
[93] C.H. Goay, A. Abd Aziz, N.S. Ahmad, P. Goh, Eye diagram contour modeling using multilayer perceptron neural networks with adaptive sampling and feature selection, IEEE Trans. Components Packaging Manuf. Technol. 9 (12) (2019) 2427–2441.
[94] C.K. Ku, C.H. Goay, N.S. Ahmad, P. Goh, Jitter decomposition of high-speed data signals from jitter histograms with a pole–residue representation using multilayer perceptron neural networks, IEEE Trans. Electromagn. Compat. 62 (5) (2020) 2227–2237.
[95] J.H. Teo, N.S. Ahmad, P. Goh, Visual stimuli-based dynamic commands with intelligent control for reactive bci applications, IEEE Sens. J. 22 (2) (2022) 1435–1448.
[96] N.S. Ahmad, J.H. Teo, P. Goh, Gaussian process for a single-channel eeg decoder with inconspicuous stimuli and eyeblinks, Comput. Mater. Continua 73 (1) (2022) 611–628.
[97] I. Arrouch, J. Mohamad-Saleh, P. Goh, N.S. Ahmad, A Comparative Study of Artificial Neural Network Approach for Autonomous Robot's TTC Prediction, Int. J. Mech. Eng. Robot. Res. 11 (5) (2022) 345–350.
[98] I. Arrouch, N.S. Ahmad, P. Goh, J. Mohamad-Saleh, Close Proximity Time-to-Collision Prediction for Autonomous Robot Navigation: An Exponential GPR Approach, Alexandria Eng. J. 61 (12) (2022) 11171–11183.
[99] A.M. Santiago, J. Camas anzueto, J. Vazquez feijoo, H. Hernandez-De-Leon, R. Mota grajales, Neural control system in obstacle avoidance in mobile robots using ultrasonic sensors, J. Appl. Res. Technol. 12 (2014) 104–110.
[100] A. Rath, H. Das, D. Parhi, P. Kumar, Application of artificial neural network for control and navigation of humanoid robot, J. Mech. Eng. Sci. 12 (2018) 3529–3538.
[101] D. Janglova, Neural networks in mobile robot motion, Int. J. Adv. Rob. Syst. 1 (1) (2004) 15–22.
[102] Y. Chen, J. Liang, Y. Wang, Q. Pan, J. Tan, J. Mao, Autonomous mobile robot path planning in unknown dynamic environments using neural dynamics, Soft. Comput. 24 (2020) 13979–13995.
[103] J. Yu, Y. Su, Y. Liao, The path planning of mobile robot by neural networks and hierarchical reinforcement learning, Front. Neurorobot. 14 (63) (2020) 1–12.
[104] D. Zhu, C. Tian, B. Sun, C. Luo, Complete coverage path planning of autonomous underwater vehicle based on gbnn algorithm, J. Intell. Robot. Syst. 94 (2019) 237–249.
[105] B. Sun, D. Zhu, C. Tian, C. Luo, Complete coverage autonomous underwater vehicles path planning based on glasius bio-inspired neural network algorithm for discrete and centralized programming, IEEE Trans. Cogn. Develop. Syst. 11 (1) (2019) 73–84.
[106] H. Bae, G. Kim, J. Kim, D. Qian, S. Lee, Multi-robot path planning method using reinforcement learning, Appl. Sci. 9 (15) (2019) 3057.
[107] M. Algabri, H. Mathkour, H. Ramdane, Mobile robot navigation and obstacle-avoidance using anfis in unknown environment, Int. J. Comput. Appl. 91 (2014) 36–41.
[108] P.K. Mohanty, D.R. Parhi, A new hybrid intelligent path planner for mobile robot navigation based on adaptive neuro-fuzzy inference system, Austr. J. Mech. Eng. 13 (3) (2015) 195–207.
[109] M. Lazreg, N. Benamrane, Intelligent system for robotic navigation using anfis and acor, Appl. Artif. Intell. 33 (5) (2019) 399–419.
[110] M. Samadi Gharajeh, H.B. Jond, An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system, Ain Shams Eng. J. 13 (1) (2022) 101491.
[111] J.K. Pothal, D.R. Parhi, Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system, Robot. Autonomous Syst. 72 (2015) 48–58.
[112] D.A. Pandey, D. Parhi, Multiple mobile robots navigation and obstacle avoidance using minimum rule based anfis network controller in the cluttered environment, SOJ Robot. Autom. 1 (2016) 1–11.
[113] D.A. Pandey, A.K. Kashyap, D. Parhi, B. Patle, Autonomous mobile robot navigation between static and dynamic obstacles using multiple anfis architecture, World J. Eng. 16 (2019) 275–286.
[114] A. Aouf, L. Boussaid, A. Sakly, TLBO-Based Adaptive Neurofuzzy Controller for Mobile Robot Navigation in a Strange Environment, Comput. Intell. Neurosci. 2018 (3145436) (2018) 1–8.
[115] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, MA, USA, 1992.

[116] Y.V. Pehlivanoglu, A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous uav, Aerosp. Sci. Technol. 16 (1) (2012) 47–55.
[117] M. Samadi, M.F. Othman, Global path planning for autonomous mobile robot using genetic algorithm, 2013 International Conference on Signal-Image Technology Internet-Based Systems (2013) 726–730.
[118] J. Lee, D.W. Kim, An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph, Inf. Sci. 332 (2016) 1–18.
[119] A. Tuncer, M. Yildirim, Dynamic path planning of mobile robots with improved genetic algorithm, Comput. Electr. Eng. 38 (6) (2012) 1564–1572.
[120] M. Alajlan, A. Koubaa, I. Chaari, H. Bennaceur, A. Ammar, Global path planning for mobile robots in large-scale grid environments using genetic algorithms, 2013 International Conference on Individual and Collective Behaviors in Robotics (2013) 1–8.
[121] Z. Qiongbing, D. Lixin, A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems, Expert Syst. Appl. 60 (2016) 183–189.
[122] C. Lamini, S. Benhlima, A. Elbekri, Genetic algorithm based approach for autonomous mobile robot path planning, in: Proceedings of the First International Conference on Intelligent Computing in Data Sciences, vol. 127, 2018, pp. 180–189.
[123] R.M.C. Santiago, A.L. De Ocampo, A.T. Ubando, A.A. Bandala, E.P. Dadios, Path planning for mobile robots using genetic algorithm and probabilistic roadmap, in: 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, 2017, pp. 1–5.
[124] C.C. Tsai, H.C. Huang, C.K. Chan, Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation, IEEE Trans. Industr. Electron. 58 (10) (2011) 4813–4821.
[125] X. Zhang, Y. Zhao, N. Deng, K. Guo, Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm, Int. J. Adv. Rob. Syst. 13 (3) (2016) 91.
[126] H. Qu, K. Xing, T. Alexander, An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots, Neurocomputing 120 (2013) 509–517.
[127] K. Hao, J. Zhao, K. Yu, C. Li, C. Wang, Path planning of mobile robots based on a multi-population migration genetic algorithm, Sensors 20 (20) (2020) 5873.
[128] J. Ni, K. Wang, H. Huang, L. Wu, C. Luo, Robot path planning based on an improved genetic algorithm with variable length chromosome, in: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016, pp. 145–149.
[129] J. Xin, J. Zhong, F. Yang, Y. Cui, J. Sheng, An improved genetic algorithm for path-planning of unmanned surface vehicle, Sensors 19 (11) (2019) 2640.
[130] A. Kumar, P. Kumar, D. Parhi, Intelligent navigation of humanoids in cluttered environments using regression analysis and genetic algorithm, Arab. J. Sci. Eng. 43 (2018) 7655–7678.
[131] Z. Han, D. Wang, F. Liu, Z. Zhao, Multi-agv path planning with double-path constraints by using an improved genetic algorithm, PLOS ONE 12 (7) (2017) 1–16.
[132] B. Patle, D. Parhi, A. Jagadeesh, S.K. Kashyap, Matrix-binary codes based genetic algorithm for path planning of mobile robot, Comput. Electr. Eng. 67 (2018) 708–728.
[133] Y. Liang, L. Wang, Applying genetic algorithm and ant colony optimization algorithm into marine investigation path planning model, Soft. Comput. 24 (2020) 8199–8210.
[134] B. Abhishek, S. Ranjit, S. Thangavelu, G. Eappen, P.R. Sivasankar, A Hybrid pso-hsa and pso-ga algorithm for 3d path planning in autonomous uavs, SN, Appl. Sci. 2 (1805) (2020) 1–16.
[135] A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, B. Bouzouia, Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control, Robot. Autonomous Syst. 89 (2017) 95–109.
[136] A.K. Rath, D.R. Parhi, H. Das, P.B. Kumar, M.K. Mahto, Design of a hybrid controller using genetic algorithm and neural network for path planning of a humanoid robot, Int. J. Intell. Unmanned Syst. 9 (3) (2020) 169–177.
[137] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[138] J. Wang, C. Ju, Y. Gao, A.K. Sangaiah, G. jun Kim, A PSO based Energy Efficient Coverage Control Algorithm for Wireless Sensor Networks, Computers, Mater. Continua 56 (3) (2018) 433–446.
[139] J. Wang, Y. Cao, L. Bin, H.J. Kim, S. Lee, Particle swarm optimization based clustering algorithm with mobile sink for WSNs, Future Gener. Comput. Syst. 76 (2016) 452–457.
[140] Y. Guo, W. Li, A. Mileham, G. Owen, Applications of particle swarm optimisation in integrated process planning and scheduling, Robot. Comput.-Integr. Manuf. 25 (2) (2009) 280–288.
[141] H. Mo, L. Xu, Research of biogeography particle swarm optimization for robot path planning, Neurocomputing 148 (2015) 91–99.
[142] M. Fetanat, S. Haghzad, S.B. Shouraki, Optimization of dynamic mobile robot path planning based on evolutionary methods, 215 AI Robotics (IRANOPEN) (2015) 1–7.
[143] Y. Zhang, D. wei Gong, J. hua Zhang, Robot path planning in uncertain environment using multi-objective particle swarm optimization, Neurocomputing 103 (2013) 172–185.
[144] H.C. Huang, Fpga-based parallel metaheuristic pso algorithm and its application to global path planning for autonomous robot navigation, J. Intell. Robot. Syst. 76 (2013) 475–488.
[145] P. Das, D.H. Behera, B. Panigrahi, A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning, Swarm Evol. Comput. 28 (2016) 14–28.
[146] X. Li, D. Wu, J. He, M. Bashir, M. Liping, An improved method of particle swarm optimization for path planning of mobile robot, J. Control Sci. Eng. 2020 (2020) 1–12.
[147] B. Song, Z. Wang, L. Zou, On global smooth path planning for mobile robots using a novel multimodal delayed pso algorithm, Cogn. Comput. 9 (2017) 5–17.
[148] L. Xu, B. Song, M. Cao, A new approach to optimal smooth path planning of mobile robots with continuous-curvature constraint, Syst. Sci. Control Eng. 9 (2021) 138–149.
[149] G. Li, W. Chou, Path planning for mobile robot using self-adaptive learning particle swarm optimization, Sci. China Inf. Sci. 61 (2018) 052204.
[150] J.H. Zhang, Y. Zhang, Y. Zhou, Path planning of mobile robot based on hybrid multi-objective bare bones particle swarm optimization with differential evolution, IEEE Access 6 (2018) 44542–44555.
[151] H.S. Dewang, P.K. Mohanty, S. Kundu, A robust path planning for mobile robot using smart particle swarm optimization, in: International Conference on Robotics and Smart Manufacturing (RoSMa2018), vol. 133, 2018, pp. 290–297.
[152] P. Das, H. Behera, B. Panigrahi, Intelligent-based multi-robot path planning inspired by improved classical q-learning and improved particle swarm optimization with perturbed velocity, Eng. Sci. Technol. 19 (1) (2016) 651–669.
[153] J. Yang, X. Wang, P.H. Bauer, Extended pso based collaborative searching for robotic swarms with practical constraints, IEEE Access 7 (2019) 76328–76341.
[154] Y. Du, A novel approach for swarm robotic target searches based on the dpso algorithm, IEEE Access 8 (2020) 226484–226505.
[155] H. Tang, W. Sun, H. Yu, A. Lin, M. Xue, Y. Song, A novel hybrid algorithm based on pso and foa for target searching in unknown environments, Appl. Intell. 49 (2018) 2603–2622.
[156] F.H. Ajeil, I.K. Ibraheem, M.A. Sahib, A.J. Humaidi, Multi-objective path planning of an autonomous mobile robot using hybrid pso-mfb optimization algorithm, Appl. Soft Comput. 89 (2020) 106076.
[157] M. Lazreg, N. Benamrane, Hybrid system for optimizing the robot mobile navigation using anfis and pso, Robot. Autonomous Syst. 153 (2022) 104114.
[158] X. Cheng, J. Li, C. Zheng, J. Zhang, M. Zhao, An improved pso-gwo algorithm with chaos and adaptive inertial weight for robot path planning, Front. Neurorobot. 15 (2021) 1–10.
[159] Z. Lin, M. Yue, G. Chen, J. Sun, Path planning of mobile robot with pso-based apf and fuzzy-based dwa subject to moving obstacles, Trans. Inst. Meas. Control 44 (1) (2022) 121–132.
[160] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1997) 53–66.
[161] J. Liu, J. Yang, H. Liu, X. Tian, M. Gao, An improved ant colony algorithm for robot path planning, Soft. Comput. 21 (2017) 5829–5839.
[162] Y. Tao, H. Gao, F. Ren, C. Chen, T. Wang, H. Xiong, S. Jiang, A mobile service robot global path planning method based on ant colony optimization and fuzzy control, Appl. Sci. 11 (2021) 3605.
[163] K. Akka, F. Khaber, Mobile robot path planning using an improved ant colony optimization, Int. J. Adv. Rob. Syst. 15 (3) (2018) 1–7.
[164] P.B. Kumar, C. Sahu, D.R. Parhi, A hybridized regression-adaptive ant colony optimization approach for navigation of humanoids in a cluttered environment, Appl. Soft Comput. 68 (2018) 565–585.
[165] X. Dai, S. Long, Z. Zhang, D. Gong, Mobile robot path planning based on ant colony algorithm with a* heuristic method, Front. Neurorobot. 13 (15) (2019) 1–9.
[166] H.J. Wang, Y. Fu, Z.Q. Zhao, Y.J. Yue, K. Watanabe, An improved ant colony algorithm of robot path planning for obstacle avoidance, J. Robot. 2019 (2019) 1–8.
[167] J. Ou, M. Wang, Path planning for omnidirectional wheeled mobile robot by improved ant colony optimization, 2019 Chinese Control Conference (CCC) (2019) 2668–2673.
[168] Q. Luo, H. Wang, Y. Zheng, J. He, Research on path planning of mobile robot based on improved ant colony algorithm, Neural Comput. Appl. 32 (2020) 1555–1566.
[169] L. Wang, Path planning for unmanned wheeled robot based on improved ant colony optimization, Meas. Control 53 (5) (2020) 1014–1021.
[170] S. Zhang, J. Pu, Y. Si, L. Sun, Path planning for mobile robot using an enhanced ant colony optimization and path geometric optimization, Int. J. Adv. Rob. Syst. 18 (3) (2021) 1–15.
[171] W. Gao, Q. Tang, B. Ye, Y. Yang, J. Yao, An enhanced heuristic ant colony optimization for mobile robot path planning, Soft. Comput. 24 (2020) 6139–6150.
[172] F. Hassan Ajeil, I. Ibraheem, A. Azar, A. Humaidi, Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments, Sensors 20 (2020) 1880.
[173] M. Wang, C. Zhu, F. Wang, T. Li, X. Zhang, Multi-factor of path planning based on an ant colony optimization algorithm, Ann. GIS 26 (2) (2020) 101–112.

[174] B. Song, H. Miao, L. Xu, Path planning for coal mine robot via improved ant colony optimization algorithm, Syst. Sci. Control Eng. 9 (1) (2021) 283–289.
[175] G. Chen, J. Liu, Mobile robot path planning using ant colony algorithm and improved potential field method, Comput. Intell. Neurosci. 2019 (2019) 1–10.
[176] Y. Cui, J. Ren, Y. Zhang, Path planning algorithm for unmanned surface vehicle based on optimized ant colony algorithm, IEEJ Trans. Electr. Electron. Eng. 136 (2022) 1–9.
[177] X.S. Yang, S. Deb, Cuckoo search via lévy flights, 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (2009) 210–214.
[178] R. Rajabioun, Cuckoo optimization algorithm, Appl. Soft Comput. 11 (8) (2011) 5508–5518.
[179] P. Mohanty, D. Parhi, Optimal path planning for a mobile robot using cuckoo search algorithm, J. Exp. Theor. Artif. Intell. 28 (2014) 1–18.
[180] K. Sharma, S. Singh, R. Doriya, Optimized cuckoo search algorithm using tournament selection function for robot path planning, Int. J. Adv. Rob. Syst. 18 (3) (2021) 1–11.
[181] P. Mohanty, An intelligent navigational strategy for mobile robots in uncertain environments using smart cuckoo search algorithm, J. Ambient Intell. Humanized Comput. 11 (2020) 1–16.
[182] J.S. Pan, J.L. Liu, S.C. Hsiung, Chaotic cuckoo search algorithm for solving unmanned combat aerial vehicle path planning problems, in: Proceedings of the 2019 11th International Conference on Machine Learning and Computing, 2019, p. 224–230.
[183] P.C. Song, J.S. Pan, S.C. Chu, A parallel compact cuckoo search algorithm for three-dimensional path planning, Appl. Soft Comput. 94 (2020) 106443.
[184] P. Mohanty, D. Parhi, A new hybrid optimization algorithm for multiple mobile robots navigation based on the cs-anfis approach, Memetic Comput. 7 (2015) 255–273.
[185] G.G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, M. Shao, A hybrid metaheuristic de/cs algorithm for ucav three-dimension path planning, The Scientific World J. 2012 (2012) 583973.
[186] M. Saraswathi, G.B. Murali, B. Deepak, Optimal path planning of mobile robot using hybrid cuckoo search-bat algorithm, in: International Conference on Robotics and Smart Manufacturing (RoSMa2018), vol. 133, 2018, pp. 510–517.
[187] C. Xie, H. Zheng, Application of improved cuckoo search algorithm to path planning unmanned aerial vehicle, Intelligent Computing Theories and Application (2016) 722–729.
[188] J. Wang, X. Shang, T. Guo, J. Zhou, S. Jia, C. Wang, Optimal path planning based on hybrid genetic-cuckoo search algorithm, in: 2019 6th International Conference on Systems and Informatics (ICSAI), 2019, pp. 165–169.
[189] X.S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84.
[190] C. Liu, Z. Gao, W. Zhao, A new path planning method based on firefly algorithm, 2012 Fifth International Joint Conference on Computational Sciences and Optimization (2012) 775–778.
[191] G. Xu, T.W. Zhang, Q. Lai, J. Pan, B. Fu, X. Zhao, A new path planning method of mobile robot based on adaptive dynamic firefly algorithm, Mod. Phys. Lett. B 34 (29) (2020) 2050322.
[192] B. Patle, D. Parhi, J. Anne, S.K. Kashyap, On firefly algorithm: Optimization and application in mobile robot navigation, World J. Eng. 14 (2017) 65–76.
[193] B. Patle, A. Pandey, A. Jagadeesh, D. Parhi, Path planning in uncertain environment by using firefly algorithm, Defence Technol. 14 (6) (2018) 691–701.
[194] C. Liu, Y. Zhao, F. Gao, L. Liu, Three-dimensional path planning method for autonomous underwater vehicle based on modified firefly algorithm, Math. Problems Eng. 2015 (2015) 1–10.
[195] A. Hidalgo-Paniagua, M.A. Vega-Rodríguez, J. Ferruz, N. Pavón, Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach, Soft. Comput. 21 (2015) 949–964.
[196] T.W. Zhang, G.H. Xu, X.S. Zhan, T. Han, A new hybrid algorithm for path planning of mobile robot, J. Supercomput. 78 (3) (2022) 4158–4181.
[197] H. Fu, X. Liu, A path planning method for mobile robots based on fuzzy firefly algorithms, Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science) 14 (9) (2021) 3040–3045.
[198] T. Fan, J. Wang, M. Feng, X. Zhang, J. Wang, R. Wu, Application of multi-objective firefly algorithm based on archive learning in robot path planning, Int. J. Intell. Inf. Database Syst. 12 (3) (2019) 199–211.
[199] Z. Garip, D. Karayel, M. Erhan Çimen, A study on path planning optimization of mobile robots based on hybrid algorithm, Concurrency and Computation: Practice and Experience 34 (5) (2022) e6721.
[200] B. Patel, B. Patle, Analysis of firefly–fuzzy hybrid algorithm for navigation of quad-rotor unmanned aerial vehicle, Inventions 5 (2020) 48.
[201] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[202] B. Li, L.G. Gong, W.L. Yang, An improved artificial bee colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning, The Scientific World J. 2014 (2014) 232704.
[203] F. Xu, H. Li, C.M. Pun, H. Hu, Y. Li, Y. Song, H. Gao, A new global best guided artificial bee colony algorithm with application in robot path planning, Appl. Soft Comput. 88 (2020) 106037.
[204] J.H. Liang, C.H. Lee, Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm, Adv. Eng. Softw. 79 (2015) 47–56.
[205] M.A. Contreras-Cruz, V. Ayala-Ramirez, U.H. Hernandez-Belmonte, Mobile robot path planning using artificial bee colony and evolutionary programming, Appl. Soft Comput. 30 (2015) 319–328.
[206] S. Kumar, A. Sikander, Optimum mobile robot path planning using improved artificial bee colony algorithm and evolutionary programming, Arab. J. Sci. Eng. 47 (2022) 3519–3539.
[207] F. Abdul Qadir, S. Sharma, A. Shukla, R. Tiwari, J. Dhar, Multi-robot multi-target dynamic path planning using artificial bee colony and evolutionary programming in unknown environment, Intel. Serv. Robot. 11 (2018) 171–186.
[208] A. Nayyar, N. Nhu, R. Poonia, S. Kumar, Robot path planning using modified artificial bee colony algorithm, Front. Intell. Comput.: Theory Appl. 1014 (2020) 25–36.
[209] B.O. Rkaa Kamil, Mohamed Mohamed, Path planning of mobile robot using improved artificial bee colony algorithm, Eng. Technol. J. 38 (9) (2020) 1384–1395.
[210] H. Liu, J. Ge, Y. Wang, J. Li, K. Ding, Z. Zhang, Z. Guo, W. Li, J. Lan, Multi-uav optimal mission assignment and path planning for disaster rescue using adaptive genetic algorithm and improved artificial bee colony method, Actuators 11 (1) (2022) 1–30.
[211] X.S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Springer, Berlin, Heidelberg, 2010.
[212] I.K. Ibraheem, F.H. Ajeil, Path planning of an autonomous mobile robot in a dynamic environment using modified bat swarm optimization, ArXiv abs/1807.05352.
[213] J. Guo, Y. Gao, G. Cui, The path planning for mobile robot based on bat algorithm, Int. J. Autom. Control 9 (1) (2015) 50–60.
[214] X. Zhou, F. Gao, X. Fang, Z. Lan, Improved bat algorithm for uav path planning in three-dimensional space, IEEE Access 9 (2021) 20100–20116.
[215] F. Hassan Ajeil, I. Ibraheem, A. Azar, A. Humaidi, Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment, Int. J. Adv. Rob. Syst. 17 (2020) 1–15.
[216] G.G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, A bat algorithm with mutation for ucav path planning, The Scientific World J. 2012 (2012) 418946.
[217] G.G. Wang, H.E. Chu, S. Mirjalili, Three-dimensional path planning for ucav using an improved bat algorithm, Aerosp. Sci. Technol. 49 (2016) 231–238.
[218] X. Yuan, X. Yuan, X. Wang, Path planning for mobile robot based on improved bat algorithm, Sensors 21 (13) (2021) 4389.
[219] N. Lin, J. Tang, X. Li, L. Zhao, A novel improved bat algorithm in uav path planning, Comput. Mater. Continua 61 (2019) 323–344.
[220] W.T. Pan, A new fruit fly optimization algorithm: Taking the financial distress model as an example, Knowl. Based Syst.- KBS 26 (2012) 69–74.
[221] X. Zhang, X. Lu, S. Jia, X. Li, A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to uav path planning, Appl. Soft Comput. 70 (2018) 371–388.
[222] M. Li, Three-dimensional path planning of robots in virtual situations based on an improved fruit fly optimization algorithm, Adv. Mech. Eng. 6 (2014) 314797.
[223] Y. Mao, D. Huang, N. Qin, L. Zhu, J. Zhao, Cooperative 3d path planning of multi-uav via improved fruit fly optimization, Res. Square (2021) 1–15.
[224] K. Shi, X. Zhang, S. Xia, Multiple swarm fruit fly optimization algorithm based path planning method for multi-uavs, Appl. Sci. 10 (8) (2020) 2822.
[225] X. Zhang, S. Xia, X. Li, Quantum behavior-based enhanced fruit fly optimization algorithm with application to uav path planning, Int. J. Comput. Intell. Syst. 13 (2020) 1315–1331.
[226] R. Luo, H. Zheng, J. Guo, Solving the multi-functional heterogeneous uav cooperative mission planning problem using multi-swarm fruit fly optimization algorithm, Sensors 20 (18) (2020) 5026.
[227] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[228] S. Zhang, Z. Li, W. Pan, Grey wolf optimizer for unmanned combat aerial vehicle path planning, Adv. Eng. Softw. 99 (2016) 121–136.
[229] P.W. Tsai, T.T. Nguyen, T.K. Dao, Robot path planning optimization based on multiobjective grey wolf optimizer, in: J.S. Pan, J.C.W. Lin, C.H. Wang, X.H. Jiang (Eds.), Genetic and Evolutionary Computing, Springer International Publishing, Cham, 2017, pp. 166–173.
[230] R.K. Dewangan, A. Shukla, W.W. Godfrey, Three dimensional path planning using grey wolf optimizer for uavs, Appl. Intell. 49 (6) (2019) 2201–2217.
[231] F. Ge, K. Li, W. Xu, Y. Wang, Path planning of uav for oilfield inspection based on improved grey wolf optimization algorithm, 2019 Chinese Control And Decision Conference (2019) 3666–3671.
[232] C. Qu, W. Gai, J. Zhang, M. Zhong, A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (uav) path planning, Knowl.-Based Syst. 194 (2020) 105530.
[233] W. Zhang, S. Zhang, F. Wu, Y. Wang, Path planning of uav based on improved adaptive grey wolf optimization algorithm, IEEE Access 9 (2021) 89400–89411.
[234] Y. Hou, H. Gao, Z. Wang, C. Du, Improved grey wolf optimization algorithm and application, Sensors 22 (10) (2022) 3810.
[235] R. Kumar, L. Singh, R. Tiwari, Path planning for the autonomous robots using modified grey wolf optimization approach, J. Intell. Fuzzy Syst. 40 (2021) 9453–9470.
[236] Y. Liu, X. Li, A hybrid mobile robot path planning scheme based on modified gray wolf optimization and situation assessment, J. Robot. 2022 (2022) 1–9.

[237] R.E. Precup, E.I. Voisan, E.M. Petriu, M.L. Tomescu, R.C. David, A.I. Szedlak-Stinean, R.C. Roman, Grey wolf optimizer-based approaches to path planning and fuzzy logic-based tracking control for mobile robots, Int. J. Comput. Commun. Control 15 (3) (2020) 1–17.
[238] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[239] T.K. Dao, T.S. Pan, J.S. Pan, A multi-objective optimal mobile robot path planning based on whale optimization algorithm, in: 2016 IEEE 13th International Conference on Signal Processing (ICSP), 2016, pp. 337–342.
[240] Z. Yan, J. Zhang, Z. Yang, J. Tang, Two-dimensional optimal path planning for autonomous underwater vehicle using a whale optimization algorithm, Concurrency and Computation: Practice and Experience 33 (9) (2021) e6140.
[241] Z. Yan, J. Zhang, J. Zeng, J. Tang, Three-dimensional path planning for autonomous underwater vehicles based on a whale optimization algorithm, Ocean Eng. 250 (2022) 111070.
[242] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, H. Yang, Path planning for solar-powered uav in urban environment, Neurocomputing 275 (2018) 2055–2065.
[243] J. Liu, Z. Chen, Q. Liu, R. Shen, L. Hou, Y. Zhang, Design of mobile robot path planning algorithm based on improved whale optimization algorithm, in: 2022 7th International Conference on Multimedia and Image Processing, ICMIP 2022, Association for Computing Machinery, New York, NY, USA, 2022, pp. 231–235.
[244] X. Zong, J. Liu, Z. Ye, Y. Liu, Whale optimization algorithm based on levy flight and memory for static smooth path planning, Int. J. Mod. Phys. C 33 (10) (2022) 2250138.
[245] S. Kumar, D.R. Parhi, A.K. Kashyap, M.K. Muni, Static and dynamic path optimization of multiple mobile robot using hybridized fuzzy logic-whale optimization algorithm, Proc. Inst. Mech. Eng. Part C 235 (21) (2021) 5718–5735.
[246] M. Zhou, Z. Wang, J. Wang, Z. Dong, A hybrid path planning and formation control strategy of multi-robots in a dynamic environment, J. Adv. Comput. Intell. Intell. Inf. 26 (2022) 342–354.
[247] V. Sathiya, M. Chinnadurai, Evolutionary algorithms-based multi-objective optimal mobile robot trajectory planning, Robotica 37 (8) (2019) 1363–1382.
[248] S. Jain, S. Kumar, V.K. Sharma, R.C. Poonia, Peregrine preying pattern based differential evolution for robot path planning, J. Interdiscip. Math. 23 (2) (2020) 555–562.
[249] L. Huo, J. Zhu, Z. Li, M. Ma, A hybrid differential symbiotic organisms search algorithm for uav path planning, Sensors 21 (9) (2021) 3037.
[250] M.I.I. Abdi, M.U. Khan, A. Güneş, D. Mishra, Escaping local minima in path planning using a robust bacterial foraging algorithm, Appl. Sci. 10 (21) (2020) 7905.
[251] Y. Quan, H. Ouyang, C. Zhang, S. Li, L.Q. Gao, Mobile robot dynamic path planning based on self-adaptive harmony search algorithm and morphin algorithm, IEEE Access 9 (2021) 102758–102769.
[252] A. Hidalgo-Paniagua, M.A. Vega-Rodríguez, J. Ferruz, N. Pavón, MOSFLA-MRPP: Multi-Objective Shuffled Frog-Leaping Algorithm applied to Mobile Robot Path Planning, Eng. Appl. Artif. Intell. 44 (2015) 123–136.
[253] Z. Elmi, M. Önder Efe, Online path planning of mobile robot using grasshopper algorithm in a dynamic and unknown environment, J. Exp. Theor. Artif. Intell. 33 (3) (2021) 467–485.
[254] D. Agarwal, P.S. Bharti, Implementing modified swarm intelligence algorithm based on slime moulds for path planning and obstacle avoidance problem in mobile robots, Appl. Soft Comput. 107 (2021) 107372.
[255] P. Das, Hybridization of kidney-inspired and sine–cosine algorithm for multi-robot path planning, Arab. J. Sci. Eng. 45 (2020) 2883–2900.
[256] Y. Zhang, G. Guan, X. Pu, The robot path planning based on improved artificial fish swarm algorithm, Math. Problems Eng. 2016 (2016) 1–11.
[257] F. Li, Y. Du, K. Jia, Path planning and smoothing of mobile robot based on improved artifcial fish swarm algorithm, Scientific Rep. 12 (2022) 659.
[258] E. Tuba, I. Strumberger, D. Zivkovic, N. Bacanin, M. Tuba, Mobile robot path planning by improved brain storm optimization algorithm, 2018 IEEE Congress on Evolutionary Computation (CEC) (2018) 1–8.
[259] E. Dolicanin, I. Fetahovic, E. Tuba, R. Capor-Hrosik, M. Tuba, Unmanned combat aerial vehicle path planning by brain storm optimization algorithm, Stud. Inf. Control 27 (2018) 15.
[260] B. Evans, B. Xue, M. Zhang, What's inside the black-box?: a genetic programming method for interpreting complex machine learning models, The Genetic and Evolutionary Computation Conference (2019) 1012–1020.
[261] M. Saraswathi, B. Gunji, B. Deepak, Optimal path planning of mobile robot using hybrid cuckoo search-bat algorithm, Proc. Comput. Sci. 133 (2018) 510–517.
[262] C. Di Franco, G. Buttazzo, Energy-aware Coverage Path Planning of UAVs, in: Proceedings - 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, 2015, pp. 111–117.
[263] J.L. Choo, N.S. Ahmad, P. Goh, Autonomous navigation via visual servoing with a hybrid sensor system, Int. J. Mech. Eng. Robot. Res. 9 (6) (2020) 906–910.
[264] J.H. Teo, A. Loganathan, P. Goh, N.S. Ahmad, Autonomous mobile robot navigation via RFID signal strength sensing, Int. J. Mech. Eng. Robot. Res. 9 (8) (2020) 1140–1144.
[265] T.M. Ting, N.S. Ahmad, P. Goh, J. Mohamad-Saleh, Binaural modelling and spatial auditory cue analysis of 3d-printed ears, Sensors 21 (1) (2021) 227.
[266] C. Rascon, I. Meza, Localization of sound sources in robotics: A review, Robot. Autonomous Syst. 96 (2017) 184–210.
[267] A. Ko, H. Lau, Robot assisted emergency search and rescue system with a wireless sensor network, Int. J. Adv. Sci. Technol. 3 (2010) 69–78.
[268] S.Y. Chan, N.S. Ahmad, W. Ismail, Anti-windup compensator for improved tracking performance of differential drive mobile robot, 2017 IEEE International Systems Engineering Symposium (ISSE) (2017) 1–5.
[269] S.Y. Ng, N.S. Ahmad, Obstacle avoidance strategy for wheeled mobile robots with a simplified artificial potential field, in: K. Arai, R. Bhatia, S. Kapoor (Eds.), Intelligent Computing, Springer International Publishing, Cham, 2019, pp. 1247–1258.
[270] G.A. Cardona, J.M. Calderon, Robot swarm navigation and victim detection using rendezvous consensus in search and rescue operations, Appl. Sci. 9 (8) (2019) 1702.
[271] M. Schranz, M. Umlauft, M. Sende, W. Elmenreich, Swarm robotic behaviors and current applications, Front. Robot. AI 7 (2020) 1–20.
[272] T. Kim, K. Goto, I. Hiroki, K. Kon, N. Sato, F. Matsuno, Path planning for an autonomous mobile robot considering a region with a velocity constraint in a real environment, Artif. Life Robot. 16 (2012) 514–518.
[273] S.Y. Ng, N.S. Ahmad, A bug-inspired algorithm for obstacle avoidance of a nonholonomic wheeled mobile robot with constraints, in: Intelligent Computing, Springer International Publishing, 2019, pp. 1235–1246.
[274] V. Rajagopalan, M. Cetin, A. Kelly, Slip-aware model predictive optimal control for path following, 2016 IEEE International Conference on Robotics and Automation (ICRA) (2016) 4585–4590.
[275] F. Bertoncelli, F. Ruggiero, L. Sabattini, Wheel slip avoidance through a nonlinear model predictive control for object pushing with a mobile robot, in: 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV, vol. 52, 2019, pp. 25–30.