algorithms

MDPI

Review

Unmanned Ground Vehicle Path Planning Algorithms: A Review

Qiji Ma [1], Maolin Cai [2], Hui Zhang [3], Yeming Zhang [1,*], Feng Wei [1], Hao Yun [1] and Chong Lv [1]

[1] School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo 454000, China; 212305010031@home.hpu.edu.cn (Q.M.); elite@hpu.edu.cn (F.W.); yunhao@hpu.edu.cn (H.Y.); lvchong@hpu.edu.cn (C.L.)
[2] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China; caimaolin@buaa.edu.cn
[3] Pingyuan Filter Co., Ltd., Xinxiang 453700, China; zhangh611@avic.com
* Correspondence: zym@hpu.edu.cn

Abstract

As the core technology for realizing autonomous navigation of unmanned ground vehicles, the path planning algorithm directly determines the reliability and stability of navigation tasks in complex dynamic environments. With the expanding range of application scenarios, traditional path planning approaches have become increasingly inadequate in terms of real-time performance, dynamic obstacle avoidance, and multi-objective optimization. The recent rise in AI-based methods has provided new opportunities for this field. This paper systematically analyzes the latest research progress in this area. By reviewing and analyzing the highly recognized literature in recent years, we classify mainstream path planning and related algorithms into six types: graph-search-based, sampling-based, local optimization-based, meta-heuristic optimization, AI-based, and optimal control methods. The core improvement trends, advantages, and inherent limitations of each algorithm type are deeply analyzed. Through bibliometric analysis, we identify major gaps in current research, including over-reliance on simulation methods, overly restrictive environmental assumptions, and insufficient handling of multiple objectives. Finally, we point out the critical gap between simulation environments and real-world deployment and advocate the use of hybrid algorithms to address the deficiencies of single algorithms, along with effective validation in real environments. This direction is crucial for promoting the broader practical application of unmanned ground vehicle technology.

Keywords: unmanned ground vehicles; path planning; literature review; systematic analysis

Check for updates

Academic Editor: Roberto Montemanni

Received: 25 March 2026
Revised: 23 May 2026
Accepted: 25 May 2026
Published: 1 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Unmanned ground vehicles (UGVs) are core equipment in intelligent manufacture, logistics transport and outdoor exploration. Their path planning is the key technology that enables them to navigate autonomously. In the last few years, the research of UGV path planning has received enormous attention as several review papers have systematically reviewed its progress (Badamasi et al. [1], Dong et al. [2]). Advancements in this field have shown that the core of UGVs lies in their ability to plan and execute a trajectory from the starting point to the target point in a specific environment, satisfying safety and efficiency requirements. Therefore, the quality of the path planning algorithm directly determines the intelligence level and application effectiveness of UGVs.

Path planning acts as the "brain" for unmanned ground vehicle navigation. It is responsible for computing a feasible or optimal trajectory that satisfies specific constraints

within a given environmental model. As the application scenarios expand from structured factory floors to complex dynamic public spaces, path planning algorithms have become increasingly demanding. Algorithms must also respond to dynamic interference in real time apart from static obstacles now. In addition to theoretical optimality (shortest distance, minimum time, etc.) they need to trade off a number of objectives (path smoothness, safety, energy use, etc.). Consequently, the intelligence of the robot and the limits of its application are closely related to the performance of the path planning algorithm.

To address the aforementioned challenges, scholars both domestically and internationally have proposed a wide range of path planning algorithms over the past few decades. These algorithms with different mathematical principles or biomimetic mechanisms create different but complementary technical schools. Therefore, academia and industry urgently need a comprehensive overview of the latest developments, existing challenges, and future trends to systematically summarize the new methods and improved algorithms that have emerged with technological breakthroughs in the past, especially in the last five years.

In line with the needs of the above industries, this paper aims to provide a comprehensive technical perspective by systematically reviewing representative high-level literature in the field, particularly since 2020, while also taking into account early, pioneering, and necessary classical literature. This paper not only categorizes and summarizes existing algorithms but also focuses on the core improvement concepts, performance advantages, and inherent limitations of each algorithm type. Through a comprehensive analysis of the evaluation methods and objective settings in the existing literature, we identify the gap between current research and complex real-world needs, as well as future research trends.

In order to present the structure of this article as well as the algorithmic system investigated, a taxonomy of unmanned ground vehicle path planning algorithms is established (see Figure 1). This taxonomy is classified according to core technical principles, relevant ground scenarios and optimization objectives. The rest of the paper is organized as follows: Section 1 is this introduction. Following the classification framework in the roadmap, Section 2 will present six representative algorithms and recent improvements from six fundamental algorithm categories, namely graph-search-based, sampling-based, local optimization-based, meta-heuristic optimization-based, AI-based and optimal control-based methods. Section 3 provides a thorough analysis that examines the trends of field development, the preferences of researchers as well as the main challenges together with their bibliometric data. Section 4 describes the conclusion of the work and future works. Through this systematic review and analysis, it is hoped that this paper can provide an important reference for researchers and practitioners in related fields.

While several recent surveys have made valuable contributions to the field of UGV path planning, their scopes and emphases differ from the present work. Many existing reviews focus on specific subdomains, such as unstructured environments, specific algorithm families (e.g., meta-heuristic methods), or sensor fusion for navigation. Others concentrate on depth-oriented analyses of particular technical challenges, such as terrain traversability or geometric optimization.

In contrast, the present review offers a complementary broad-scope perspective with the following distinctive contributions:

(1) Unified taxonomy across six algorithmic paradigms:

We systematically categorize path planning algorithms into graph-search, sampling, local optimization, meta-heuristic, AI-based, and optimal control methods under one coherent framework. This panoramic view helps readers navigate the field and understand

the interrelationships among different approaches—a perspective that specialized surveys cannot provide.

(2) Emphasis on cross-paradigm fusion trends:

While existing surveys tend to treat algorithm categories separately, we explicitly analyze hybrid and hierarchical approaches (e.g., A*+DWA, RRT+APF, and DRL+IL) as an emerging direction. We argue that such fusion is critical for bridging the gap between global optimality and real-time adaptability in practical UGV deployment.

(3) Bibliometric analysis of research practices:

Based on the literature cited in this review, we quantitatively examine evaluation methods (simulation-only vs. experimental validation), research objectives (single- vs. multi-objective), and testing environments (static vs. dynamic). This analysis reveals systemic biases—such as the predominance of simulation-only validation—and provides an evidence-based diagnosis of open challenges.

(4) Focus on real-world deployment for UGVs:

Beyond algorithmic descriptions, we discuss UGV-specific constraints, including terrain traversability, nonholonomic and kinodynamic dynamics, localization and perception uncertainty, and energy consumption. We also offer actionable recommendations—such as standardized benchmarks and tiered sim-to-real validation protocols—to bridge the simulation-to-reality gap.

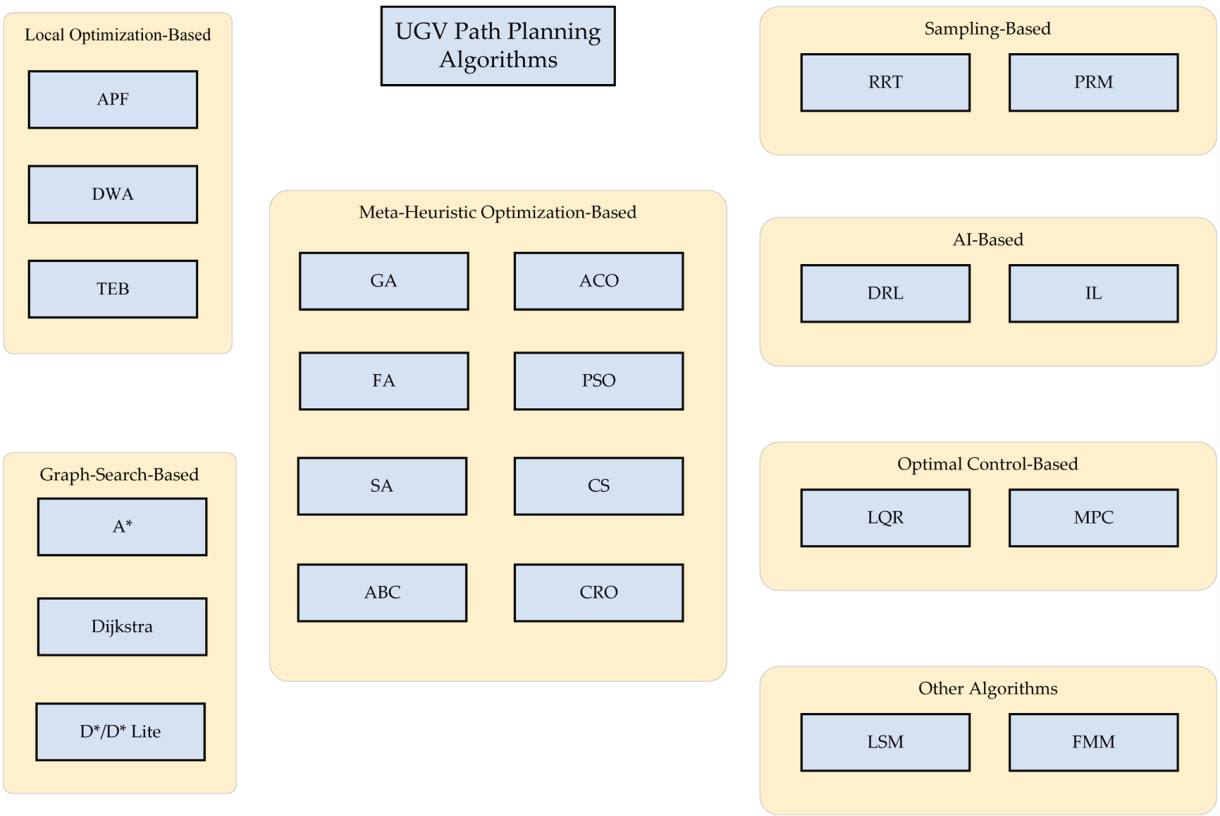


Figure 1. Taxonomy diagram of path planning algorithms for unmanned ground vehicles, categorized according to their core technical principles and mathematical foundations: graph-search (discrete optimization on graphs), sampling-based (randomized exploration of configuration space), local optimization (reactive or online trajectory generation), meta-heuristic (bio-inspired or physics-inspired stochastic search), AI-based (data-driven end-to-end learning), and optimal control (model-based trajectory optimization) methods (Created by the authors).

Taken together, this review serves as a panoramic entry point for researchers and practitioners seeking to understand the broader landscape of UGV path planning, complementing existing depth-oriented surveys that focus on specific subdomains.

Review Methodology

To ensure transparency and reproducibility, this review followed a systematic literature search and selection process. The following databases were searched: Web of Science, Scopus, IEEE Xplore, and Google Scholar. Search terms included combinations of domain keywords (e.g., "unmanned ground vehicle," "UGV," and "mobile robot") and algorithm-specific keywords (e.g., "path planning," "A*," "RRT," "DWA," "genetic algorithm," "reinforcement learning," and "MPC"). The search covered publications from 2010 to 2025, with emphasis on the last five years.

Publications were included if they proposed or improved a path planning algorithm applicable to UGVs, were peer-reviewed, and provided sufficient technical detail. Publications that focused exclusively on UAVs, manipulators, or non-robotic domains were excluded. After screening, core references formed the basis of this review. A supplementary ZIP file containing the PDFs of all cited references and an Excel spreadsheet with their categorization results (evaluation method, environment type, and optimization objective) has been submitted alongside this manuscript to ensure full reproducibility (Supplementary Materials).

For the bibliometric analysis in Section 3, each reference was categorized according to the following criteria:

Evaluation method: A study was classified as simulation-only if validation was conducted solely in a simulated environment; experiment-only if validation was performed exclusively on physical robot platforms; and combined if both simulation and real-robot experiments were included.

Environment type: A study was classified as static if it considered only fixed obstacles with no environmental changes over time; dynamic if it included moving obstacles, changing obstacle positions, or time-varying conditions; and mixed if it tested algorithms in both static and dynamic scenarios.

Optimization objective: A study was classified as single-objective if it optimized only one criterion (e.g., path length, computation time, or energy consumption) and multi-objective if it considered two or more criteria simultaneously (e.g., path length plus smoothness, safety, or energy efficiency).

2. Path Planning Algorithms

This section discusses each type of algorithm in detail based on the in-depth classification outlined in Figure 1. Our discussion is not limited to a description of the algorithms themselves but extends to an in-depth discussion of their rationale, history, performance advantages, and intrinsic limitations. This analysis sets clear technical boundaries for each type and prepares to identify trends between different paradigms.

The classification presented in Figure 1 is based on the core technical principles that underpin each algorithmic paradigm. Graph-search methods discretize the environment into graphs and perform a systematic search; sampling-based methods explore the configuration space through random sampling; local optimization methods generate trajectories online using reactive or optimization-based strategies; meta-heuristic methods simulate natural processes for a stochastic search; AI-based methods learn planning policies from data; and optimal control methods compute trajectories by solving model-based optimization problems. These categories are not strictly mutually exclusive. For instance, AI-based methods can be used to learn heuristics or costs for graph-search

algorithms, and optimal control methods (e.g., MPC) are increasingly integrated with learning-based approaches. Hybrid and hierarchical frameworks—which combine methods from two or more categories to exploit their respective strengths—appear frequently throughout the literature and are explicitly discussed within the relevant subsections below.

For completeness, this section also provides a brief overview of other important approaches, such as the level set method and the fast marching method.

2.1. Graph-Search-Based Algorithms

A path planning algorithm based on a graph search (such as the Dijkstra algorithm, A* algorithm, D* algorithm, etc.) is one of the most classic and systematic methods to solve the global or quasi-global path planning problem of UGVs. The main approach to these algorithms is to transform a continuous robotic workspace into a graphic structure consisting of nodes and edges (e.g., grid maps and topology maps). These algorithms, by defining explicit cost functions and utilizing systematic graph-search strategies (such as a breadth-first search, a heuristic search, an incremental update, etc.), find an optimal/feasible path from the starting point to the goal in an environmental model that is known/partially known. The following subsections will present the new advances in the field, from static optimality to dynamic adaptability and from independent solutions to systematic integration.

2.1.1. Dijkstra's Algorithm

Dijkstra's algorithm was first proposed by E.W. Dijkstra in 1959 [3]. This algorithm is a classic graph theory algorithm used to solve the single-source shortest path problem in weighted graphs, which involves calculating the shortest path from a specified starting point to all other nodes. Its core idea is to gradually expand the shortest path through a greedy strategy, ensuring that the currently known unvisited node with the smallest distance is selected each time and updating the distances of its adjacent nodes. Ahmad et al. [4] proposed a framework named dynamic waypoint allocation with a piecewise cubic Bezier curve (DWAPCBC), which combines Dijkstra's algorithm with dynamically allocated control points and piecewise cubic Bezier curves for path smoothing optimization to enhance path safety and fluency. Alyasin et al. [5] implemented optimal path selection for an intelligent mobile robot within a known road network by integrating Dijkstra's algorithm with the Python programming language and a Raspberry Pi 3 microcontroller, optimizing the efficiency and accuracy of path planning. Furthermore, Wang et al. [6] proposed a fuzzy logic path planning algorithm based on geometric landmarks and kinematic constraints. By combining Dijkstra's algorithm with a fuzzy logic system, this proposed algorithm was optimized to better suit the motion patterns of mobile robots. Some scholars proposed an extended version of Dijkstra's algorithm, such as Luo et al. [7], who utilize Delaunay triangulation to model surface environments. This method equivalently converts triangular meshes on the surface into triangles on a two-dimensional plane, solves for the optimal path on the 2D plane, and then obtains the optimal path on the surface through inverse transformation, thereby improving the accuracy and efficiency of surface path planning.

As one of the classic shortest-path algorithms, it performs well and guarantees optimality in static environments. Nevertheless, it has low efficiency and poor adaptive performance in dynamic environments. As a result, improvement methods, such as heuristic optimization, parallelization, and dynamic updating, have been developed to make it more feasible for the algorithm to be used in more complex scenarios.

2.1.2. A* Algorithm

The A* algorithm is generally used for global static path planning. The A* algorithm is a classic heuristic algorithm, and it is an efficient search method for finding the shortest path. It was first proposed in Hart et al. [8]. Based on previous search algorithms, such as Dijkstra's algorithm and best-first search, this research attempted to do so using heuristic functions. The A* algorithm is complete—it will always find a path if one exists—and optimal—it guarantees the shortest path—provided that the heuristic function is admissible (i.e., it never overestimates the true cost to the goal). The choice of heuristic function significantly affects the algorithm's efficiency: a well-designed heuristic can greatly reduce the search space, while a poor heuristic may degrade performance to that of Dijkstra's algorithm. In practice, A* is widely used for global path planning in static environments due to its balance of optimality and computational efficiency. Figure 2 shows an example of the A* algorithm. The red node at the bottom left represents the start point, and the red node at the top right represents the goal point. Black grids indicate static ground obstacles. White nodes represent the open list (nodes to be explored), and green nodes represent the closed list (nodes already explored). The cyan path is the optimal path generated by the algorithm.

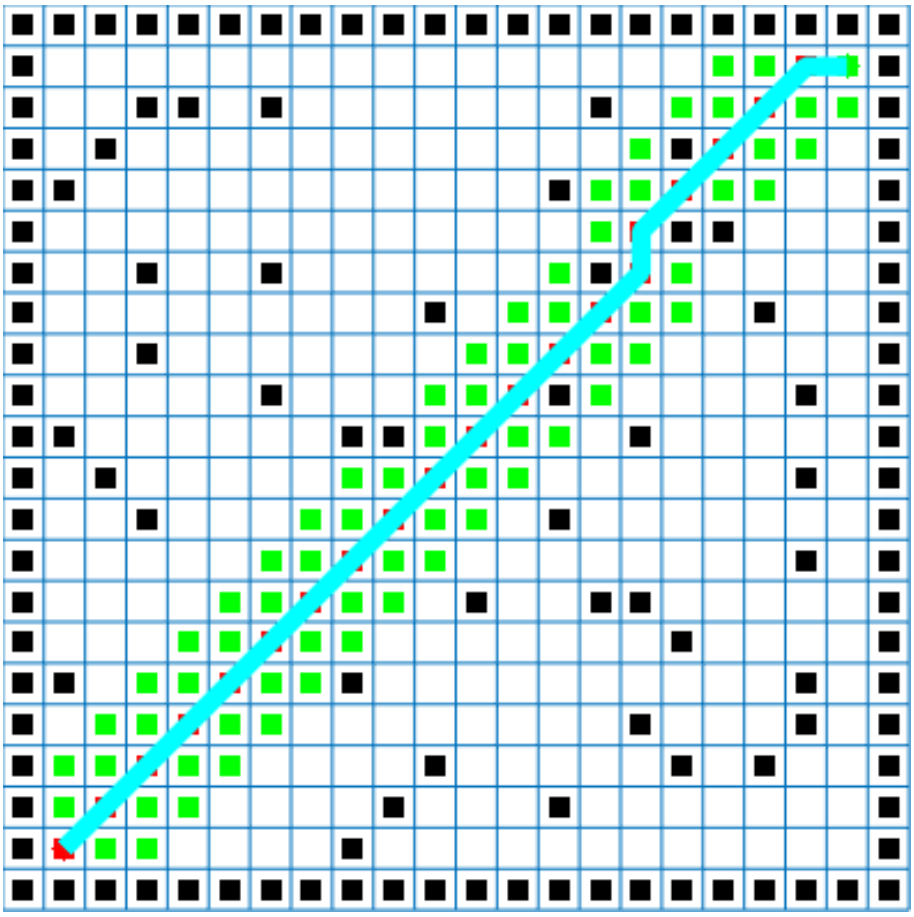


Figure 2. An example of the A* algorithm (Created by the authors).

Nonetheless, the A* algorithm has a drawback because its performance and resource usage are influenced heavily by the design of the heuristic function, and its application in dynamic environments and large-scale scenarios is very limited. Due to these shortcomings, scholars have made corresponding optimizations that have made progress. Li et al. [9] optimized the number of turning points and operational efficiency of the A* algorithm by adaptively adjusting step sizes, smoothing the path with cubic

Bezier curves, and integrating the dynamic window approach for dynamic obstacle avoidance. Lai et al. [10] proposed a fusion algorithm combining an improved A* algorithm with segmented Bezier curves. It reduced the traditional 8-neighborhood search to a 5-neighborhood search, introduced a dynamic weight factor and a key node extraction strategy, and incorporated segmented second-order Bezier curves to reduce redundant nodes and path lengths. Similarly, Liu et al. [11] generated Voronoi diagram key points via Delaunay triangulation to serve as priority search nodes for the A* algorithm, combined the grid method to extract obstacle boundaries for optimizing the obstacle avoidance strategy, and added a direct judgment mechanism to simplify the A* algorithm's search process. Furthermore, Liu et al. [12] proposed the A Star Leading Dynamic Window Approach (ASL-DWA) algorithm. This algorithm improved the A* algorithm by introducing a hybrid heuristic function (combining the Euclidean distance and point-to-line distance), the global path yaw angle, and an adaptive weight mechanism, effectively reducing search nodes, avoiding polyline paths, and handling local unknown obstacles. Likewise, Wang et al. [13] improved the A* algorithm through a dynamically weighted heuristic function (incorporating a distance weight factor and a logarithmic decay factor) and a path optimization strategy (removing redundant nodes and turning points) to enhance search efficiency and shorten the path length.

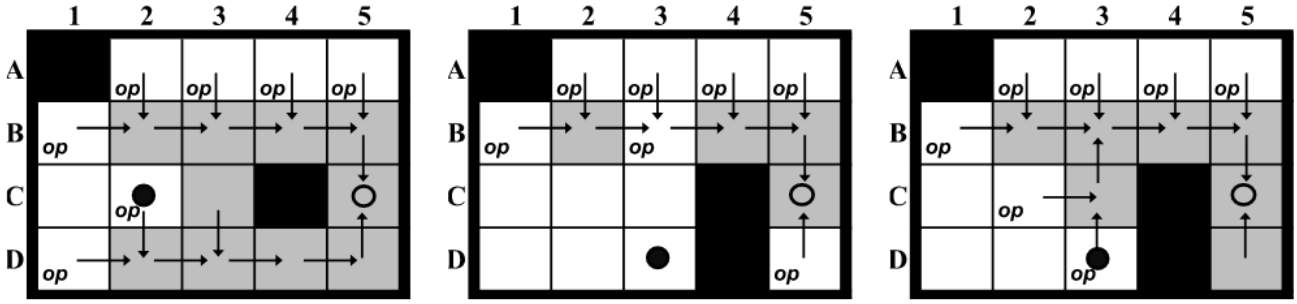
Considering the efficiency of the A* algorithm in static global planning and its limitations in dynamic environments and large-scale scenarios, existing research has primarily focused on enhancing the algorithm's adaptability, optimizing search efficiency, and improving dynamic response performance. Future development trends will concentrate on the intelligent integration of algorithms, multi-objective trade-offs, and enhancing real-time performance and robustness in complex dynamic scenarios.

2.1.3. D* Algorithm

The D* algorithm is a highly renowned dynamic path planning algorithm. Its core idea lies in efficiently replanning the path through incremental updates when environmental changes occur, without requiring global recalculation. The algorithm was first proposed by Stentz [14]. This particular implementation uses an inverse search strategy. It works back from the goal point to calculate the costs to other points on the map. As a result, when the robot moves, and there is a change (for example a new obstacle) in the environment, the D* algorithm will locally modify the cost of only the affected nodes for generating a new path quickly. It is much more efficient than standard forward-searching algorithms, such as A*. The subsequently proposed D* Lite algorithm [15], while fully inheriting this concept, achieved further simplification and efficiency improvement by introducing clearer concepts and data structures. It has become the de facto standard for dynamic planning.

Although the traditional D* algorithm possesses the advantage of dynamic replanning, it still exhibits inherent limitations when confronting complex real-world constraints, multi-objective coordination, and high-level autonomous decision-making requirements. To break through these bottlenecks, recent research has deeply expanded and innovatively integrated the D* algorithm from multiple dimensions. Yao et al. [16] effectively constrained the safe distance between the path and obstacles by combining the D* Lite algorithm with the artificial potential field (APF), significantly enhancing the navigation practicality of unmanned surface vessels in complex waterways. Zheng et al. [17] incorporated a fusion approach, wherein the D* algorithm executed global path planning and extraction of characteristic points. DWA took these points as intermediate targets for local real-time obstacle avoidance and optimal trajectory planning. By introducing a combination of static and dynamic path decision-making, the obstacle-avoidance safety and motion stability of

AGVs in complex environments was greatly improved. Maurovic et al. [18] proposed a negative edge weight D* algorithm to enable active simultaneous localization and mapping (SLAM) that simultaneously optimizes movement and localization. This allows for the robot's path to be actively attracted to key loop-closure points, thereby continuously optimizing the map and localization without interrupting the primary task. Regarding enhancements of the algorithm's core efficiency, the D* Extra Lite algorithm proposed in Przybylski et al. [19] took a unique approach. It achieved re-initialization of the affected search space through search tree branch cutting, realizing faster response times in dynamic environments compared to the classic D* Lite (see Figure 3).



(**a**) Initial status (**b**) Encounter with dynamic obstacles (**c**) Dynamic replanning

Figure 3. Example execution of D* Extra Lite on a 4-connected grid in which the filled circle represents the initial cell and the unfilled circle shows the goal cell. In the initial phase, a backwards search is carried out until the goal state is found. In (**a**), states in the open list are labeled with op, and the arrows represent the search tree of the backwards search. When a new obstacle appears in cell D4, D* Extra Lite first prunes the search tree (**b**) and then restarts the search until the initial state appears at the top of Open [20] (Reprinted from Ref. [20], with permission).

In summary, the current research trend demonstrates cross-domain integration and an intelligence-oriented direction. On the one hand, the D* algorithm is being deeply integrated into specific fields, such as robot dynamics and communication models, evolving into an optimizer for task and resource coordination. On the other hand, its design is trending towards being lightweight and modularization, serving as a core sub-module that combines with high-level semantic understanding, multi-agent collaboration, and other AI technologies to collectively construct the next-generation autonomous decision-making systems adaptable to complex scenarios.

2.1.4. Summary

Over the last few years, we have observed a shift from the shortest path pursuit of graph-search-based path planning algorithms in static environments towards improved dynamic adaptiveness, computational efficiency, and deep integration as modular components in the complete perception–decision–control stack. The major developmental trend is mainly reflected in the two aspects of intelligent core search and modular architecture. Table 1 summarizes representative graph-search-based algorithms and their key improvements.

Table 1. Representative graph-search-based algorithms: key improvements and evaluation metrics.

Algorithm	Article	Year	Improvement	Environment	Objective	Evaluation
A*	[21]	2024	Redundant safety space, turning cost, pre-judgment inflection point elimination, and safe corridor	Static	Single	Simulation
	[22]	2022	Expansion distance, bidirectional search, and path smoothing	Static	Single	Simulation + real robot

Table 1. *Cont.*

Algorithm	Article	Year	Improvement	Environment	Objective	Evaluation
D*	[23]	2024		Dynamic	Single	Simulation
Dijkstra	[24]	2022	Online obstacle detection and node exclusion for dynamic replanning	Dynamic (with online updates)	Single	Real robot experiment
	[25]	2023	PSO-based weight tuning for Dijkstra, considering time and turns	Static	Single	Simulation

To be more specific, Dijkstra's algorithm, which was used earlier for accurate cost calculation in a global search, has transformed into a cost function provider. This is for other layers of planning (for example, sampling algorithms and semantically segmented regions). The new algorithm effectively enhances computational efficiency by either improving the cost function of the original Dijkstra method or integrating other algorithms as a new cost function [24,25]. The core evolution of the A* algorithm concentrates on the intelligent design of its heuristic function. Path planning efficiency and the obstacle-avoidance effect are effectively improved by means of dynamic weighting, path optimization of a smooth path curve, or a two-way search [21,22]. The D* family of algorithms has evolved from providing simple replanning capability to an efficient incremental update engine that rapidly reacts to changes in the environment, making it the core replanning engine for safely navigating dynamically changing and unknown environments [23].

2.2. Local Optimization-Based Algorithms

Local optimization-based path planning algorithms like the artificial potential field (APF), dynamic window approach (DWA), and Timed Elastic Band (TEB) algorithm form a major category that is distinct from global graph-search methods. Their approach does not use global discretization, modeling, and a search of the environment. They operate in a continuous configuration space or velocity space or spatiotemporal domains instead. They produce, based on real-time sensory information, local motion trajectories/control commands online that satisfy robot kinematic/dynamic constraints and immediate obstacle-avoidance needs. We can do this with methods such as physics-based model simulations, multi-trajectory sampling evaluations, or a nonlinear numerical optimization. In the next sections, we will discuss the recent research progress, from the resolution of very basic obstacle avoidance to balancing multiple objective performance and from independent modules to tight coupling with global planning.

2.2.1. Artificial Potential Field Method

The artificial potential field (APF) method is a classic local path planning method whose core idea originates from the concept of potential fields in physics. This method constructs a virtual force field for autonomous path planning, where the robot moves under the combined influence of an attractive force (towards the goal) and repulsive forces (away from obstacles). Since its initial proposal by Khatib in 1986 [26], the APF method has been widely applied in obstacle avoidance and navigation tasks for autonomous systems such as robots and unmanned vehicles due to its advantages of simple computation and strong real-time responsiveness.

As shown in Figure 4, at any point in the motion space, the robot is subjected to the combined action of the attractive force, F_{att}, from the goal point and the repulsive forces, F_{rep1} and F_{rep2}, from surrounding obstacles. These forces are vectorially summed to obtain the resultant force, F_{total}, acting on the robot. The robot will move along the direction of this resultant force, thereby being "attracted" by the goal while being "pushed away" by

obstacles, achieving the purpose of automatic obstacle avoidance and progression toward the target.

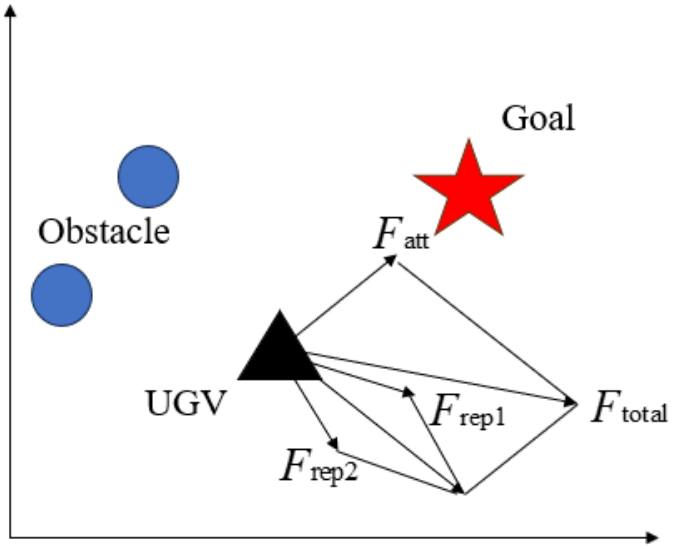


Figure 4. Schematic diagram of the APF algorithm. F_{att} denotes the attractive force generated by the goal position, which guides the robot toward the target. F_{rep1} and F_{rep2} represent the repulsive force exerted by obstacles, enabling collision avoidance. F_{total} is the resultant force obtained by the vector sum of F_{att}, F_{rep1}, and F_{rep2}, determining the robot's motion direction at each iteration (Created by the authors).

Nonetheless, the classic approach of APF is impeded by some inbuilt issues, which include the local minimum and the non-reachability of goals in complicated environments. In recent years, researchers have worked from different perspectives to improve and extend the APF. Yu et al. [27] combined APF with the Soft Actor-Critic (SAC) algorithm, introducing potential field supervised learning into the critic network to enhance the safety and convergence efficiency of the global path. Yang et al. [28] proposed an improved A* algorithm integrated with APF, introducing an adaptive guidance angle and a potential field force function for path planning and achieving dynamic maintenance of a safe distance in ocean current environments. Zhai et al. [29] designed the water-drop-shaped repulsive potential field and the dynamic obstacle potential field with relative velocity and acceleration for unmanned tracked vehicles in an unstructured environment with dynamic and static obstacles. The designed potential field thus significantly improves the smoothness and safety of the trajectory. Luo et al. [30] combined an adjustable-radius virtual potential field detection circle model with a Long Short-Term Memory (LSTM)-based Q-learning algorithm. This integration enabled the early prediction of local minima and adaptive tracking of dynamic obstacles, markedly enhancing the online obstacle-avoidance success rate and path planning reliability for unmanned vehicles in semi-enclosed dynamic obstacle environments.

To sum up, one of the development and optimization directions of the artificial potential field algorithm is the local minimum and the limitations in dynamic obstacle environments, and the other is the improvement of the dynamic obstacle potential field function in an artificial potential field. There are other remedial solutions that allow the robot to sink into the local optimum and then fix the problem with other solutions, such as adding a disturbance (random walk) or backtracking at the local minimum to jump out of the local minimum.

2.2.2. Dynamic Window Approach

The dynamic window approach (DWA) is an effective method used in UGVs' real-time local path planning. The core idea is to sample multiple groups of speed candidate solutions (linear speed and angular speed) in the speed feasible area of an unmanned vehicle, simulate the motion trajectory of each speed solution, score the trajectory with an evaluation function, and select the speed corresponding to the trajectory with the highest score as the current motion command of the unmanned vehicle to realize local obstacle avoidance and path tracking. This method was first proposed by Fox et al. in 1997 [31]. Because of its efficiency, real-time behavior and natural compatibility to robot kinematic and dynamic constraints, it has been increasingly adopted in autonomous mobile systems, especially in situations requiring online responses to dynamic obstacles.

As shown in Figure 5, within each control cycle, DWA first generates a dynamic window in the linear- and angular-velocity space based on the robot's current velocity, acceleration limits, and obstacle distribution. This window represents a set of feasible velocity pairs. Subsequently, using the robot's kinematic model, these velocities are used to simulate trajectories, yielding multiple predicted paths for a short future time horizon. In the end, an evaluation function is created that equips all factors, such as the trajectory angle with the goal, distances to obstacles and the forward speed, and all the trajectories are scored based on that. The robot's motion is driven using the velocity command related to the highest scoring trajectory.

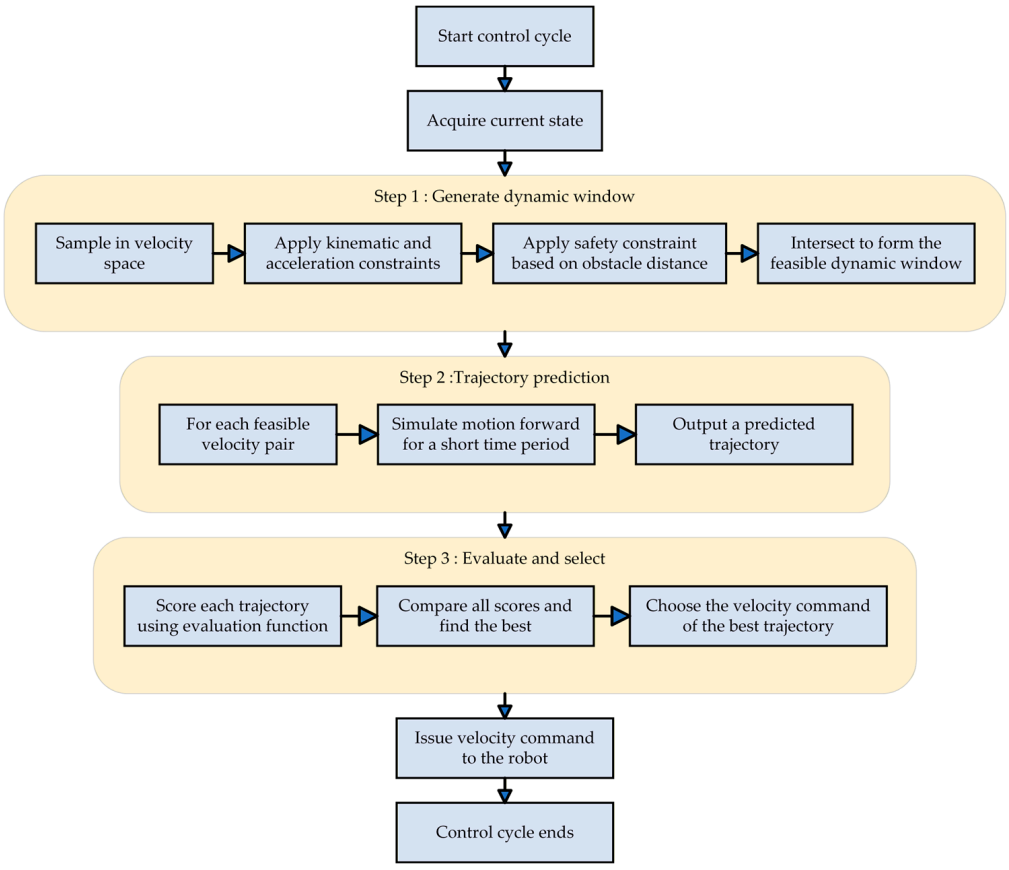


Figure 5. Flowchart of the DWA algorithm (Created by the authors).

Nonetheless, the classical DWA technique continues to have major weaknesses in complex scenarios. The first includes the tendency to fall into local optima in the absence of global path guidance. The reliance on empirical tuning for the weights of the evaluation function is similar. Then, the unstable performance in narrow passages or areas with many

dynamic obstacles is similar. Several strategies to improve these have been suggested by researchers in recent years. Hu et al. [32] presented a groundbreaking algorithm named ARIME-DWA. This algorithm systematically enhanced the traditional DWA by fusing the RIME algorithm with the Artemisinin Optimization algorithm. It utilizes key points from a global path as local sub-goals and introduces a dynamic obstacle distance term into the evaluation function, significantly improving planning safety and path quality in dynamic environments. Liu et al. [33] proposed an Adaptive Dynamic Window Approach, which dynamically adjusts the sampling velocity range based on environmental complexity and differentiates the handling of known and unknown obstacles, enhancing the system's adaptability in dynamic, unknown environments. Li et al. [34] combined a fine-tuned genetic algorithm with DWA. Through the use of key node extraction, an optimized path in order to get the fitting sub-goals for DWA, and the brake-and-wait technique for encountering oncoming dynamic obstacles, the algorithm shows efficient and safe dynamic path planning. Hou et al. [35] optimized the dynamic window approach (DWA) in two ways: first, by using the key points of the global path generated by an improved Sparrow Search Algorithm (ISA), which served as a local sub-goal sequence for DWA, and second, by designing an evaluation function that combines the tracking of the global path with separate evaluations for static and dynamic obstacles. At the same time, an adaptive velocity adjustment mechanism can automatically balance linear velocity and angular velocity in dense obstacles, thus achieving safe, smooth, real-time obstacle avoidance and effective operation in complex dynamic environments.

In conclusion, the development trend and improvement direction of the dynamic window method have evolved from basic local path generation to an improved method that can participate in global planning and adaptive path optimization. Its trends reflect a comprehensive pursuit of real-time performance, adaptability, safety, and system integration, and continue to play a key role in the navigation of UGVs.

2.2.3. Timed Elastic Band Algorithm

The Timed Elastic Band (TEB) algorithm was first proposed by Roesmann et al. [36]. The core idea is to regard the local motion trajectory of the unmanned vehicle as a "flexible" strip curve. Through joint optimization of the time and space dimensions of the trajectory, on the premise of meeting the kinematic constraints of an unmanned vehicle and obstacle-avoidance constraints, a smooth and efficient local motion trajectory is generated, and at the same time, it can dynamically respond to the real-time changes of environmental obstacles, so it is very suitable for the dynamic obstacle-avoidance requirements of an unmanned vehicle.

Despite its advantages, the TEB algorithm has certain drawbacks in practical applications. It has high computational complexity, not a good enough response to dynamic obstacles and difficulty guaranteeing the global optimality of the path. These difficulties have led researchers to boost speed and efficiency. Wu et al. [37] proposed an improved TEB algorithm that, by introducing a minimum distance constraint, enables an Automated Guided Vehicle (AGV) to drive close to walls during turns, thereby reducing the planning time and energy consumption. Li, et al. [38] proposed an improved A*-TEB fusion framework. By incorporating turning costs and dynamic weights and adopting equivalent circular obstacle modeling and a hierarchical planning strategy, the local obstacle-avoidance capability and global coordination performance of multi-robot systems were enhanced. Furthermore, Wang et al. [39] proposed a method fusing a Multiple Heuristic Rapidly Exploring Random Tree (MH-RRT) with an improved two-stage TEB. The local trajectory shows good smoothness and computational efficiency as a result of path decision and speed decision optimization in a stepwise manner. Liu et al. [40] further proposed an

improved TEB algorithm, which significantly reduced the number of node expansions and search time by employing a bidirectional A* search combined with dynamic weights and a 5-neighborhood search and achieved efficient fusion of global path planning and local obstacle avoidance by integrating it with the TEB algorithm.

The TEB algorithm performed reasonably well in a dynamic environment in real-time as an efficient local trajectory optimization method. Nonetheless, it suffers from high computational complexity, not being sufficiently responsive to dynamic obstacles, and not having a globally optimal path. With that, recent studies have tried to add heuristic constraints, hierarchical planning, stepwise optimization, or complete integration with other global planners to upgrade its overall performance under complex dynamic environments.

2.2.4. Summary

In recent years, the approach based on local optimization has been transformed from the traditional approach of avoiding near obstacles only with fixed parameters to emphasize the smoothness and balance of locally planned paths and the directional optimization of global planned path offsets. This development trend can be summa-rized in two aspects: "increasingly diversified optimization objectives" and "increas-ingly integrated decision-making". Table 2 lists representative local optimiza-tion-based algorithms and their evaluation metrics.

Table 2. Representative local optimization-based algorithms: key improvements and evaluation metrics.

Algorithm	Article	Year	Improvement	Environment	Objective	Evaluation
APF	[41]	2024	Optimizes repulsive field to place the goal at the global potential energy minimum, avoiding local minima.	Dynamic	Single	Simulation
DWA	[42]	2024	Introduces a moving evaluation function to enhance navigation in narrow passages.	Dynamic	Single	Simulation + Real Robot
TEB	[43]	2025	Adds smoothness and jerk objectives to TEB and uses fuzzy logic to dynamically adjust objective weights.	Static	Multi-objective	Simulation + Real Robot

In particular, the main development path of algorithms such as DWA and TEB is no longer to demonstrate their ability to operate in simple dynamic scenarios. Instead, we seek to build a system that will operate stably and efficiently in narrow passages and assemblies of dynamic obstacles. By incorporating global path tracking, relative velocity to dynamic obstacles, trajectory smoothness and final orientation into the design of multi-objective evaluation functions, along with the use of adaptive weight adjustment or optimization-based solvers, the local optimization framework can generate safe, comfortable and globally consistent trajectories in complex dynamic environments [42,43]. Meanwhile, APF has progressed from its basic attractive–repulsive model to become a flexible guidance field that can integrate various navigation priors. Through improvements to the potential field function to eliminate local minima or by combining potential field forces with machine learning-driven decision modules, APF has enhanced both the stability of traditional methods and its capability for semantic obstacle avoidance in unstructured environments [41].

Overall, local optimization-based planning methods are advancing towards the comprehensive and adaptive integration of optimization objectives, spatiotemporal coordination with global paths, and the explicit modeling and robust handling of uncertainties. Advancements in real-time computation and constraint satisfaction will be key for improving local optimizers, while higher-level-scene semantic understanding, pedestrian intention prediction, and multi-robot cooperation will ensure a smoother future autonomous vehicle experience. For seamless and anticipated motion in a fast-paced human–robot coexistent environment, this integration is a must.

2.3. Sampling-Based Algorithms

Sampling-based path planning algorithms (represented by the Probabilistic Roadmap Method (PRM) and the Rapidly Exploring Random Tree (RRT) method) constitute an important category distinct from graph-search methods. The foundational concept does not depend on fully modeling or searching the entire configuration space. They progressively sample regions and check for collisions, which helps them overcome high-dimensional, complex or dynamic environments. The subsequent subsections elaborate on the latest research developments in this sector.

2.3.1. Rapidly Exploring Random Tree Method

The Rapidly Exploring Random Tree (RRT) algorithm is a highly renowned sampling-based path planning algorithm. The basic principle of the RRT algorithm is to quickly explore the space and find a feasible path from the beginning to the end by randomly sampling points in the space and gradually constructing a tree structure (search tree). This algorithm was first proposed by LaValle [44]. The fast-searching random tree algorithm can effectively explore the path and find a feasible path in complex environments, but the path quality is unstable, and there may be jitters. The subsequently proposed RRT* algorithm by Karaman et al. [45] achieves continuous optimization of path quality by introducing an asymptotic optimality mechanism while completely retaining the random sampling framework. Unlike the basic RRT, which only finds a feasible path, the RRT* algorithm achieves asymptotic optimality—meaning that as the number of samples increases, the probability that the found path approaches the optimal solution tends to one. However, its convergence rate to the optimal solution is relatively slow, and the initial paths generated are typically non-smooth and require post-processing. Despite these limitations, RRT* remains one of the most influential sampling-based algorithms due to its optimality guarantee.

While the RRT algorithm can quickly find feasible paths in high-dimensional spaces, due to its randomness, it has low search efficiency and poor adaptability to complex environments (e.g., narrow passages and dense obstacles). In addition, it often generates paths of poor quality (i.e., non-optimal and non-smooth). These characteristics do not meet the practical requirements of dynamic environments or specific robot platforms directly. To overcome the shortcomings of existing algorithms, recent works have substantially improved their performance through the introduction of external information guidance and hybridization strategies. For example, Qi et al. [46] proposed a new Multi-objective RRT* (MOD-RRT*) algorithm by incorporating multi-objective optimization and dynamic replanning. It allows a robot to balance path length and smoothness in dynamic environments with rapid replanning at the millisecond level. Sheng et al. [47] combined a bidirectional search with an optimized artificial potential field (APF) to propose a Bidirectional Rapidly Exploring Random Tree Algorithm based on Adaptive Mechanisms and Artificial Potential Fields (AB-APF-RRT*). Through dynamic sampling and expansion strategies, it effectively solves search stagnation and

goal non-reachability issues in dense obstacle environments. Targeting narrow passages, Zhou et al. [48] proposed the Grid-based Variable Probability Rapidly Exploring Random Tree (GVP-RRT) algorithm. By utilizing map gridding and a variable probability growth strategy, the GVP-RRT algorithm substantially increases sampling density and planning success rates in narrow-passage regions. For unstructured mine roadways, Wu et al. [49] fused a fan-shaped goal-biased RRT with PRM. Through an adaptive step size and post-processing path optimization, it enhanced planning efficiency while ensuring the kinematic feasibility of the path.

In summary, it can be said that the current improvements of the RRT algorithm obviously follow the trend of "strategic integration and task orientation". For example, the deep integration of external strategies like APF, PRM, and multi-objective optimization compensates for the blindness caused by solely relying on random sampling. While generalization remains a focus, there is a growing trend to study more detailed modeling for specific scenarios, such as dynamic environments, narrow passageways, and unstructured terrains. This shifts the algorithm from a general-purpose exploration tool to a more specialized planner. This planner is designed for a specific set of real physical constraints and task requirements.

2.3.2. Probabilistic Roadmap Method

The Probabilistic Roadmap Method (PRM) is a classic sampling-based path planning algorithm which was first proposed in 1996 by Kavraki et al. [50]. The basic idea is to create a network representing the free space by randomly sampling a series of collision-free nodes in the configuration space and connecting these nodes to an undirected graph (i.e., the "roadmap"). PRM is probabilistically complete: if a feasible path exists, the probability of finding it approaches one as the number of random samples increases. It is particularly well-suited for multi-query planning in static environments, as the constructed roadmap can be reused for multiple start-goal queries. PRM can efficiently deal with high-dimensional state spaces and complex geometric constraints in the query phase by using this network to search for a path from start to goal, thus escaping the combinatorial explosion of graph-search methods. Due to its reasonable probabilistic completeness, PRM has become a popular choice for global path planning in robotics and autonomous driving.

PRM has serious limitations in narrow-passage situations. Because of its dependence on uniform random sampling, a narrow region which has a small volume cannot get enough valid sample points. This causes the roadmap to not connect well in these areas, which could even result in the failure to find a feasible path. In view of this bottleneck, recent studies have come up with different improvement methods, such as optimizing the sampling strategy, hybrid algorithm fusion, and intelligent search guidance. For example, Cao et al. [51] proposed an obstacle-density-based adaptive sampling strategy, which significantly increases the density of sample points within narrow passages by performing secondary uniform sampling around obstacles. Ravankar et al. [52] fused APF with PRM to propose the Hybrid Potential-based Probabilistic Roadmap (HPPRM) algorithm. This method guides the distribution of sample points using potential field information, effectively enhancing the planning success rate and path quality in narrow passages. Chai et al. [53] introduced a network of migrating sample points generated by the particle swarm optimization algorithm. The presence of obstacles allows the sample points to move towards free space through information sharing. As a result, this increases node utilization in narrow-passage regions. Ali et al. [54] optimized the learning phase of PRM by combining quasi-random sampling with the APF algorithm to increase sampling density in narrow passages. Simultaneously, it employed a bidirectional A* algorithm and

path pruning techniques to optimize the query phase, thereby achieving more efficient and safer path planning in complex narrow-passage environments.

To sum up, the ongoing optimization research on PRM has begun riding the wave of "problem orientation" and "strategy fusion". Numerous targeted sampling enhancement mechanisms have been proposed to tackle typical bottlenecks, like narrow passages, on the one hand. Conversely, the introduction of external information/algorithms involving other modalities, such as potential fields, swarm intelligence, and geometric analysis, helps in overcoming the blindness of pure random sampling. All this results in the improvements, success rates and practicality of PRM for complex environments. With increasing demand from practical scenarios such as dynamic environments and multi-robot collaboration, the improvement of PRM in the future will emphasize more on real-time performance and adaptability, as well as deep integration with local planning and motion control layers.

2.3.3. Summary

In recent years, sampling-based path planning algorithms, represented by PRM and RRT, have evolved from independent algorithms focused on single-point efficiency into systematic planning frameworks oriented towards complex environments and task requirements. Their developmental trend is characterized by two main features: "intelligent sampling" and "architectural fusion." Table 3 provides a summary of representative sampling-based algorithms.

Table 3. Representative sampling-based algorithms: key improvements and evaluation metrics.

Algorithm	Article	Year	Improvement	Environment	Objective	Evaluation
RRT	[55]	2024	Uses Halton sequence sampling instead of pseudo-random sampling, with mouse-inspired goal guidance and candidate sampling pool.	Static	Single	Simulation + Real Robot
	[56]	2025	Proposes parallel heuristic sampling, bidirectional guidance mechanism, path optimization, and offset-guided DWA.	Dynamic	Single	Simulation + Real Robot
PRM	[57]	2022	APF-enhanced PRM for narrow channels: quasi-random sampling, obstacle-to-free point conversion, bidirectional A*, and path pruning.	Static	Single	Simulation

To be exact, PRM's core development route has changed from traditional uniform random sampling to structured sampling enhancement for narrow passages. The coverage and connectivity of the roadmap have been enhanced in critical areas of the roadmap, which is the trajectory of the vehicle, by using guidance mechanisms such as artificial potential fields, quasi-random sequences, and swarm intelligence. At the same time, PRM is gradually becoming a hierarchically planning basic module that can embed local optimization, real-time querying, and safety constraints from a global static roadmap builder [57]. On the other hand, RRT has evolved from a fast exploration heuristic to a planning kernel, balancing optimality, real-time performance and dynamic adaptability. As the asymptotic optimality of the RRT-like RRT* was considered good enough other research enhancements using RRT variants, such as heuristic guidance, bidirectional expansion,

and dynamic replanning, were created to form task-adaptive types to deal with dynamic, dense and unstructured environments [55,56]. RRT is transitioning from a "feasible path finder" to a "high-quality trajectory generator." More importantly, the boundary between the two algorithm categories is becoming increasingly blurred, showing evidence of deep synergy: PRM provides structural priors for RRT convergence, and RRT*-like optimization mechanisms are being used with PRM to enhance path quality. Concurrently, their cores are undergoing continuous intellectualization, with traditional random sampling gradually being augmented by learning-driven sampling strategies, adaptive parameter tuning, and end-to-end planning reasoning.

Overall, the sampling-based planning system is advancing towards the integration of structure-guided and random exploration, adaptive sampling processes, and executable planning outcomes. In the future, key directions for ongoing breakthroughs in this field will include deep integration with semantic environment understanding and dynamic obstacle prediction, as well as achieving safe, smooth, and real-time high-quality path planning in open, dynamic, and uncertain environments.

2.4. Meta-Heuristic Optimization Algorithms

Meta-heuristic optimization algorithms (such as Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Chemical Reaction Optimization (CRO), cuckoo search (CS), firefly algorithm (FA), genetic algorithm (GA), particle swarm optimization (PSO), Simulated Annealing (SA), etc.) constitute a class of general-purpose optimization methods distinct from traditional deterministic search algorithms. As the name implies, meta-heuristic optimization is an abstract algorithm that has been proven effective—as opposed to a deterministic search algorithm that has a guaranteed end result. The core objective of these algorithms is not to solve problems directly but to create a global exploratory and local exploitative iterative searching framework by simulating macroscopic mechanisms found in nature, like collective biological behavior, physical phenomena, or evolution. This framework will efficiently find near-optimal solutions in complex solution spaces. Algorithms of this category do not require exact gradient information of the problem and are relaxed on continuity, convexity, etc., of the objective function. They are particularly suited to complex optimization problems like path planning that are high-dimensional, nonlinear, multi-modal, and difficult to describe using traditional mathematical programming methods. The upcoming sections will elaborate on various representative algorithms of this kind.

2.4.1. Genetic Algorithm

The genetic algorithm (GA) is a global optimization algorithm that simulates the natural biological evolutionary process. It was first proposed by John Holland in 1975 [58]. Its core idea is to iteratively search for optimal solutions within the solution space by simulating genetic operations such as "selection, crossover, and mutation." In the field of UGV path planning, GA is widely adopted due to its good adaptability to complex, nonlinear, and multi-constrained problems. It can handle path optimization in both static and dynamic environments, demonstrating strong robustness and generality, especially in global path planning where environmental information is fully or partially known.

Despite this, traditional genetic algorithms still have some limitations in UGV path planning. They usually have a reduced convergence rate or can only converge with the local optimal solution because no suitable parameter is selected. In addition, the quality of the initial population also affects the performance of the algorithm. To deal with these shortcomings, subsequent studies have made various improvement proposals. For the complexity of path planning for reconfigurable robots with multi-

ple morphologies, as shown in Figure 6, Cheng et al. [59] designed a path planning framework based on the Non-dominated Sorting Genetic Algorithm II (NSGA-II) for the hinged-Tetromino reconfigurable robot (hTetro), termed hTetro-GA. They proposed four fitness functions encompassing goal reachability, time consumption, path smoothness, and safety, achieving multi-objective Pareto-optimal path search within complex-obstacle layouts. Targeting their insufficient local search capability and premature convergence, Feng et al. [60] proposed a Hybrid Adaptive Genetic Algorithm (HAGA). By introducing knowledge-based operators for initial path pre-optimization and adopting a hierarchical optimization strategy (dividing the population into a TOP layer and a regular layer, with the former applying an adaptive 2-opt* operator and the latter using an adaptive crossover/mutation mechanism incorporating Sigmoid/TANH functions), the algorithm significantly enhances its local search capability and convergence speed. Addressing path safety and multi-task adaptability in dynamic environments, the study also constructed a functional model linking task risk levels and road factors. Weight adjustment within this model enables safe path planning for different risk-level tasks, improving the algorithm's generalization capability in complex road environments. To tackle the defects of traditional GA being prone to local optima and slow convergence in dynamic obstacle environments, Changan et al. [61] proposed incorporating the hill-climbing method to enhance the performance of the mutation operation to overcome the shortcomings of the traditional GA, which is easily stuck in local optima and converges slowly in dynamic obstacle environments.

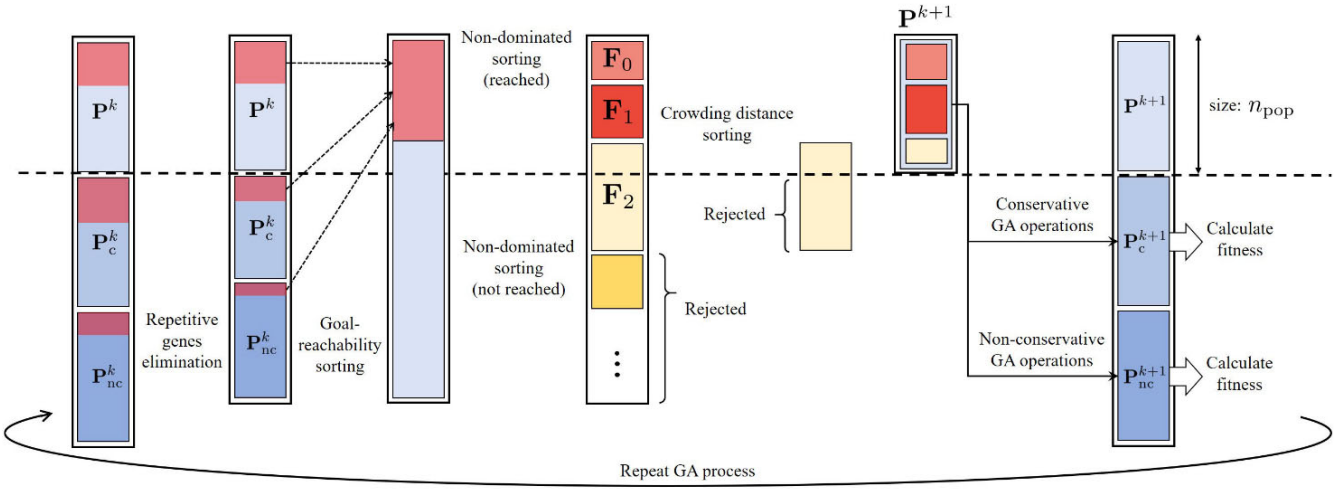


Figure 6. Applications of NSGA-II in hTetro-GA [59] (Reprinted from Ref. [59], with permission).

In short, GAs in UGV path planning have evolved from optimal evaluation functions or operators to hybrid adaptive systems incorporating multiple intelligent strategies. While this evolution greatly improves the performance of algorithms, the integration of multiple intelligent strategies sometimes leads to increased computational complexity, which may become a constraint for small, resource-constrained robots. There have also been recent studies to improve the adaptability of algorithms in dynamic and complex environments through structural innovation or the fusion of different mechanisms. However, structural innovations often require tedious parameter calibrations to balance adaptability and computational efficiency. At the same time, in order to better transition from simulations to reality, joint validation with real robots is emphasized—although validation is critical, it is often time-consuming and expensive due to the need for repeated physical testing. This development provides the necessary technical support for a smarter, more robust, and more efficient autonomous navigation system.

2.4.2. Ant Colony Optimization

Ant Colony Optimization (ACO) is a swarm intelligence optimization algorithm that mimics the foraging behavior of ants in nature. It was first proposed by the Italian scholar Marco Dorigo in his doctoral thesis in 1992 [62]. The algorithm's inspiration stems from the collective behavior of real ants, which communicate indirectly by depositing pheromones to collaboratively discover the shortest path between their nest and a food source. In ACO, artificial "ants" construct candidate solutions within the solution space. The probability of path selection is jointly determined by pheromone concentration and heuristic information. The pheromone is subsequently updated based on the quality of the solutions, forming a positive feedback mechanism that continuously strengthens the pheromone on superior paths, ultimately guiding the entire ant colony to converge on an optimal or near-optimal solution. Due to its characteristics of distributed parallelism, self-organization, positive feedback, and strong robustness, ACO has been successfully applied to various combinatorial optimization fields, including the Traveling Salesman Problem (TSP), vehicle routing, scheduling, and robot path planning.

Although ACO is highly successful in various problems, its limitations prevent it from being useful in complex dynamic environments. At the beginning, all ants have the same amount of pheromones, so in early stages they explore in random directions. As a result, the convergence speed is slow. To address this issue, Liu et al. [33] adopted an adaptive two-layer ACO structure for this issue. Through the application of a non-uniform initial pheromone distribution that took into account the distance between nodes and an adaptively varying heuristic function, a significant enhancement in the direction and efficiency of the initial search was achieved. In subsequent generations, the pheromone could concentrate too strongly on a handful of paths that were found earlier in the process, which could lead to the algorithm being stuck in local optima. To overcome this premature convergence problem, Wang et al. [63] introduced an iterative adjustment function into the global pheromone update rule. This function applies differentiated rewards and penalties to the best and worst paths in each iteration and adds random perturbations, effectively helping the ant colony escape local optima. Finally, traditional ACO is sensitive to parameter settings and suffers from insufficient global exploration capability and path diversity when facing complex terrains, like dense obstacles and narrow passages. To this end, Lin [64] proposed a novel idea of incorporating a negative feedback mechanism into the pheromone update process to reduce the selection of low-quality paths. The crossover and mutation operability of the Particle Differential Optimization (PDO) algorithm was also integrated into the model; this causes the path sequences to introduce global perturbations at periodic intervals, thereby significantly enhancing the algorithm's robustness and solution quality in a complex static environment. Similarly, for the problem of path node blockages in warehouse environments, Wu et al. [65] introduced a blockage factor into the heuristic function as negative feedback information. They designed a two-layer planning model combining dynamic priority adjustment and a real-time conflict-resolution strategy, further improving ACO's adaptability and overall operational efficiency in dynamic multi-robot systems.

In summary, the ACO algorithm, as a classic swarm intelligence optimization method, solves complex optimization problems by simulating the positive feedback collaboration mechanism of ant colonies. Later studies have substantially advanced the speed of convergence and capacity for the global search of ACO, as well as adaptation to the environment, by optimizing the initialization, state transition rule, pheromone update, and fusion with other algorithms. Currently, the improvement and application of ACO have progressed from single-algorithm optimization to stages of multi-layer collabora-

tion, adaptive regulation, and multi-scenario generalization. Future work will continue to advance towards parameter self-adaptive learning, dynamic environment adaptation, high-dimensional/multi-objective planning, and cross-platform deployment, further expanding its practical value in fields such as autonomous robots, intelligent logistics, and network optimization.

2.4.3. Firefly Algorithm

The firefly algorithm (FA) is a swarm intelligence meta-heuristic optimization algorithm that simulates the flashing behavior (e.g., mating and foraging) of natural fireflies. It was first proposed by Yang in 2009 [66]. The algorithm searches based on the mechanism where fireflies are mutually attracted by their light intensity: individuals with a lower light intensity move toward those with a higher intensity, and attractiveness is inversely proportional to distance.

Due to its simple structure, ease of implementation, and strong global search capability, FA has been widely utilized in continuous optimization tasks, including advanced areas like robot path planning and trajectory optimization. In FA, operator generation for path planning is used to find optimal collision-free paths from a start to a goal. The fitness function usually first combines the path length and collision cost and then optimizes the path gradually with the collaboration of the fireflies.

However, applying FA to real-world robot navigation still faces numerous challenges, and subsequent research has conducted targeted optimizations around these core limitations. To address the issues of traditional FA being stuck in local optima and converging slowly in unknown environments, Chaudhary et al. [67] proposed the FAStep-Kinematic hybrid algorithm. This algorithm combines FA with robot kinematic equations and innovatively introduces an "on-demand activation" mechanism where FA is activated for obstacle-avoidance planning only when obstacles are detected. Sometimes, efficient kinematic equations make the robot drive directly to the goal. This greatly limits invalid searches and greatly increases convergence speed and real-time performance. Li et al. [68] proposed the Self-Adaptive Population Size Firefly Algorithm (SPSFA) to avoid the problem of the fixed size population in FA, which leads to a lack of diversity in early iterations and too many repetitive individuals in later iterations, giving low computational efficiency. This algorithm can dynamically adjust the population size through the collision degree between paths and obstacles and, in the process of deleting individuals, first removes the infeasible path, balancing exploration capability and cost. In order to improve the exploration efficiency and strategy learning ability of FA in completely unknown, complex environments, Sun et al. [69] designed the FADQN algorithm, which deeply integrates FA with DQN. The authors utilized FA's action output as heuristic guidance for the agent's initial exploration. They implemented an adaptive ε-greedy strategy and a prioritized experience replay mechanism. This speeds up the agent's learning of the knowledge and policy convergence in an unknown environment, improving the success rate and stability of path planning.

Overall, the firefly algorithm for robot path planning is gradually moving toward adaptive attraction models, dynamic population tuning, and a combination with other intelligent methods. Future efforts will focus on integrating FA with environmental perception and deep reinforcement learning, as well as obtaining more efficient and stable paths in dynamic and unstructured environments.

2.4.4. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based intelligent optimization algorithm proposed by Kennedy and Eberhart in 1995 [70]. Its inspiration stems from the social cooperative mechanism observed in bird flock foraging behavior (Figure 7). In UGV path planning, PSO searches for a collision-free optimal path from the start to the goal by simulating the flight and collaboration of particles within the solution space, continuously updating both individual and global best positions. Due to its simple structure, few parameters, and fast convergence speed, PSO has been widely applied to path planning problems in both static and dynamic environments, making it a significant algorithm in the field of robot autonomous navigation.

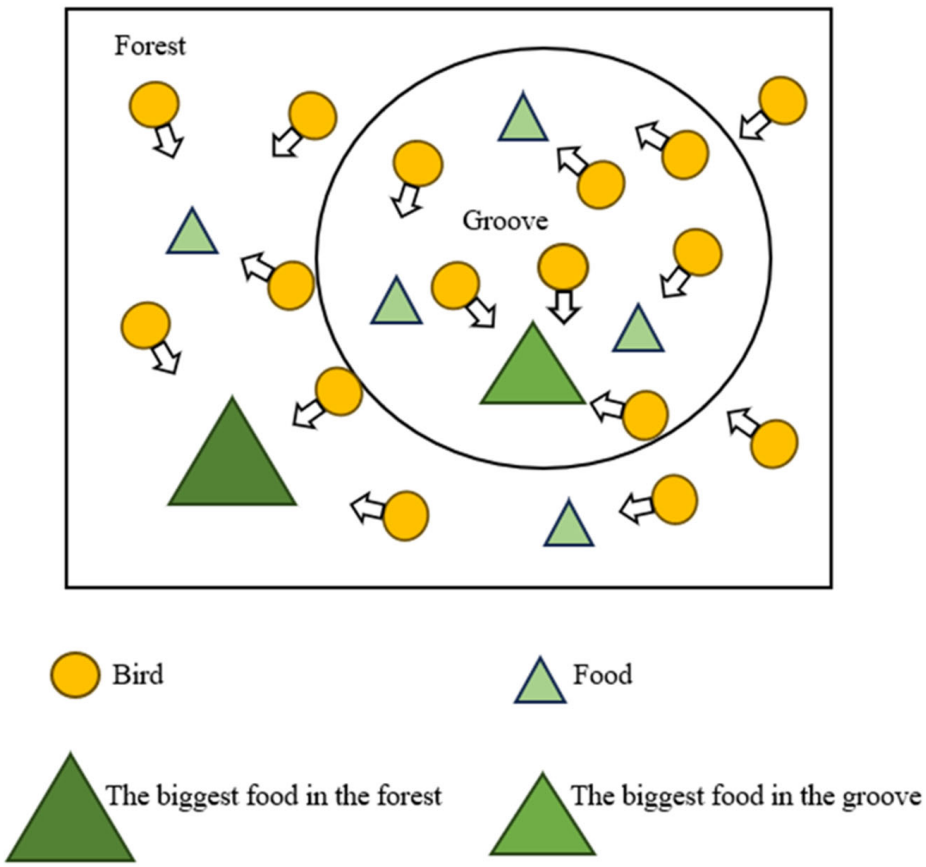


Figure 7. Schematic diagram of the PSO algorithm simulating bird foraging (Created by the authors).

The traditional particle swarm optimization algorithm shows promise when applied to path planning of UGVs. However, it also suffers from several core limitations. Firstly, it is prone to becoming trapped in local optima, resulting in planned paths that are not globally shortest. Secondly, its search efficiency is low, and convergence is slow in complex environments. Finally, its adaptability is weak in dynamic or high-dimensional environments. To solve these problems, several effective improvement strategies, through deep integration with other methods, have been proposed recently. As an example, Zheng et al. [71] proposed the APF-PSO (apfrPSO) algorithm to counter the local optima issue using the artificial potential field mechanism. In this algorithm, a virtual force field is used to direct the particles towards the target region, which makes it effective at both global exploration and local exploitation capabilities. In response to the inefficiencies caused by low search and slow convergence, Wang et al. [72] creatively integrated GANs with PSO to propose the LPSO algorithm. The creation of prospect regions serves as heuristic information, which greatly reduces the invalid search space and significantly speeds up the planning and

quality of paths. Furthermore, to enhance the algorithm's robustness in complex static environments and improve the initial population quality, Huang et al. [73] presented a modified APSO algorithm fused with an A* algorithm to increase the robustness of the algorithm in complex static environments as well as the quality of the initial population. The approach employs the A* algorithm to quickly obtain high-quality initial solutions for PSO. In addition, random inertia weight and random opposition-based learning strategies are used to further increase convergence speed and improve the final path's smoothness and practicality.

To summarize, PSO, as a classical swarm intelligence optimization algorithm, is of great significance for mobile robot path planning. Current research trends indicate that integrating traditional path planning algorithms, machine learning-assisted strategies, and adaptive parameter designs will significantly overcome these limitations and optimize the planning performance in complex dynamic environments. In the future, further optimization of PSO could focus on areas such as multi-objective collaborative optimization, real-time dynamic obstacle handling, and simulation-to-real (Sim2Real) transfer learning, promoting its ability to achieve more efficient and reliable autonomous decision-making in practical robot navigation systems.

2.4.5. Simulated Annealing Algorithm

The Simulated Annealing (SA) algorithm is a stochastic optimization algorithm inspired by the physical annealing process of solids. Its core idea originates from the Metropolis criterion proposed by Metropolis et al. in 1953 [74], which provides the mathematical foundation for accepting "inferior" states with a certain probability to explore the solution space. The SA algorithm can search the global solution space heuristically and simulate the process of temperature decrease of the states and the stabilization of the system like in physical annealing during robot path planning. Multi-robot task assignment and area coverage sequence optimization among other combinatorial optimization problems are widely solved with it. In theory, under proper parameter settings, this probabilistic acceptance mechanism allows SA to escape local optima. Such an optimization strategy plays an important role in environment modeling and path generation for typical combinatorial problems, including region coverage order optimization and multi-robot task assignment.

Nonetheless, in actual path planning applications for UGVs, most particularly in dynamic, unstructured ground environments with real-time constraints, SA suffers from limitations. Because of design and environmental restrictions, such as an imperfect cooling schedule, limited allowed iteration time, uneven terrain, and dynamic obstacles, the algorithm remains helpless to avoid local optimum and converges too slowly in real-world deployments. To counter these practical difficulties, later studies have been carried out that aim to fix its main shortcomings. Shi et al. [75] proposed a Monotonically Heated Simulated Annealing (MHSA) algorithm. The algorithm's ability to escape from local extremes can be improved by monitoring when the solution is stagnantly unchanging in its neighborhood for some time and increasing the temperature suitably. This has achieved a coverage path planning for cleaning robots with a repetition rate of not more than 5% and full coverage of 100%. Addressing the coupled optimization challenge of task allocation and path sequencing for SA in multi-robot systems, Shi et al. [76] designed encoding and decoding mechanisms based on Boolean specifications. They encoded the robots' task areas and destination areas as individuals and employed neighborhood search strategies such as random exchange, 2-opt, and re-insertion. This significantly reduced the total travel distance and computation time while satisfying high-level logical constraints. To address the issue that paths generated by SA in dynamic environments might not comply with

robot kinematic constraints, Gao et al. [77] proposed a Constrained Simulated Annealing–Enhanced Artificial Potential Field (CSA-APF) algorithm. By introducing angle and safe distance constraints into SA, the generated virtual target points better align with the actual motion characteristics of the robot. By this measure, oscillations and local stagnation are effectively avoided during local path planning, enhancing the smoothness and safety of motions during dynamic obstacle avoidance.

In summary, the research on the Simulated Annealing algorithm in robot path planning has evolved from a global optimization framework belonging to the Metropolis criterion to an enhanced optimization algorithm that incorporates the specific features of the problem and constraints on the motions. Current trends are more towards the improvement of the algorithm's practicality, convergence speed and the path quality in dynamic multi-obstacle environments through the hybrid strategy embedding associated with ant colonies or potential fields, constraint-embedding kinematics and safety, adaptive mechanism–dynamic temperature and so on. In the future, when further validated in practical settings and complemented with cross-layer optimization methods, SA and its derivatives can provide greater assistance for real-time decision-making and the scheduling of complicated tasks in autonomous systems.

2.4.6. Cuckoo Search Algorithm

The cuckoo search (CS) algorithm is a swarm intelligence optimization algorithm inspired by the parasitic brood-rearing behavior of cuckoos in nature. It was formally proposed by Yang and Deb in 2009 [78]. The main mechanism of the cuckoo search algorithm mimics a cuckoo searching for host bird nests to lay its eggs. It uses Lévy flights for global exploration and discards poor-quality solutions with a certain probability to simulate host birds discovering and abandoning foreign eggs. The algorithm was quickly applied to optimization problems due to its simple structure, few parameters and ease of implementation. In robot path planning, the CS algorithm searches for a near-optimal collision-free path in an environment which has many obstacles, in which the path is represented as sequence of co-ordinates or a sequence of nodes. The CS algorithm framework constructs a fitness function based on the path length, smoothness, safety, etc. This shows that bio-inspired models can provide engineering solutions.

Although the basic CS framework is concise, its inherent limitations gradually become apparent when tackling complex path planning problems that are high-dimensional and multi-modal. Subsequent research has conducted extensive and in-depth improvements focusing on these core weaknesses. To address the issue of premature convergence caused by strong randomness in algorithm initialization, Min et al. [79] introduced tent chaotic mapping and a random migration strategy to enhance the uniformity and diversity of the initial population distribution. An adaptive adjustment formula of step size and elimination probability was also designed, and the Sine Cosine Algorithm (SCA) was applied for local exploitation in later iterations, which contributed to the improvement of the convergence accuracy and speed of the proposed algorithm. To overcome the drawbacks of Lévy flights, such as direction blindness and low search efficiency, Alireza et al. [80] designed a non-uniform immigration coefficient based on the H-spread measure in their improved mutated cuckoo optimization algorithm (MCOA). This dynamically balanced exploration and exploitation, effectively handling local minima in complex geometric environments. In a study by Chen and Wang [81], the basic CS algorithm was shown to be able to directly apply and work effectively in special structured instances for industrial welding robots, showing it to be useful for deterministic environments. On top of that, as the traditional point-to-point path planning problem has been well-studied for single robots, more complex multi-agent collaborations have been considered. A hybrid algorithm

integrating a Modified Cuckoo Search (MCS), SCA, and PSO, as proposed by Sahu et al. [82], successfully realized collision-free and smooth path planning for the case of multiple robots jointly transporting an object in dynamic and static environments. By synthesizing many strategies and designing a multi-objective function containing inter-robot cooperation constraints, this was achieved. The validation of its effectiveness and robustness took place on a real robot platform.

To summarize, it can be said that the research on the cuckoo search algorithm in the case of robot path planning has moved from a proof-of-concept to a multi-dimensional hybrid improvement phase targeting choke points. Current research trends indicate that its original limitations can be effectively overcome through hybrid strategies, adaptive mechanisms, and structural innovations. These improvements have not only significantly enhanced the algorithm's planning accuracy, convergence speed, and robustness in static and dynamic environments but also promoted its application transition from single-robot navigation to multi-robot collaboration and from theoretical simulations to real physical platforms. In the future, CS and its variants will still have broad exploration space in reducing computational complexity to meet real-time requirements and enhancing adaptive capabilities in completely unknown, unstructured environments.

2.4.7. Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC) algorithm is a swarm intelligence optimization algorithm that simulates the foraging behavior of honey bees. It was first proposed by Karaboga [83] and was initially used to solve multivariable function optimization problems. The algorithm imitates the mechanism of task division and collaboration between employed, onlooker and scout bees in a colony of bees. This is when sharing updates about where to find new nectar sources guides the group search towards better solution areas. The ABC algorithm has been widely applied in scheduling optimization, neural network training, and cluster analysis because of its advantages, such as strong global exploration capability, fast convergence in well-structured static environments, and simple parameter settings. This method shows good performances in mobile robot path planning, especially in cases with single-objective optimization or multi-objective optimization problems in static environments of low complexity—when the solution space is relatively regular and obstacles are sparse.

Despite its numerous advantages, the ABC algorithm still has some limitations in practical UGV path planning scenarios (e.g., dynamic or high-complexity static environments). First, the initial population quality depends on random generation, which can reduce search efficiency when obstacles are irregularly distributed in unstructured ground environments. Second, its local search mechanism is relatively simple: the algorithm tends to converge quickly to promising regions early on, but as population diversity decreases, it becomes prone to local optima or evolutionary stagnation, especially in complex solution spaces with multiple local extrema. Third, its adaptability and convergence performance in multi-objective or dynamic environments (e.g., moving obstacles) still require improvements, as the traditional ABC lacks effective strategies to adjust search behaviors dynamically in response to environmental changes. In recent years, numerous strategies have been suggested to improve upon these weaknesses. For example, to tackle the problem of poor initial solution quality, Yu et al. [84] proposed a hybrid initialization strategy. This combines the RRT algorithm with a space-partitioning method to generate a high-quality initial population. It enhances local exploitation by employing a variable neighborhood local search strategy and incorporates a global search mechanism based on Pareto front information, which efficiently prevents evolutionary stagnation and improves the performance of the algorithm for static multi-objective

path planning. Kumar and Sikander [85], noticing that the existing ABC algorithm and evolutionary programming (EP) optimization algorithm did not take into account the distance of obstacles, proposed an improved ABC-EP algorithm. Choosing the best food point concurrently takes into account its distance to the goal and the nearest obstacle, which significantly reduces the path length and search time, making the path safer. Li et al. [86] proposed the IACO-IABC hybrid algorithm from the perspective of algorithm fusion. By improving the heuristic mechanism of the ACO; introducing the contraction encircling and spiral updating strategy of the ABC; and incorporating the path optimization operation, the number of path turning points was significantly reduced, leaving the convergence speed of the algorithm and the smoothness of the path improved.

To sum up, as an efficient swarm intelligence optimization algorithm, ABC has been widely investigated and deeply applied in robot path planning. Recent studies not only offer many new improvement methods addressing one of the most crucial limitations, initial population generation, local search, multi-objective improvements, etc., but also increase its applicability and robustness in complex dynamic environments via fusion with other intelligent algorithms (such as ACO, EP, and RRT). In the future, continuous optimization of the algorithm structure, and the continuous expansion of practical application scenarios, ABC and improved variants are expected to play a greater role in areas like autonomous navigation, multi-robot collaboration, and dynamic obstacle avoidance and push robot path planning technology towards a more efficient and intelligent development.

2.4.8. Chemical Reaction Optimization

CRO is a meta-heuristic optimization algorithm that was inspired by a chemical reaction. Alatas [87] first systematically expounded the core ideas. The algorithm mimics important processes in chemical reactions, including molecular collisions, decomposition, synthesis, and energy transfer. This method encodes several solutions of the optimization problem as molecules, represents the objective function value that we want to optimize with the molecule's potential energy, and represents the activity level of the proposed solution using kinetic energy. It examines the solution space through four fundamental reaction operators: ineffective collision, intermolecular collision, decomposition, and synthesis. Over the years, the CRO algorithm has shown great potential for dynamic optimization problems, especially path planning. Due to its good global search capability and robust characteristics, the CRO algorithm finds wide applications in many important areas, like task scheduling and others.

Although CRO exhibits a strong global search ability, it does have certain limitations. The speed of convergence is slower, and solution accuracy is not high when dealing with continuous optimization problems. The second drawback of the framework is its difficulty in balancing the global exploration and local exploitation task, which makes it prone to either early convergence or low search efficiency. Thirdly, the parameter setting relies on experience and is not sufficiently adaptable to the issue. Researchers in the past have proposed various strategies to enhance the performance of CRO. To tackle the imbalance between global and local search, Yang et al. [88] proposed the Dual-Container Chemical Reaction Optimization (DCCRO) algorithm. By establishing a main container and a subsidiary container focusing on global exploration and local exploitation, respectively, and designing a molecular exchange mechanism to dynamically balance these two search capabilities, the algorithm improves the convergence speed and path quality in path planning. To further enhance CRO's performance in continuous optimization, Zhang et al. [89] proposed an Advanced Adaptive Chemical Reaction Optimization (AACRO) algorithm based on balancing

local and global search. This algorithm combines the advantages of Adaptive CRO (ACRO) and PSO, introduces an adaptive step-size mechanism and a "completion" operator and designs parameters to control the proportion of global to local search, significantly improving convergence accuracy and speed.

In summary, as an incoming bio-inspired optimization algorithm, CRO is significantly evolving both theoretically and in application extensions. Currently, research trends are aimed at getting over its inherent problems of slow convergence and difficulty in balancing exploration through structural innovations like container mechanisms, hybrid strategies, adaptive parameter designs, and fusions with other intelligent algorithms. In the future, the CRO algorithm is expected to find wider and deeper applications in more complex optimization scenarios, such as dynamic path planning, multi-objective optimization, and real-time scheduling. Further integration with engineering practice will promote its development from theory towards practical and systematic implementation.

2.4.9. Summary

Table 4 presents representative meta-heuristic optimization algorithms and their improvements. In recent years, meta-heuristic optimization algorithms used in UGV path planning have changed a lot. Early algorithms mostly simulated a single biological behavior, like the foraging behavior of ants or fireflies, which often had drawbacks, such as slow convergence, easy trapping in local optima, and poor adaptability to complex environments. Now, most improved algorithms focus on solving these practical problems, rather than just pursuing theoretical perfection. These algorithms are mainly optimized in three aspects: first, combining with traditional planning algorithms or machine learning to make up for the deficiencies of a single algorithm; second, dynamically adjusting parameters and population sizes according to the environment to avoid premature convergence or slow search speeds; and third, adding constraints such as UGV kinematics and a safety distance to ensure that the planned path can actually be executed by UGVs in real scenarios. Among them, GA has developed from a basic "selection–crossover–mutation" framework into a hierarchical adaptive system incorporating elements like PSO and attention mechanisms. ACO has enhanced its search efficiency and diversity in complex terrains through initialization optimization, negative feedback mechanisms, and fusion with methods like differential evolution [90]. FA has achieved "on-demand planning" and adaptive search balance by integrating with kinematic equations and deep reinforcement learning. PSO has effectively overcome the limitations of premature convergence and low search efficiency by employing guidance strategies such as artificial potential fields and generative adversarial networks. SA has improved its global search capability and real-time performance in dynamic environments through methods like monotonic heating, constraint embedding, and hybridization with potential field methods. CS has increased its convergence accuracy in complex geometric environments using chaotic mapping, adaptive step sizes, and hybrid strategies. The ABC algorithm has enhanced its multi-objective planning capability via hybrid initialization, a variable neighborhood search, and fusion with ACO and evolutionary programming. CRO has achieved a better balance between global exploration and local exploitation through dual-container mechanisms, adaptive step sizes, and hybridization with PSO [91].

Table 4. Representative meta-heuristic optimization algorithms: key improvements and evaluation metrics.

Algorithm	Article	Year	Improvement	Environment	Objective	Evaluation
GA	[92]	2025	Knowledge-guided genetic operators, new path representation and evaluation method.	Static and Dynamic	Single	Simulation + Real Robot
ACO	[90]	2023	Multi-step strategy and terminal distance index replace pheromone to accelerate convergence.	Static	Single	Simulation
FA	[93]	2024	Chaotic mapping and courtship learning strategy improve global search and convergence.	Static	Single	Simulation
	[94]	2025	Integrates DWA to enhance dynamic obstacle avoidance.	Dynamic	Single	Simulation + Real Robot
PSO	[95]	2018	Improved PSO with swap operator/sequence and enhanced GA selection for faster convergence.	Static	Single	Simulation
SA	[96]	2023	Improved SA with initial path selection and deletion operation for reduced computation and dynamic adaptability.	Dynamic	Single	Simulation
CS	[97]	2021	Enhanced cuckoo search with genetic operators, 2-opt, Metropolis criterion, and local search methods.	Static	Single	Simulation
ABC	[98]	2024	Enhanced ABC with TLBO learning strategy for better exploitation; proposed a meta-heuristic implementation method for multi-robot path planning.	Static	Multi-objective	Simulation
CRO	[91]	2021	Redesigned CRO operators and introduced two repair operators for path shortening and smoothing.	Static	Single	Simulation

In summary, meta-heuristic optimization algorithms are advancing towards hybrid intelligence (fusion of multiple strategies), self-adaptation (dynamic adjustment of parameters and populations), scenario generalization (from static to dynamic and single-robot to multi-robot systems), and practical applications (validation with real robots). In the future, key steps for these methods to achieve broader practical applications in autonomous navigation systems will involve further reducing computational complexity, enhancing real-time adaptive capabilities in completely unknown and unstructured environments, and achieving deep integration with modules such as prediction and communication.

2.5. AI-Based Algorithms

New categories of path planning algorithms like deep reinforcement learning (DRL) and imitation learning (IL) based on AI will not be included in the traditional model-based and search-based planning. Their underlying concept does not depend on accurate modeling of the environment or explicit design of search strategies. Rather, they learn a mapping policy from states to actions or paths in an end-to-end manner, driven by interaction data between agent and environment, which adapts to highly dynamic, uncertain or complex decision-making environments that are hard to model. The key advancements and common techniques in this area will be elaborated in the following sections.

2.5.1. Deep Reinforcement Learning

Deep reinforcement learning (DRL) is an artificial intelligence (AI) method that combines the representational power of deep neural networks with the decision-making framework of reinforcement learning. Its core lies in the agent continuously interacting with the environment and receiving reward feedback to iteratively optimize its behavioral policy, aiming to maximize long-term cumulative return. In the field of unmanned ground vehicles, DRL has become a key technological paradigm for solving path planning and obstacle-avoidance problems in dynamic, unknown environments due to its powerful ability to learn complex navigation strategies directly from raw sensor data (e.g., LiDAR images). This technology enables end-to-end autonomous decision-making, from environment perception to motion control [99].

Nevertheless, there are still many obstacles to applying DRL to real-world robot navigation, and later studies have optimized around the core limitations. To tackle the issue of traditional experience replay methods failing to capture the dependencies in crucial state sequences during environmental shifts, which results in less efficient learning, Gao et al. [100] proposed a mechanism aided by Transformers. By modeling interaction trajectories following sequences by self-attention, it encourages emphasis on learning and sampling useful obstacle avoidance and turning experience aids in stabilizing training and improving the performance of the policy. In response to the insufficiency of a single policy network in hybrid action spaces, which may result in parameter interference and imprecise control, the same research proposed a multi-branch parameterized policy network. It allocates independent sub-networks to yield continuous control parameters for various distinct and discrete semantic actions, such as "move forward," "turn left," and "turn right," thus improving the expressive power and deployment efficiency of the policy. To mitigate the issue of poor generalization, where models must relearn from the beginning after being trained under one environment and ending up in another, Kumaar and Kochuvila [101] developed a β-decay transfer learning algorithm. As a result, robots will have the ability to make use of the topological knowledge they acquired in old environments to quickly adapt to new or modified environmental layouts. This achieves a multiplicative effect on the speed of convergence and leads to lifelong learning. In order to improve the quality of the path and the convergence of the algorithm, Bai et al. [102] improved the DDQN algorithm by designing a composite reward function that incorporates heuristic distance information and adopted an adaptive mechanism to adjust the network update rate dynamically. This not only speeds up the algorithm's convergence toward the target but also ensures the safety of the planned path and the smoothness of the UGV's motion in practical operations. Additionally, addressing the problems of traditional DRL algorithms being prone to overfitting and having low exploration efficiency in complex scenarios, Yang and Liu [103] proposed an improved algorithm named Deep Reinforcement Learning Algorithm for Path Planning (DRL-PP). By introducing a Multi-Layer Perceptron (MLP) to optimize the network structure and designing a fine-grained cubic reward function based on target information, it more delicately distinguishes the value of different actions, effectively accelerating the convergence process and improving the final path planning outcome.

The application of DRL, which was previously only used in algorithms, is being optimized and integrated into systems used for robot path planning, as the literature suggests. Recent developments emphasize bolstering the performance of algorithms and the efficiency of their infrastructure for dynamic environments via structural innovations (e.g., attention mechanisms and network decoupling), enhancing system adaptability to unknown changes through continual learning mechanisms and continuously closing theorization, engineering and simulation-to-reality gaps through refining reward engineering

and progressive simulation-to-reality transitions. This drives the development of robust, reliable, and efficient autonomous navigation systems and brings this to reality.

2.5.2. Imitation Learning

Imitation learning (IL) enables an agent to acquire a decision-making policy by learning from demonstration data provided by an expert (e.g., state-action sequences), thereby avoiding the difficulty of manually designing complex reward functions. The pioneering application of imitation learning in robot path planning can be traced back to the seminal work published by Pomerleau et al. in 1989 [104]. This study first demonstrated the feasibility of using a simple three-layered feedforward neural network to achieve end-to-end path tracking for autonomous vehicles. This task is achieved by the neural network mimicking the steering control data of human drivers from camera images while driving on real roads. It demonstrated that neural networks can learn to map from raw sensory input (image pixels) to driving policies without engineering features by hand or programming rules explicitly, thus offering an important conceptual and practical foundation for subsequent research on learning-based autonomous navigation.

Nonetheless, as application scenarios evolved from simple road following to complex dynamic multi-agent environments, the limitations of traditional imitation learning became increasingly apparent, prompting multi-faceted algorithmic optimizations. To begin with, expert data is expensive and scarce, compromising the feasibility of algorithms. In order to surmount this limitation, Rizk et al. [105] propose a sketch-guided imitation learning framework that enables the user to provide demonstrations by drawing a path on the screen using sketches. This converts expensive sensor recording or teleoperation to low-cost, intelligible human–computer interaction, greatly lowering the application threshold. Besides the above issues, the compounding error and distributional shift in methods like behavioral cloning leads to performance deterioration of the policy when deployed in the real world. To solve this problem systematically, Yang et al. [106] proposed Mixed Generative Adversarial Imitation Learning (MixGAIL) and designed a special optimizer for it to jointly resolve the two challenges of compounding errors and high sample complexity, thereby improving robustness and data efficiency. In intricate situations such as densely populated multi-robot scenarios without communication, policy networks are unable to depict long-distance dependencies and global collaboration structures. In response, Chen et al. [107] first introduced the Transformer architecture into the policy network. Using its powerful sequence modeling and feature extraction capabilities, this allows robots to infer hidden cooperative relationships from local observations, resulting in a significant increase in path planning success rates in dense environments without communication. Also, classical imitation learning approaches do not offer convergence and safety guarantees in safety-critical situations. Policies that rely on data cannot guarantee safe convergence that is asymptotic and to the goal from arbitrary initial states. Because of this, such policies cannot be reliably applied in high-risk applications, like autonomous driving. In line with this, Lai et al. [108] proposed a trajectory-planning framework inspired by Lyapunov theory called the Convergent and Safe Dynamics from Flat Demonstrations (C&S-DFD). This framework successfully fuses the differential flatness property of wheeled vehicles with Lyapunov stability theory; it learns these basic motions from a few optimal demonstrations using Gaussian Mixture Regression and designs an asymmetric quadratic Lyapunov function capable of embedding obstacle information. In light of this, we derived an analytical auxiliary control law by solving a quadratic programming problem that compensates for the learned model's prediction error, ensuring the global asymptotic stability and safety of the whole cryptography system.

The optimization direction of IL for robot path planning is gradually changing—instead of solely relying on ideal expert data, researchers are now focusing on solving core problems in practical deployment. The key research directions include reducing demonstration density through simpler human–computer interaction, improving the robustness and generalization of the policy to avoid performance degradation caused by data distribution shifts, enhancing the policy's reasoning and decision-making abilities (such as multi-agent cooperation), and most importantly, adding appropriate safety and stability guarantees to data-driven policies to ensure safe operation in real-world high-risk scenarios. These directions help move IL from lab experiments to safer, more widespread practical applications.

2.5.3. Summary

Table 5 summarizes representative AI-based path planning algorithms. We have found that in recent years, AI-based path planning algorithms have gradually moved away from the early end-to-end learning models—these models used to require training neural networks in large-scale simulation environments, which was time-consuming and not practical for real deployment. Instead, they are now moving toward task-adaptive frameworks that focus more on generalization, safety, and tight integration with classical planning systems. The core development route of DRL is shifting from proving the ability to solve specific scenarios to creating a robust system of zero-shot generalization. By designing observation spaces that support map compressing and centering capabilities, implementing action masking mechanisms that ensure safety, and utilizing reward functions that ensure scale invariance of value functions, DRL frameworks can make use of different maps, sensors and local tasks without retraining, thus increasing their practicality and reliability in structured static environments [109]. IL, on the other hand, has evolved from merely cloning expert behavioral trajectories into an adaptive module flexibly integrated into classical navigation stacks. Through a two-stage paradigm of "pre-training followed by human demonstration fine-tuning," imitation learning modules can efficiently inject human semantic preferences (e.g., maintaining a certain distance from specific obstacles) between geometry-based global planning and local control, enabling the system to exhibit task-oriented intelligent behavior in dynamic or semantically rich environments [110].

Table 5. Representative AI-based algorithms: key improvements and evaluation metrics.

Algorithm	Article	Year	Improvement	Environment	Objective	Evaluation
DRL	[109]	2025	Proposed a zero-shot CPP framework using DRL with map compression, action masking, and size-invariant rewards for better generalization.	Static	Single	Simulation
IL	[110]	2021	Integrated an IL module into a classical navigation stack to adapt robot behavior via human demonstrations, improving task success.	Dynamic	Single	Simulation

In general, AI-based planning methods are making progress in fusing the generalization and safety in reinforcement learning with the efficiency and teachability present in imitation learning, with the warming of learning processes, with prior knowledge, and with the compatibility of decision results with physical constraints. In the future, a major breakthrough for this field is going to be the combination of DRL's exploratory and adaptive abilities for unknown environments with IL's rapid ability to grasp complex intentions from human demonstrations and their further deep integration with modules like dynamic

obstacle prediction and multi-agent collaborative communication, for safe, smooth, and interpretable autonomous navigation in open-world scenarios.

2.6. Optimal Control Methods

Optimal control algorithms that are also known as Linear Quadratic Regulators (LQRs) and model predictive control (MPC) are classical control algorithms, but they differ from dedicated path planning approaches while complementing them closely. Their central idea is a precise mathematical modeling of the dynamics of the system and the control objectives: by solving a well-defined optimization problem, either online or offline, they compute the optimal sequence of control signals or commands from the current state to a predefined waypoint or trajectory in a feedback or receding horizon manner.

LQR and MPC are not path planners. They do not plan out the geometric path, which is the series of positions that go from start to goal (e.g., in a ground environment). They do, however, optimize the dynamic execution of paths that have been planned by other methods, which could be RRT*, A*, or sampling-based methods [111]. Under the assumption of an accurate model, they provide stable, reliable, and high-performance control strategies for linear or approximately linear systems, ensuring that UGVs follow the pre-planned path smoothly, safely, and in compliance with kinematic/dynamic constraints. By providing complementary solutions, they become an integral part of the full UGV navigation pipeline, linking a planner (the geometric path) and a controller (the dynamic trajectory).

2.6.1. Linear Quadratic Regulator

The Linear Quadratic Regulator (LQR) algorithm is a classic optimal control. The Lyapunov equation approach aims to find a state feedback gain matrix that stabilizes a system. Furthermore, it optimizes performance by solving a quadratic cost function involving state variables and control inputs. The Linear Quadratic Regulator (LQR) was first proposed and systematized by Kalman et al. [112] in the 1960s. T LQR is widely used in robotics, aerospace, process control, and other fields because of its clear structure, ease of implementation, and good robustness. It remains one of the most fundamental and important design methods in modern control theory.

The classical form of the LQR algorithm has the advantages of theoretical rigor and easy implementation; however, it suffers from serious defects when dealing with practical control problems: its performance heavily relies on empirical tuning of the weighting matrices, and there is no systematic parameter optimization method; it has inherent insufficient robustness to system nonlinearities and model uncertainties; and its pure feedback strategy will cause steady-state tracking errors when realizing dynamic trajectory tracking. Recent research on algorithm fusion proposes different innovative solutions to resolve these bottlenecks. In dealing with the problem of parameter tuning, Zou et al. [113] paired a genetic algorithm (GA) with LQR wherein a GA-LQR controller was proposed. The GA-LQR controller optimizes the weighting matrices automatically and bettered the dynamic tracking accuracy of micro-robot groups. In an iterative LQR approach, Zhang et al. [114] enhanced adaptability to nonlinear and nonholonomic constrained systems. By re-linearizing and determining optimal feedback at every stage along the trajectory, it successfully achieved high-precision trajectory tracking of wheeled mobile robots in complex environments. In addition, Zhang et al. [115] developed a PSO-LQR controller to eliminate steady-state errors and improve control robustness at high speeds. The approach utilizes the PSO algorithm for optimizing weights and introduces feedforward compensation based on path curvature, leading to a reduced path tracking error for autonomous vehicles operating at high speeds.

In robot motion control, LQR-related research has evolved from traditional linear optimal control toward a more flexible design that integrates various intelligent methods. Instead of relying on fixed linear models and manual parameter tuning, current studies tend to adopt structured optimization approaches, such as genetic algorithms and particle swarm optimization, to realize adaptive adjustments of the LQR controller, so as to reduce the dependence on accurate modeling and improve robustness against model errors and environmental disturbances. Meanwhile, researchers are also exploring new combinations between the LQR control structure and upper-level planning or decision-making modules, enabling it to better match actual robot motion tasks. With these improvements, LQR is gradually transforming from a static, model-dependent control method into a key closed-loop control module that can handle uncertainties, maintain stability, and achieve high-performance control effects in both simulation environments and real-world robot applications.

2.6.2. Model Predictive Control

Model predictive control (MPC) is a modern regulation control technique that is capable of allowing for new possibilities in automation. The central concept behind this process is to predict the controlled system's behavior over a finite future horizon in each control cycle. By addressing a constrained optimization issue on the fly, it secures a series of ideal forthcoming control inputs and executes the first control input in this series on the system. This procedure is repeated at the next sampling time using the updated state measurements. This forms a "receding horizon" optimization framework. The effectiveness of model predictive control comes from the fact that constraints can be handled naturally and directly, whether they are on states or inputs, while optimizing complex multivariable dynamic performance indices. Early famous examples of its industrial application include Dynamic Matrix Control, proposed by Cutler et al. [116] for asymptotically stable objects, and Model Predictive Heuristic Control, proposed by Richalet et al. [117], which emphasized heuristics and impulse response models. The foundation of MPC in process industries was established by these early works. With the evolution of theory and computational techniques MPC has become a canonical paradigm for solving motion planning and optimal control problems for complex systems such as robots and vehicles.

Yet, the traditional MPC struggles with the key challenges of robot motion planning, namely their strong model dependence, non-smooth dynamic handling difficulty, local optimum issue in complex environments, and heavy computational burden in multi-agent settings. Recent studies proposed powerful optimization solutions by combining new simulation tools, distributed architectures, and global planning to mitigate these limitations. To address the need for an accurate and differentiable model, Pezzato et al. [118] used the GPU-parallel physics simulator IsaacGym as the forward dynamics model for a Model Predictive Path Integral (MPPI) controller. The technique uses parallel sampling of a large number of trajectories, thereby avoiding gradient computation and explicit modeling. Furthermore, it efficiently handles non-smooth dynamics like contact and collision and achieves real-time control for complex contact manipulation tasks. In order to provide scalability and real-time performance for dynamic multi-agent systems, Xin et al. [119] constructed a distributed MPC framework based on a Mixed Logical Dynamical (MLD) model for AGV systems and used the Alternating Direction Method of Multipliers (ADMM) for coordinated decomposition. This allows for the global collision-avoidance issue to be decomposed into local sub-issues which can be solved in parallel, thereby significantly improving the speed and robustness of the system's decision-making. In order to counteract the local optimum problem caused by the lack of a global perspective in complex-obstacle

environments, Li et al. [120] proposed a globally guided MPPI framework. The integration of global planning algorithms, such as A* and RRT, allows for the global path sequence to be utilized by the local MPPI as a time-varying tracking target, giving long-horizon guidance to the sampling-based optimization process. As a result, this largely mitigates planning failures of large-scale complex situations and enhances the navigation success rate and path quality. To solve the issue of coordinating obstacle avoidance in real time and tracking preciseness in dynamic complex environments, Li et al. [121] proposed a hybrid framework combining MPC with APF. When obstacles are detected, APF quickly generates local virtual targets and collision-free paths, with a PID controller rapidly avoiding the obstacle. In safe zones, it resorts to explicit MPC for precise trajectory tracking, thus obtaining reliable tracking and static obstacle avoidance in fenced environments.

To summarize, robot motion planning using MPC is evolving into a new stage. This evolution goes beyond traditional approaches which depend on mathematical models and centralized optimization with guaranteed convergence. Also, the new evolution introduces high-fidelity parallel physics simulations, distributed optimization, and global–local hierarchical fusion to construct better rigidity against environmental infectivity and more reliable control over complex dynamics. These works, utilizing modern computing tools and algorithmic technologies, are broadening the scope of traditional MPC applications. In so doing, they support MPC as a fundamental, practical technology for solving complex decision-making and control problems in real-world robot applications.

2.6.3. Summary

Table 6 lists representative optimal control methods and their applications. In recent years, optimal control methods represented by LQR and MPC have developed from traditional model-based optimal control into practical frameworks better suited for UGV path planning. For LQR, the main trend has shifted from fixed-weight linear control to an integrated design combined with intelligent path planning strategies. As demonstrated in agricultural UGV applications [122], instead of manual tuning, its weighting matrices, Q and R, are now optimized automatically through meta-heuristic algorithms, such as GA and PSO, with objectives including trajectory smoothness and obstacle safety. Combined with iterative linearization and feedforward compensation, LQR can better handle nonholonomic constraints of wheeled UGVs and enhance path-tracking accuracy, gradually becoming a stable control module that works reliably under uncertain environments. For MPC, development has moved beyond reliance on accurate mathematical models toward a structure incorporating high-fidelity simulations, distributed optimization, and hierarchical global–local planning. With GPU-based parallel simulation tools such as IsaacGym, MPC can efficiently model terrain and collision dynamics and adjust local paths in real time. Using distributed algorithms like ADMM and hierarchical planning, it also supports multi-UGV coordination and avoids local minima in cluttered environments.

Overall, the evolutionary direction of optimal control algorithms lies in balancing model accuracy with computational tractability, unifying global optimality with local real-time performance and deeply integrating theoretical frameworks with engineering practice. In the future, we envision that these systems will be expected to tackle an increasingly larger variety of autonomous navigation and manipulation tasks. In particular, a key enabler towards this will be further reducing their reliance on accurate models, improving their real-time decision-making and generalization abilities in highly uncertain and unstructured environments, and achieving complementary advantages with learning-based methods.

Table 6. Representative optimal control methods: key improvements and evaluation metrics.

Algorithm	Article	Year	Improvement	Environment	Objective	Evaluation
LQR	[122]	2024	Uses LQR controller to optimize motion performance of a four-wheeled UGV in agricultural scenarios.	Static	Single	Simulation
MPC	[123]	2025	Proposes the ODMP + ST-TSMC framework for trajectory planning and tracking in dynamic obstacle environments.	Dynamic	Single	Simulation + Real Robot
	[124]	2022	Proposes distributed MPCC for real-time multi-robot motion planning and collision avoidance.	Dynamic	Multi-objective	Simulation

2.7. Other Algorithms

In contrast to emerging data-driven AI methods, traditional path planning algorithms primarily rely on explicit environmental modeling and numerical computation based on mathematical theory. Among them, the level set method (LSM) defines a moving front as the zero level set of a higher-dimensional function surface and drives its evolution by solving partial differential equations, naturally handling topological changes. It is suitable for path search and shape optimization in complex dynamic environments. The fast marching method (FMM) can be seen as its efficient special case. It simulates the one-way propagation of a wavefront at variable speeds, rapidly constructing a global navigation field by solving the arrival time function. It is basically a deterministic shortest-path algorithm in continuous space and is widely used in robot navigation, image analysis and more. These two approaches, which rely on solving partial differential equations numerically, form an independent planning paradigm, quite different from graph-search-based, sampling-based, meta-heuristic optimization, and classical optimal control methods.

2.7.1. Level Set Method

The basic concept of the level set method (LSM) was introduced in the foundational work of Osher and Sethian [125]. This method implicitly represents a moving interface as the zero level set of a higher-dimensional function and simulates its evolution driven by speed related to curvature by solving Hamilton–Jacobi-type partial differential equations. In the field of robot path planning, this mathematical framework has been successfully adapted: the robot's reachable area and obstacle boundaries are treated as the interface to be evolved. By solving the Eikonal equation to quickly compute the distance field, collision-free paths are generated. Due to its innate ability to handle topological changes, high-dimensional spaces, and complex boundaries, this method laid the theoretical foundation for numerous subsequent path planning studies.

The conventional LSM generally exhibits problems of inferior path quality, weak environmental adaptability, insufficient computational efficiency, and great difficulty in the collaboration of multiple agents for practical applications. Subsequent research has systematically optimized these core issues from different dimensions: Zhang et al. [126] significantly improved path smoothness and optimality and achieved flexible control of safe distance through an elastic particle model and a nearest boundary point diffusion method. Li et al. [127] innovatively analogized the problem to thermal conduction topology optimization, combining growth simulations and the concept of temporary safe paths to effectively solve online planning and deadlock avoidance in dynamic unknown environments. Stagg and Peterson [128] extended this framework to the field of multi-agent collaboration. By introducing B-spline parameterization, differential flatness constraints,

and the block coordinate ascent algorithm, they achieved, for the first time, distributed multi-agent level set estimation and path planning under kinematically feasible conditions. These works collectively propelled LSM from a theoretical concept towards an efficient, reliable, and collaborative practical path planning solution.

LSM arises from the elegant mathematical theory of interface evolution, and it has shown strong vitality in the field of robot navigation. More recent studies have incorporated models of elastic mechanics, topology optimization of thermal conduction, strategies of multi-agent collaboration and techniques of parameterization of the path under kinematic constraints into the basic computation of distance fields. They deeply address core application challenges such as path optimality, dynamic adaptability, collaboration efficiency, and motion feasibility. These methods effectively solve key practical problems, including path optimality, dynamic adaptability, multi-robot coordination efficiency, and executable motion feasibility. While preserving the basic ideas of LSM theory, they greatly improve the applicability and robustness of the control framework in complex scenarios. Future research is expected to further integrate LSM theory with modern learning techniques, such as deep learning, to support more sophisticated autonomous decision-making in intelligent systems.

2.7.2. Fast Marching Method

The fast marching method (FMM), proposed by Sethian in his seminal paper in 1996 [129], is a very effective numerical method for solving the Eikonal equation in modeling the monotonic outward expansion of the wavefront (or interface) at a constant speed. The core idea implements a propagation strategy that satisfies an entropy condition. Also, it uses a priority queue (for example, a heap) to update grid points in increasing order of arrival time. Thus, it can quickly compute the shortest arrival time from a source point to every point in space. Owing to its computational efficiency and the guarantee of a unique solution, FMM enjoys a considerable amount of popularity in many applications ranging from image-processing to fluid simulations. Furthermore, a direct theoretical justification of the idea of 'fast marching' is provided in many later optimal motion planning algorithms.

Although FMM is highly effective in computing static distance fields, directly applying it to robot path planning presents intrinsic limitations. FMM's underlying nature is grid-based, and it propagates outward at a single fixed speed. Consequently, this makes it challenging to directly address important issues in planning. Such tasks include dynamic obstacles, global path optimization, real-time replanning, and a multiple-target path search. In order to overcome such limitations, further studies have proposed different targeted optimization algorithms based on the FMM framework: To address the real-time performance and obstacle avoidance needed in dynamic environments, Liu et al. [130] proposed the Prediction-Based Real-Time Bidirectional Fast Marching Tree (P-RT-BFMT*), introducing an event-triggered replanning mechanism based on collision prediction within the fast marching framework, effectively reducing the frequency of local path replanning. To improve the efficiency and quality of global-path convergence, Wu et al. [131] put forward the so-called Secure Tunnel Fast Marching Tree (ST-FMT*). By generating sampling guidance through "secure tunnels", the planning speed and path optimality are significantly improved in narrow and crowded situations. For real-time interaction and multi-query requirements, Silveira et al. [132] proposed the Real-Time Fast Marching Tree (RT-FMT*), which generates local paths synchronously during expansion and dynamically updates the tree root, enabling parallel real-time obstacle avoidance and plan execution. For multi-target path planning problems, the Fast Marching Firework (FMF) method proposed by Giang et al. [133] allows multiple

targets to propagate simultaneously as independent wave sources and continues expanding after intersection, thereby establishing richer inter-target connecting paths in complex-obstacle maps, effectively reducing the total travel cost.

The FMM, an efficient numerical algorithm, and a series of improved algorithms for robot motion planning, such as P-RT-BFMT, ST-FMT, RT-FMT*, and FMF, show a complete development path from the basic wavefront propagation theory to practical application systems. The design of these algorithms is novel, aimed at core issues such as adaptive dynamics, global optimization efficiency, interactive real-time capability, and multi-target expansion, which greatly improve the practicality, robustness, and efficiency of path planning in complex, dynamic and multi-task scenarios. These innovations offer critical technical assistance to enable autonomous robots to navigate autonomously in real-life applications, like warehousing, inspections and servicing. Future tasks may investigate their broader implementation and systemization in high-dimensional space, robot teamwork, and cross-situation transfer.

2.7.3. Summary

Table 7 provides a summary of other representative path planning algorithms, including LSM and FMM. Current research trends in LSM and FMM have moved beyond simply solving partial differential equations numerically and now focus on systematic optimization for complex real-world navigation tasks. Main improvements include enhancing real-time performance and replanning ability in dynamic and unknown environments, raising global optimality and path smoothness while increasing adaptability to narrow passages, extending to multi-agent systems for distributed coordination and collision avoidance and boosting computational efficiency and practicality by introducing mechanisms such as physical model analogy and safe tunnel guidance. Overall, these advances have pushed these theoretical methods toward more robust, efficient and collaborative practical navigation schemes, laying a solid foundation for the further application of advanced AI techniques in robot path planning [134,135].

Table 7. Other representative algorithms: key improvements and evaluation metrics.

Algorithm	Article	Year	Improvement	Environment	Objective	Evaluation
LSM	[134]	2013	Provides a theoretical basis for local replanning using Level Sets, reducing computational costs.	Static	Single	Simulation
FMM	[135]	2015	Extends FM^2 to 3D formation planning, using a Frenet–Serret frame and grey-level adjustment for shape adaptation.	Static	Multi-objective	Simulation

It is worth noting that the practical deployment of the algorithms discussed above is often constrained by UGV-specific factors. While these constraints—including terrain traversability, nonholonomic dynamics, and perception uncertainty—are frequently abstracted away in algorithm-centric studies, they are explicitly addressed in Section 3, which revisits each algorithmic paradigm from the perspective of real-world UGV operations.

3. Discussion

This section intends to provide the objective research status, development trend and issues being faced regarding path planning of UGVs based on a quantitative analysis of the literature and qualitative review of algorithm characteristics. The analysis synthesizes annual publication trends, the distribution of evaluation methods across algorithm families, statistics on research objectives and testing environments, as well as the core characteristics

of various methods in terms of theoretical performance, applicable scenarios, and inherent limitations, as revealed in Table 8. Table 9 offers a unified multi-dimensional comparison of major algorithm categories across optimality, real-time performance, robustness, interpretability, and integration costs for engineering reference.

Table 8. Advantages and limitations of current path planning algorithms for UGVs.

Algorithm	Advantages	Limitations
Dijkstra	Guarantees global optimality in static environments. Simple and reliable with solid theoretical foundation.	Inefficient in large-scale or dynamic environments. Blind search leads to high computational cost.
A*	More efficient than Dijkstra with heuristic guidance. Guarantees optimal path if heuristic is admissible.	Performance heavily depends on heuristic design. High memory consumption in large-scale environments.
D*	Efficient incremental replanning for dynamic environments. Does not require complete global re-calculation.	Higher memory usage and complex implementation. Less effective with large-scale or frequent environmental changes.
APF	Simple computation and high real-time performance. Naturally generates smooth paths.	Prone to local minima (trapping near obstacles). Goal may be unreachable in complex force fields.
DWA	Considers robot dynamics and kinematic constraints. Excellent for real-time local obstacle avoidance.	Lacks global guidance and prone to local optima. Performance sensitive to evaluation function weights.
TEB	Real-time, online trajectory optimization for dynamic obstacle avoidance. Efficiently replans in local, cluttered, and dynamic environments.	High computational complexity for online optimization. May not guarantee global optimality.
RRT	Efficient in high-dimensional state spaces. Probabilistically complete; does not need environment modeling.	Paths are often non-optimal and non-smooth. Inefficient in narrow passages and slow convergence to optima.
PRM	Suitable for multi-query planning in static environments. Effective for global path planning in high dimensions.	Poor performance in narrow passages due to uniform sampling. Not suitable for dynamic environments.
GA	Strong global search capability and good for complex problems. Robust and can handle multiple constraints/objectives.	Slow convergence speed. Prone to premature convergence (local optima).
ACO	Fast convergence in static scenes via positive feedback. Distributed and parallelizable.	Slow initial search and sensitive to parameters. Parameters are sensitive and require tuning.
FA	Simple structure and easy to implement. Automatically subdivides population for local/global search.	May converge slowly in complex environments. Performance depends on parameter settings.
PSO	Fast convergence speed and few parameters. Simple principle and easy to implement.	Easily falls into local optima. Low precision in later search stages.
SA	Capable of escaping local optima via probabilistic acceptance. Suitable for combinatorial optimization problems.	Convergence speed is slow. Sensitive to cooling schedule parameters.
CS	Simple, few parameters, and strong global search via Lévy flights.	Search direction can be blind and low efficiency. Fixed population size may reduce efficiency.
ABC	Strong global optimization ability and fast convergence. Balances exploration and exploitation well.	Initial solution quality affects performance. Local search mechanism can be simple, leading to stagnation.
CRO	Good global search ability and robustness. Inspired by chemical reactions and offers unique search operators.	Slow convergence for continuous optimization. Difficult to balance exploration and exploitation.
DRL	End-to-end learning from perception to action. Excellent adaptability to complex, unknown, and dynamic environments.	High training cost. Poor safety guarantees and generalization, and “black-box” nature.
IL	Avoids complex reward engineering by learning from demonstrations. Can quickly acquire expert-like behavior.	Dependent on quality and quantity of expert data. Suffers from distributional shift, leading to poor performance in unseen states.

Table 8. *Cont.*

Algorithm	Advantages	Limitations
LQR	Provides optimal control with rigorous theoretical guarantees. Simple implementation and good robustness for linear/near-linear systems.	Requires an accurate linear model of the system. Performance relies on empirical tuning of weight matrices.
MPC	Handles state and input constraints explicitly. Rolling optimization provides feedback correction and good for trajectory tracking.	High computational cost for online optimization. Performance heavily depends on model accuracy.
LSM	Naturally handles topological changes and complex boundaries. Suitable for path planning in continuous space.	High computational cost for solving PDEs. Path optimality is not guaranteed.
FMM	Computationally efficient for computing distance fields. Provides a continuous navigation function.	Primarily for static environments. Path is based on grid discretization, affecting smoothness and optimality.

Table 9. A unified multi-dimensional comparison matrix of UGV path planning algorithms.

Algorithm Category	Optimality	Real-Time Performance	Robustness	Interpretability	Integration Cost
Graph Search	Globally optimal	Moderate	Moderate	High	Low–Moderate
Local Optimization	Locally optimal	High	High	Moderate	Low
Sampling-based	Asymptotically optimal	Moderate	Moderate–High	Moderate	Moderate
Meta-heuristic	Near optimal	Low–Moderate	Moderate	Moderate	Moderate
AI-based	Near optimal	High	High	Low	High
Optimal Control	Optimal for trajectory tracking	High (closed-loop control)	High	Moderate–High	Moderate

Although the algorithms reviewed above are broadly applicable to mobile robots, unmanned ground vehicles (UGVs) operating in real-world environments face several unique constraints that are often oversimplified or ignored in general path planning studies. Unlike indoor robots that operate on flat, predictable surfaces, UGVs must navigate terrains with varying slope, roughness, deformability, and vegetation cover, making traversability analysis a prerequisite for outdoor path planning. Most UGVs are subject to nonholonomic and kinodynamic constraints—such as turning radius limits and coupled acceleration dynamics—which path planners must account for to generate dynamically feasible trajectories. In outdoor environments, GPS signals may be unreliable, and perception systems are subject to noise, occlusions, and environmental interference (e.g., rain, snow, and dust), requiring planners to be robust to localization errors and imperfect sensor data. Energy consumption is another critical metric for UGVs deployed in agriculture, mining, or long-duration missions, as it depends on terrain, slope, velocity, and payload—highlighting the need for multi-objective optimization beyond path length or time. Furthermore, outdoor settings involve dynamic elements, such as pedestrians, animals, and changing weather conditions, which introduce uncertainty and demand real-time replanning capabilities. Incorporating these UGV-specific constraints remains an open challenge; future research should move beyond simplified assumptions and explicitly account for traversability, dynamics, uncertainty, energy, and environmental dynamics—topics currently underrepresented in the literature.

The annual publication quantity is characterized by distinct phases based on the improved algorithm literature referenced in the text and the expanded references in the summary tables of each section (Figure 8). Between 2010 and 2019, papers were published regularly with the growing popularity of various open-source frameworks, such as the

Robot Operating System (ROS) and standard simulation environments, similar to those considered in this study. This period is characterized by a systematic research process placed mainly on the verification and comparison of algorithms. Between the year 2020 and 2023, the growth rate of publications has increased rapidly following the extensive application of the method of deep reinforcement learning in robotics. The phenomenon reveals the catalytic effect that novel methodologies have on the field's output. It should also be taken into account that the overall growth in global publications, as well as the growth in publishers and conferences, has contributed to this increasing trend, which we have considered. After 2024, the number of publications declined from its peak and stabilized, potentially suggesting that researchers may be shifting focus from proposing new models to tackling deeper engineering deployment issues like robustness and interpretability.

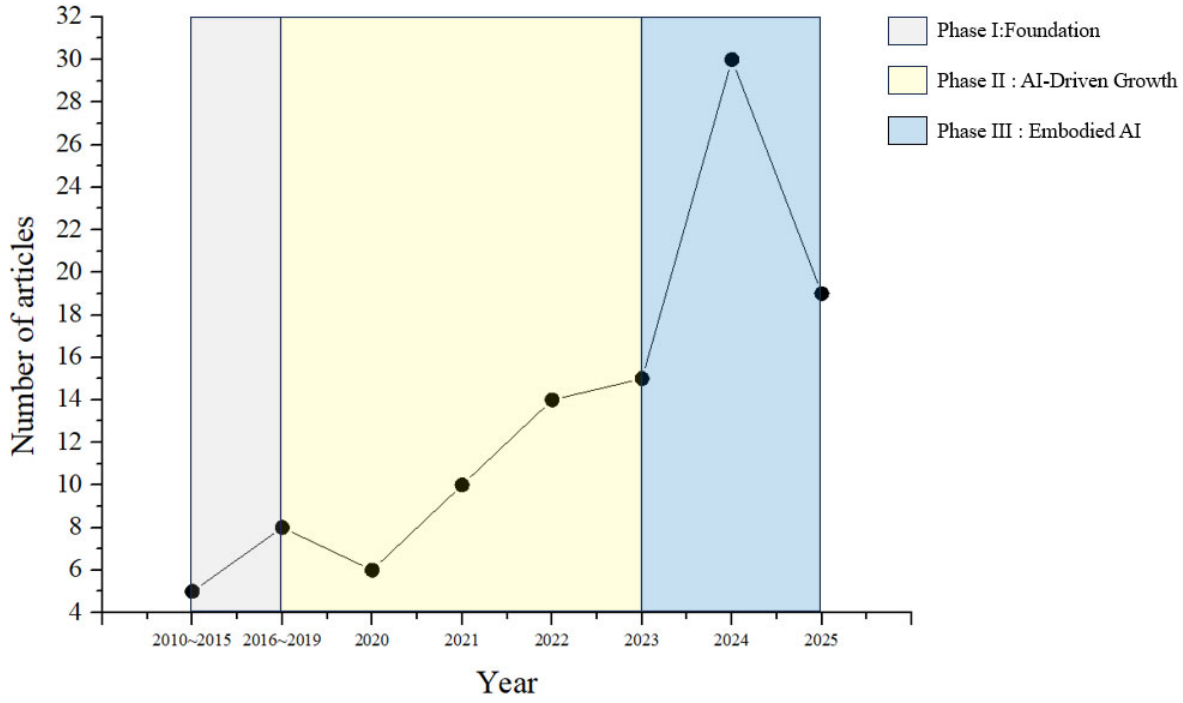


Figure 8. Annual distribution of the literature cited in this review (Created by the authors).

From a methodological perspective, current research reveals several statistically significant features regarding evaluation practices. In terms of algorithm assessment, the vast majority of studies rely heavily on software simulations. Specific data (Figure 9) shows that while evaluation methods differ across algorithm families, simulations are overwhelmingly dominant. For instance, among the AI-based path planning studies included in our statistics, a striking 90% employed simulation-only validation; among meta-heuristic algorithms, approximately 66% of studies fall into this category. In contrast, the local optimization-based category is the only one where the number of studies combining "simulations and physical experiments" exceeds those using "simulations only." This is closely related to its typical role as a real-time module directly deployable within robotic systems.

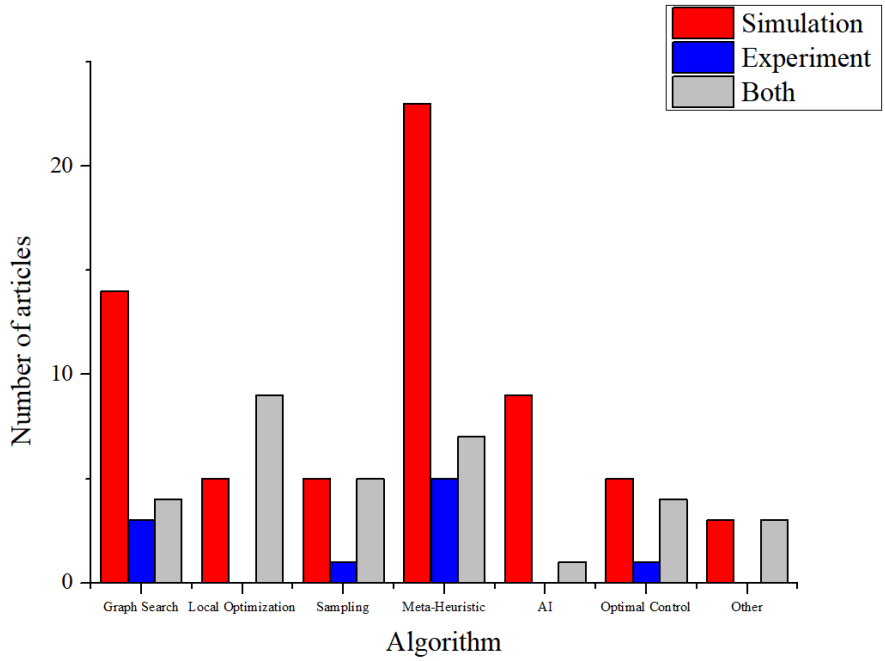
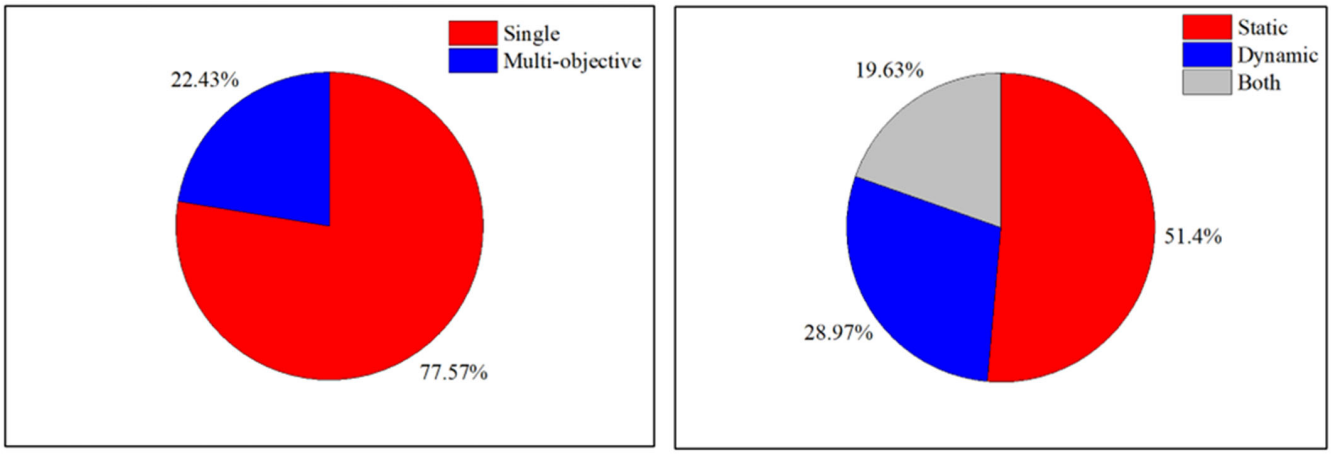


Figure 9. Histogram of evaluation methods for different algorithm categories (Created by the authors).

Regarding the setting of research objectives, statistical data (Figure 10a) indicates that studies focusing on "single-objective optimization," primarily centered on path shortness, dominate in quantity. In comparison, research on "multi-objective optimization," which simultaneously considers factors like smoothness, safety, and energy consumption, accounts for a relatively low proportion. This suggests that although comprehensive performance optimization is recognized as an important direction, a significant volume of work remains concentrated on solving more fundamental or singular optimization problems. Concerning deployment environments (Figure 10b), the number of studies targeting static environments significantly surpasses those addressing dynamic or combined static–dynamic environments. This represents a disconnect from the inherently dynamic nature of real-world application scenarios and is directly linked to the core limitation repeatedly noted in the tables for numerous algorithms: "poor adaptability to dynamic environments" and "weak handling capability for unknown environments.



(**a**) (**b**)

Figure 10. Distribution of key research characteristics in the path planning literature. (**a**) Distribution of research optimization objectives; (**b**) distribution of algorithm deployment environments (Created by the authors).

A comparison of the above trends and data enables a further clarification of a number of tensions within the field. The first refers to the gap between the verification of a simulation and physical deployment. Good performance in simulation environments is usually followed by performance in the real world. In other words, data-driven methods such as AI often show this behavior. New data clearly indicates that those methods are the most deficient in experimental verification. The extent to which these models can handle real sensor noise, model errors, and platform dynamics has yet to be explored. This is consistent with the evaluation provided in Table 8 regarding their weak generalization capability and lack of safety guarantees. Second, there is a tension between simplifications made in research settings and the demands of a complex real world. A great deal of research assumes that the environment in which the study is being conducted is static, and the objective is singular, resulting in a gap in the multi-constraint and highly dynamic complexities encountered in practice. The third involves a trade-off between the performance of theoretical algorithms and their computational efficiency. Although some graph-search and optimal control methods can guarantee the optimality of solutions in theory, they suffer from a high computational cost that cannot satisfy the real-time constraints of high-dimensional or highly dynamic scenes. On the other hand, as described in Table 8, we have approximate methods, e.g., random sampling, meta-heuristic, and learning-based methods, that offer improved processing capability but at the expense of their global optimality, stability and interpretability of solutions.

In summary, research on path planning for unmanned ground vehicles has made significant progress in methodological innovation but still faces core challenges in advancing the technology toward practical applications. Future work may need to focus more on establishing benchmark test environments that closely mimic real-world conditions; systematically balancing optimality, real-time performance, and robustness in algorithm design; and exploring hybrid methods that combine the strengths of classical models and data-driven approaches to bridge the current gap between research and the demands of complex real-world scenarios. Particular attention should be paid to significantly strengthening algorithm research for dynamic unknown environments and promoting more thorough experimental validation of research outcomes on standardized physical platforms. This may be a critical step for the field toward mature applications.

4. Conclusions

This study focuses on the latest advancements in the field of path planning for unmanned ground vehicles. By systematically reviewing and analyzing a large volume of high-impact literature from the past five years (particularly from 2022 to 2025), we conducted quantitative statistics and qualitative comparisons across multiple dimensions, including algorithm evolution, evaluation methods, research objectives, and testing environments. This work not only aims to clarify the rapidly developing technological landscape but also strives to reveal the underlying logic and common biases in research methodologies, thereby providing the field with a forward-looking cognitive framework based on the latest empirical evidence. Based on this analysis, the following three core conclusions are drawn:

(1) The basic strengths and weaknesses of different algorithm classes are in contradiction with each other. Hence, it is very difficult for one single method to satisfy all the requirements of complex applications. A case in point is that while graph-search algorithms ensure optimality, they are not able to achieve sufficient real-time performance; local optimization methods ensure fast responses but do not have a global view; and data-driven methods perform well but have extremely weak interpretability

and safety guarantees. This conflict is the main constraint on the overall performance improvement of algorithms.

(2) A significant deviation exists between the mainstream validation methods in current research and the demands of practical application scenarios. In other words, the state of the art differs from practical availability. Most studies are evaluated using pure simulations of the study software. Moreover, the volume of validation for static environments overwhelmingly surpasses that of dynamic environments. As a result, the actual effectiveness and robustness of many algorithms in the face of the high uncertainty and dynamic interference of the real world have not been verified enough.

(3) Research objectives are shifting from the isolated optimization of single metrics towards achieving comprehensive performance within system integration. Although "single-objective optimization" still dominates in terms of publication volume, the focus of both academia and industry has clearly shifted towards practical system solutions that involve multi-objective trade-offs and multi-algorithm collaboration. Path planning is increasingly being considered within the complete perception–decision–control pipeline.

Based on the above conclusions, this paper posits that future research can seek breakthroughs in the following three directions:

(1) Standardized benchmarks and evaluation protocols:

Currently, the lack of unified benchmarks makes it difficult to compare algorithms across different studies. Future research should develop open-source simulation environments and real-world testbeds with standardized metrics, including success rates under varying obstacle densities, energy efficiency (e.g., energy consumption per unit distance), worst-case computation times, and path smoothness. Such benchmarks would enable fair and reproducible comparisons across algorithms, addressing the over-reliance on simulation-only validation identified in the Conclusions (2).

(2) In-depth exploration of modular and hierarchical hybrid planning architectures:

To break through the performance bottlenecks of single algorithms, systematic research into complementary fusion mechanisms between different algorithms is required. For example, integrating models with a global perspective or strong interpretability as a safety foundation and combining them with highly adaptive data-driven models in a hierarchical or embedded manner can significantly enhance the system's intelligence in open environments while ensuring reliability.

(3) Uncertainty-aware and perception-integrated planning:

Most existing path planning algorithms assume perfect localization, complete environmental knowledge, and deterministic obstacle motion—assumptions that rarely hold in real-world UGV deployments. Future research should integrate planning with perception, SLAM, and dynamic obstacle prediction modules to account for localization errors, sensor noise, occlusions, and unpredictable behaviors of pedestrians or other vehicles. Specific directions include: developing planning algorithms that can effectively utilize scene semantics and predictive information (e.g., pedestrian intent and vehicle trajectory prediction); incorporating uncertainty quantification (e.g., chance-constrained or robust optimization) into planning frameworks; and realizing advanced autonomous navigation systems capable of long-horizon task reasoning and natural human interaction.

Achieving these directions will be key to propelling UGVs from structured scenarios towards widespread adoption in open, dynamic, and collaborative environments.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/a19060439/s1.

Author Contributions: Conceptualization, Y.Z., H.Y., C.L. and Q.M.; methodology, Q.M., M.C., H.Z., and F.W.; formal analysis, Y.Z., H.Y., C.L. and Q.M.; investigation, Q.M., Y.Z., H.Y., H.Z. and C.L.; resources, Q.M., Y.Z., H.Y., and C.L.; data curation, Q.M., Y.Z., H.Y., and C.L.; writing—original draft preparation, Q.M.; writing—review and editing, Q.M. and Y.Z.; visualization, Q.M.; supervision, Y.Z., M.C., H.Y., C.L. and F.W.; project administration, Y.Z., H.Y. and C.L.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Outstanding Young Scientists Program of Beijing Higher Education Institutions (BJJWZYJH01201910006021); a sub-project of the Key Basic Research Projects under the Basic Strengthening Plan (2019-JCJQ-ZD-120-13); the Science and Technology Research Project of Henan Province (202102210081, 252102221010, 252102220084); the Key Scientific Research Project of Colleges and Universities in Henan Province (25A460010); the Special Fund for Basic Scientific Research Operating Expenses of Colleges and Universities in Henan Province (NSFRF2502055); the Doctoral Fund of Henan Polytechnic University (B2024-42); the Principles of Sensors and Detection Technology, the second batch of "14th Five-Year Plan" textbooks for general higher education in Henan Province (Grant No. 2024XBJC09); and the Henan Provincial Natural Science Foundation (Grant No. 262300421351).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: Author Hui Zhang was employed by the company Pingyuan Filter Co., Ltd., Xinxiang 453700, China, Xinxiang, China. The remaining authors declare that this research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Badamasi Aremu, M.; Kabir, I.K.; Ahmed, G.; El-Ferik, S. Autonomous Mobile Robot Path Planning Techniques—A Review: Classical and Heuristic Techniques. *IEEE Access* **2025**, *13*, 117999–118022. [CrossRef]
2. Dong, L.; He, Z.; Song, C.; Sun, C. A Review of Mobile Robot Motion Planning Methods: From Classical Motion Planning Workflows to Reinforcement Learning-Based Architectures. *J. Syst. Eng. Electron.* **2023**, *34*, 439–459. [CrossRef]
3. Dijksta, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
4. Ahmad, J.; Ab Wahab, M.N. Enhancing the Safety and Smoothness of Path Planning through an Integration of Dijkstra's Algorithm and Piecewise Cubic Bezier Optimization. *Expert Syst. Appl.* **2025**, *289*, 128315. [CrossRef]
5. Alyasin, A.; Abbas, E.I.; Hasan, S.D. An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method. In *2019 4th Scientific International Conference Najaf (SICN)*; IEEE: Piscataway, NJ, USA, 2019; pp. 11–14.
6. Wang, J.; Xu, Z.; Zheng, X.; Liu, Z. A Fuzzy Logic Path Planning Algorithm Based on Geometric Landmarks and Kinetic Constraints. *Inf. Technol. Control* **2022**, *51*, 499–514. [CrossRef]
7. Luo, M.; Hou, X.; Yang, J. Surface Optimal Path Planning Using an Extended Dijkstra Algorithm. *IEEE Access* **2020**, *8*, 147827–147838. [CrossRef]
8. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
9. Li, Y.; Jin, R.; Xu, X.; Qian, Y.; Wang, H.; Xu, S.; Wang, Z. A Mobile Robot Path Planning Algorithm Based on Improved A* Algorithm and Dynamic Window Approach. *IEEE Access* **2022**, *10*, 57736–57747. [CrossRef]
10. Lai, R.; Wu, Z.; Liu, X.; Zeng, N. Fusion Algorithm of the Improved A* Algorithm and Segmented Bézier Curves for the Path Planning of Mobile Robots. *Sustainability* **2023**, *15*, 2483. [CrossRef]
11. Liu, Z.; Liu, H.; Lu, Z.; Zeng, Q. A Dynamic Fusion Pathfinding Algorithm Using Delaunay Triangulation and Improved A-Star for Mobile Robots. *IEEE Access* **2021**, *9*, 20602–20621. [CrossRef]
12. Liu, H.; Zhang, Y. ASL-DWA: An Improved A-Star Algorithm for Indoor Cleaning Robots. *IEEE Access* **2022**, *10*, 99498–99515. [CrossRef]
13. Wang, T.; Li, A.; Guo, D.; Du, G.; He, W. Global Dynamic Path Planning of AGV Based on Fusion of Improved A* Algorithm and Dynamic Window Method. *Sensors* **2024**, *24*, 2011. [CrossRef]

14. Stentz, A. Optimal and Efficient Path Planning for Partially-Known Environments. In *1994 IEEE International Conference on Robotics and Automation*; IEEE: San Diego, CA, USA, 1994; pp. 3310–3317.
15. Koenig, S.; Likhachev, M. D*lite. In *Eighteenth National Conference on Artificial Intelligence*; American Association for Artificial Intelligence: Menlo Park, CA, USA, 2002; pp. 476–483.
16. Yao, Y.; Liang, X.; Li, M.; Yu, K.; Chen, Z.; Ni, C.; Teng, Y. Path Planning Method Based on D* Lite Algorithm for Unmanned Surface Vehicles in Complex Environments. *China Ocean Eng.* **2021**, *35*, 372–383. [CrossRef]
17. Zheng, W.; Yu, H. Research on the Safety of AGV Path Planning Based on D* Algorithm. In *2024 6th International Conference on Artificial Intelligence and Computer Applications (ICAICA)*; IEEE: Dalian, China, 2024; pp. 216–220.
18. Maurovic, I.; Seder, M.; Lenac, K.; Petrovic, I. Path Planning for Active SLAM Based on the D* Algorithm with Negative Edge Weights. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1321–1331. [CrossRef]
19. Przybylski, M.; Putz, B. D* Extra Lite: A Dynamic A* with Search–Tree Cutting and Frontier–Gap Repairing. *Int. J. Appl. Math. Comput. Sci.* **2017**, *27*, 273–290. [CrossRef]
20. Hernandez Ulloa, C.; Baier, J.A.; Asin-Acha, R. Multipath Adaptive A*: Factors That Influence Performance in Goal-Directed Navigation in Unknown Terrain. *IEEE Access* **2020**, *8*, 116724–116732. [CrossRef]
21. Lin, Z.; Wu, K.; Shen, R.; Yu, X.; Huang, S. An Efficient and Accurate A-Star Algorithm for Autonomous Vehicle Path Planning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 9003–9008. [CrossRef]
22. Wang, H.; Lou, S.; Jing, J.; Wang, Y.; Liu, W.; Liu, T. The EBS-A* Algorithm: An Improved A* Algorithm for Path Planning. *PLoS ONE* **2022**, *17*, e0263841. [CrossRef]
23. Suanpang, P.; Jamjuntr, P. Optimizing Autonomous UAV Navigation with D* Algorithm for Sustainable Development. *Sustainability* **2024**, *16*, 7867. [CrossRef]
24. Alshammrei, S.; Boubaker, S.; Kolsi, L. Improved Dijkstra Algorithm for Mobile Robot Path Planning and Obstacle Avoidance. *Comput. Mater. Contin.* **2022**, *72*, 5939–5954. [CrossRef]
25. Sundarraj, S.; Reddy, R.V.K.; Basam, M.B.; Lokesh, G.H.; Flammini, F.; Natarajan, R. Route Planning for an Autonomous Robotic Vehicle Employing a Weight-Controlled Particle Swarm-Optimized Dijkstra Algorithm. *IEEE Access* **2023**, *11*, 92433–92442. [CrossRef]
26. Khatib, O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In *Autonomous Robot Vehicles*; Cox, I.J., Wilfong, G.T., Eds.; Springer: New York, NY, USA, 1986; pp. 396–404. ISBN 978-1-4613-8999-6.
27. Yu, L.; Chen, Z.; Wu, H.; Xu, Z.; Chen, B. Soft Actor-Critic Combining Potential Field for Global Path Planning of Autonomous Mobile Robot. *IEEE Trans. Veh. Technol.* **2025**, *74*, 7114–7123. [CrossRef]
28. Yang, C.; Pan, J.; Wei, K.; Lu, M.; Jia, S. A Novel Unmanned Surface Vehicle Path-Planning Algorithm Based on A* and Artificial Potential Field in Ocean Currents. *J. Mar. Sci. Eng.* **2024**, *12*, 285. [CrossRef]
29. Zhai, L.; Liu, C.; Zhang, X.; Wang, C. Local Trajectory Planning for Obstacle Avoidance of Unmanned Tracked Vehicles Based on Artificial Potential Field Method. *IEEE Access* **2024**, *12*, 19665–19681. [CrossRef]
30. Luo, J.; Wang, Z.-X.; Pan, K.-L. Reliable Path Planning Algorithm Based on Improved Artificial Potential Field Method. *IEEE Access* **2022**, *10*, 108276–108284. [CrossRef]
31. Fox, D.; Burgard, W.; Thrun, S. The Dynamic Window Approach to Collision Avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
32. Hu, L.; Cai, C. ARIME-DWA: Robot Path Planning Utilizing Advanced RIME Optimization and Dynamic Windows Approach. *IEEE Access* **2024**, *12*, 191480–191497. [CrossRef]
33. Liu, Y.; Guo, S.; Tang, S.; Song, J.; Zhang, J. Path Planning for Robots Based on Adaptive Dual-Layer Ant Colony Optimization Algorithm and Adaptive Dynamic Window Approach. *IEEE Sens. J.* **2025**, *25*, 19694–19708. [CrossRef]
34. Li, Y.; Zhao, J.; Chen, Z.; Xiong, G.; Liu, S. A Robot Path Planning Method Based on Improved Genetic Algorithm and Improved Dynamic Window Approach. *Sustainability* **2023**, *15*, 4656. [CrossRef]
35. Hou, J.; Jiang, W.; Luo, Z.; Yang, L.; Hu, X.; Guo, B. Dynamic Path Planning for Mobile Robots by Integrating Improved Sparrow Search Algorithm and Dynamic Window Approach. *Actuators* **2024**, *13*, 24. [CrossRef]
36. Roesmann, C.; Feiten, W.; Woesch, T.; Hoffmann, F.; Bertram, T. Trajectory Modification Considering Dynamic Constraints of Autonomous Robots. In *ROBOTIK 2012; 7th German Conference on Robotics*; IEEE: Piscataway, NJ, USA, 2012; pp. 1–6.
37. Wu, J.; Ma, X.; Peng, T.; Wang, H. An Improved Timed Elastic Band (TEB) Algorithm of Autonomous Ground Vehicle (AGV) in Complex Environment. *Sensors* **2021**, *21*, 8312. [CrossRef] [PubMed]
38. Li, X.; Li, T.; Zhang, Y.; Zhang, Y.; Li, Z.; Ban, L.; Sun, K. A*-TEB: An Improved A* Algorithm Based on the TEB Strategy for Multi-Robot Motion Planning. *Sensors* **2025**, *25*, 6117. [CrossRef] [PubMed]
39. Wang, J.; Luo, Y.; Tan, X. Path Planning for Automatic Guided Vehicles (AGVs) Fusing MH-RRT with Improved TEB. *Actuators* **2021**, *10*, 314. [CrossRef]
40. Liu, Z.; Chang, Z.; Qiao, C.; Chen, H.; Zong, G. Improved Bidirectional A* and Time-Elastic Band Algorithm Based Mobile Robot Path Planning. In *2025 37th Chinese Control and Decision Conference (CCDC)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6.

41. Kozhubaev, Y.; Yang, R. Simulation of Dynamic Path Planning of Symmetrical Trajectory of Mobile Robots Based on Improved A* and Artificial Potential Field Fusion for Natural Resource Exploration. *Symmetry* **2024**, *16*, 801. [CrossRef]
42. Xing, S.; Fan, P.; Ma, X.; Wang, Y. Research on Robot Path Planning by Integrating State-Based Decision-Making A* Algorithm and Inertial Dynamic Window Approach. *Intell. Serv. Robot.* **2024**, *17*, 901–914. [CrossRef]
43. Chen, L.; Liu, R.; Jia, D.; Xian, S.; Ma, G. Improvement of the TEB Algorithm for Local Path Planning of Car-like Mobile Robots Based on Fuzzy Logic Control. *Actuators* **2025**, *14*, 12. [CrossRef]
44. LaValle, S. Rapidly-Exploring Random Trees: A New Tool for Path Planning. *Res. Rep.* **1998**, *9811*, 10014962955.
45. Karaman, S.; Walter, M.R.; Perez, A.; Frazzoli, E.; Teller, S. Anytime Motion Planning Using the RRT*. In *2011 IEEE International Conference on Robotics and Automation*; IEEE: Piscataway, NJ, USA, 2011; pp. 1478–1483.
46. Qi, J.; Yang, H.; Sun, H. MOD-RRT*: A Sampling-Based Algorithm for Robot Path Planning in Dynamic Environment. *IEEE Trans. Ind. Electron.* **2021**, *68*, 7244–7251. [CrossRef]
47. Sheng, Z.; Song, T.; Song, J.; Liu, Y.; Ren, P. Bidirectional Rapidly Exploring Random Tree Path Planning Algorithm Based on Adaptive Strategies and Artificial Potential Fields. *Eng. Appl. Artif. Intell.* **2025**, *148*, 110393. [CrossRef]
48. Zhou, Y.; Lu, Y.; Lv, L. GVP-RRT: A Grid Based Variable Probability Rapidly-Exploring Random Tree Algorithm for AGV Path Planning. *Complex Intell. Syst.* **2024**, *10*, 8273–8286. [CrossRef]
49. Wu, J.; Zhao, L.; Liu, R. Research on Path Planning of a Mining Inspection Robot in an Unstructured Environment Based on an Improved Rapidly Exploring Random Tree Algorithm. *Appl. Sci.* **2024**, *14*, 6389. [CrossRef]
50. Kavraki, L.E.; Svestka, P.; Latombe, J.-C.; Overmars, M.H. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
51. Cao, K.; Cheng, Q.; Gao, S.; Chen, Y.; Chen, C. Improved PRM for Path Planning in Narrow Passages. In *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*; IEEE: Tianjin, China, 2019; pp. 45–50.
52. Ravankar, A.A.; Ravankar, A.; Emaru, T.; Kobayashi, Y. HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots. *IEEE Access* **2020**, *8*, 221743–221766. [CrossRef]
53. Chai, Q.; Wang, Y.; He, Y.; Xu, C.; Hong, Z. Improved PRM Path Planning in Narrow Passages Based on PSO. In *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*; IEEE: Guilin, China, 2022; pp. 41–46.
54. Ali, M.; Das, S.; Townley, S. Robot Differential Drive Navigation Through Probabilistic Roadmap and Pure Pursuit. *IEEE Access* **2025**, *13*, 22118–22132. [CrossRef]
55. Zhong, H.; Cong, M.; Wang, M.; Du, Y.; Liu, D. HB-RRT:A Path Planning Algorithm for Mobile Robots Using Halton Sequence-Based Rapidly-Exploring Random Tree. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108362. [CrossRef]
56. Hu, Y.; Chen, X.; Tang, P.; Zhang, H.; Jin, J.; Mao, S. A Path Planning Framework for Robots Based on Improved Parallel Sampling RRT and Offset Guidance DWA. *IEEE Sens. J.* **2025**, *25*, 35597–35608. [CrossRef]
57. Qiao, L.; Luo, X.; Luo, Q. An Optimized Probabilistic Roadmap Algorithm for Path Planning of Mobile Robots in Complex Environments with Narrow Channels. *Sensors* **2022**, *22*, 8983. [CrossRef]
58. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, 1st ed.; Complex Adaptive Systems; MIT Press: Cambridge, MA, USA, 1992; ISBN 9780262275552.
59. Cheng, K.P.; Mohan, R.E.; Khanh Nhan, N.H.; Le, A.V. Multi-Objective Genetic Algorithm-Based Autonomous Path Planning for Hinged-Tetro Reconfigurable Tiling Robot. *IEEE Access* **2020**, *8*, 121267–121284. [CrossRef]
60. Feng, T.; Li, J.; Jiang, H.; Yang, S.X.; Wang, P.; Teng, Y.; Chen, S.; Fu, Q.; Luo, B. The Optimal Global Path Planning of Mobile Robot Based on Improved Hybrid Adaptive Genetic Algorithm in Different Tasks and Complex Road Environments. *IEEE Access* **2024**, *12*, 18400–18415. [CrossRef]
61. Liu, C.; Yan, X.; Liu, C.; Li, G. Dynamic Path Planning for Mobile Robot Based on Improved Genetic Algorithm. *Chin. J. Electron.* **2010**, *19*, 245–248. [CrossRef]
62. Dorigo, M.; Maniezzo, V.; Colorni, A. *Positive Feedback as a Search Strategy*; 91-016; Dipartimento di Elettronica, Politecnico di Milano: Milan, Italy, 1999.
63. Wang, W.; Li, J.; Bai, Z.; Wei, Z.; Peng, J. Toward Optimization of AGV Path Planning: An RRT* -ACO Algorithm. *IEEE Access* **2024**, *12*, 18387–18399. [CrossRef]
64. Lin, N. Path Planning of Library Management Robot Based on PDO-ACO Algorithm. *IEEE Access* **2025**, *13*, 78376–78390. [CrossRef]
65. Wu, P.; Zhong, L.; Xiong, J.; Zeng, Y.; Pei, M. Two-Level Vehicle Path Planning Model for Multi-Warehouse Robots with Conflict Solution Strategies and Improved ACO. *J. Intell. Connect. Veh.* **2023**, *6*, 102–112. [CrossRef]
66. Yang, X.-S. Firefly Algorithm, Stochastic Test Functions and Design Optimisation. *Int. J. Bio-Inspir. Comput.* **2010**, *2*, 78–84. [CrossRef]
67. Chaudhary, K.; Prasad, A.; Prasad, A.; Sharma, B. Robot Motion Control Using Stepping Ahead Firefly Algorithm and Kinematic Equations. *IEEE Access* **2024**, *12*, 43078–43088. [CrossRef]

68. Li, F.; Fan, X.; Hou, Z. A Firefly Algorithm with Self-Adaptive Population Size for Global Path Planning of Mobile Robot. *IEEE Access* **2020**, *8*, 168951–168964. [CrossRef]
69. Sun, W.; Huan, J.; Wang, Q.; Yu, H.; Gao, S. FADQN: A Heuristic Reinforcement Learning Mechanism for UAV Path Planning in Unknown Environment. *IEEE Access* **2025**, *13*, 131909–131921. [CrossRef]
70. Eberhart, R.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*; IEEE: Nagoya, Japan, 1995; pp. 39–43.
71. Zheng, L.; Yu, W.; Li, G.; Qin, G.; Luo, Y. Particle Swarm Algorithm Path-Planning Method for Mobile Robots Based on Artificial Potential Fields. *Sensors* **2023**, *23*, 6082. [CrossRef]
72. Wang, L.; Liu, L.; Lu, X. Robot Path Planning Based on Generative Learning Particle Swarm Optimization. *IEEE Access* **2024**, *12*, 130063–130072. [CrossRef]
73. Huang, C.; Zhao, Y.; Zhang, M.; Yang, H. APSO: An A*-PSO Hybrid Algorithm for Mobile Robot Path Planning. *IEEE Access* **2023**, *11*, 43238–43256. [CrossRef]
74. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092. [CrossRef]
75. Shi, K.; Wu, W.; Wu, Z.; Jiang, B.; Karimi, H.R. Coverage Path Planning for Cleaning Robot Based on Improved Simulated Annealing Algorithm and Ant Colony Algorithm. *Signal Image Video Process.* **2024**, *18*, 3275–3284. [CrossRef]
76. Shi, W.; He, Z.; Tang, W.; Liu, W.; Ma, Z. Path Planning of Multi-Robot Systems with Boolean Specifications Based on Simulated Annealing. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6091–6098. [CrossRef]
77. Gao, J.; Xu, X.; Pu, Q.; Petrovic, P.B.; Rodić, A.; Wang, Z. A Hybrid Path Planning Method Based on Improved A* and CSA-APF Algorithms. *IEEE Access* **2024**, *12*, 39139–39151. [CrossRef]
78. Yang, X.S.; Deb, S. Cuckoo Search via Lévy Flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*; IEEE: Piscataway, NJ, USA, 2009; pp. 210–214.
79. Min, W.; Mo, L.; Yin, B.; Li, S. An Improved Cuckoo Search Algorithm and Its Application in Robot Path Planning. *Appl. Sci.* **2024**, *14*, 9572. [CrossRef]
80. Alireza, M.; Vincent, D.; Tony, W. Experimental Study of Path Planning Problem Using EMCOA for a Holonomic Mobile Robot. *J. Syst. Eng. Electron.* **2021**, *32*, 1450–1462. [CrossRef]
81. Chen, H.; Wang, S. Path Planning Method of Industrial Intelligent Welding Robot Based on Cuckoo Search Algorithm. *Int. J. Manuf. Technol. Manag.* **2024**, *38*, 1–13. [CrossRef]
82. Sahu, B.; Das, P.K.; Kumar, R. A Modified Cuckoo Search Algorithm Implemented with SCA and PSO for Multi-Robot Cooperation and Path Planning. *Cogn. Syst. Res.* **2023**, *79*, 24–42. [CrossRef]
83. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; TR06; Erciyes University: Kayseri, Turkey, 2005.
84. Yu, Z.; Duan, P.; Meng, L.; Han, Y.; Ye, F. Multi-Objective Path Planning for Mobile Robot with an Improved Artificial Bee Colony Algorithm. *Math. Biosci. Eng.* **2023**, *20*, 2501–2529. [CrossRef]
85. Kumar, S.; Sikander, A. Optimum Mobile Robot Path Planning Using Improved Artificial Bee Colony Algorithm and Evolutionary Programming. *Arab. J. Sci. Eng.* **2022**, *47*, 3519–3539. [CrossRef]
86. Li, G.; Liu, C.; Wu, L.; Xiao, W. A Mixing Algorithm of ACO and ABC for Solving Path Planning of Mobile Robot. *Appl. Soft Comput.* **2023**, *148*, 110868. [CrossRef]
87. Alatas, B. A Novel Chemistry Based Metaheuristic Optimization Method for Mining of Classification Rules. *Expert Syst. Appl.* **2012**, *39*, 11080–11088. [CrossRef]
88. Yang, Q.; Yang, Z.; Hu, G.-X.; Du, W. A Random Chemical Reaction Algorithm Based on Double Containers for Robot Path Planning. In *2017 International Conference on Progress in Informatics and Computing (PIC)*; IEEE: Piscataway, NJ, USA, 2017; pp. 404–409.
89. Zhang, M.; Chen, L.; Chen, X. An Advanced Chemical Reaction Optimization Algorithm Based on Balanced Local and Global Search. *Math. Probl. Eng.* **2018**, *2018*, 8042689. [CrossRef]
90. Li, D.; Wang, L.; Cai, J.; Ma, K.; Tan, T. Research on Terminal Distance Index-Based Multi-Step Ant Colony Optimization for Mobile Robot Path Planning. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 2321–2337. [CrossRef]
91. Islam, M.R.; Protik, P.; Das, S.; Boni, P.K. Mobile Robot Path Planning with Obstacle Avoidance Using Chemical Reaction Optimization. *Soft Comput.* **2021**, *25*, 6283–6310. [CrossRef]
92. Li, J.; Hu, Y.; Yang, S.X. A Novel Knowledge-Based Genetic Algorithm for Robot Path Planning in Complex Environments. *IEEE Trans. Evol. Comput.* **2025**, *29*, 375–389. [CrossRef]
93. Hou, Z.; Cheng, W.; Li, F. Path Planning of Mobile Robot Based on Chaotic Courtship Firefly Algorithm. *Mach. Toolhydraul.* **2024**, *52*, 55–59. (In Chinese)
94. Gan, L. Research on obstacle avoidance path planning of indoor robot in library based on improved firefly algorithm. *Meas. Control Technol.* **2025**, *44*, 33–37, 43. (In Chinese) [CrossRef]

95. Yifei, T.; Meng, Z.; Jingwei, L.; Dongbo, L.; Yulin, W. Research on Intelligent Welding Robot Path Optimization Based on GA and PSO Algorithms. *IEEE Access* **2018**, *6*, 65397–65404. [CrossRef]
96. Shi, K.; Wu, Z.; Jiang, B.; Karimi, H.R. Dynamic Path Planning of Mobile Robot Based on Improved Simulated Annealing Algorithm. *J. Frankl. Inst.* **2023**, *360*, 4378–4398. [CrossRef]
97. Xie, Y.; Zeng, X.; Feng, W. Application of Improved Cuckoo Search Algorithm in Multi-robot Task Allocation and Path Planning. *Comput. Appl. Softw.* **2021**, *38*, 285–290. (In Chinese)
98. Cui, Y.; Hu, W.; Rahmani, A. Multi-Robot Path Planning Using Learning-Based Artificial Bee Colony Algorithm. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107579. [CrossRef]
99. Kober, J.; Bagnell, J.A.; Peters, J. Reinforcement Learning in Robotics: A Survey. *Int. J. Robot. Res.* **2013**, *32*, 1238–1274. [CrossRef]
100. Gao, S.; Xu, Y.; Guo, X.; Liu, C.; Wang, X. DPDQN-TER: An Improved Deep Reinforcement Learning Approach for Mobile Robot Path Planning in Dynamic Scenarios. *Sensors* **2025**, *25*, 6741. [CrossRef]
101. Kumaar, A.A.N.; Kochuvila, S. Mobile Service Robot Path Planning Using Deep Reinforcement Learning. *IEEE Access* **2023**, *11*, 100083–100096. [CrossRef]
102. Bai, Z.; Pang, H.; He, Z.; Zhao, B.; Wang, T. Path Planning of Autonomous Mobile Robot in Comprehensive Unknown Environment Using Deep Reinforcement Learning. *IEEE Internet Things J.* **2024**, *11*, 22153–22166. [CrossRef]
103. Yang, K.; Liu, L. An Improved Deep Reinforcement Learning Algorithm for Path Planning in Unmanned Driving. *IEEE Access* **2024**, *12*, 67935–67944. [CrossRef]
104. Pomerleau, D.A. *Alvinn: An Autonomous Land Vehicle in a Neural Network*; Morgan Kaufmann Publishers Inc.: San Mateo, CA, USA, 1989.
105. Rizk, A.; Rizk, A.; Hana, C.A.; Bakouny, Y.; Khatounian, F. End-to-End Sketch-Guided Path Planning through Imitation Learning for Autonomous Mobile Robots. In *2025 IEEE 6th International Conference on Image Processing, Applications and Systems (IPAS)*; IEEE: Lyon, France, 2025; pp. 1–7.
106. Yang, Z.; Nai, W.; Li, D.; Liu, L.; Chen, Z. A Mixed Generative Adversarial Imitation Learning Based Vehicle Path Planning Algorithm. *IEEE Access* **2024**, *12*, 85859–85879. [CrossRef]
107. Chen, L.; Wang, Y.; Miao, Z.; Mo, Y.; Feng, M.; Zhou, Z.; Wang, H. Transformer-Based Imitative Reinforcement Learning for Multirobot Path Planning. *IEEE Trans. Ind. Inform.* **2023**, *19*, 10233–10243. [CrossRef]
108. Lai, J.; Wu, Z.; Ren, Z.; Chen, C.; Tan, Q.; Xie, S. A Lyapunov-Based Framework for Trajectory Planning of Wheeled Vehicle Using Imitation Learning. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 12118–12133. [CrossRef]
109. Carvalho, J.P.; Aguiar, A.P. Deep Reinforcement Learning for Zero-Shot Coverage Path Planning with Mobile Robots. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 1594–1609. [CrossRef]
110. Cèsar-Tondreau, B.; Warnell, G.; Stump, E.; Kochersberger, K.; Waytowich, N.R. Improving Autonomous Robotic Navigation Using Imitation Learning. *Front. Robot. AI* **2021**, *8*, 627730. [CrossRef]
111. Van Den Berg, J.; Abbeel, P.; Goldberg, K. LQG-MP: Optimized Path Planning for Robots with Motion Uncertainty and Imperfect State Information. *Int. J. Robot. Res.* **2011**, *30*, 895–913. [CrossRef]
112. Kalman, R.E. Contributions to the Theory of Optimal Control. *Bol. Soc. Mat. Mex.* **1960**, *5*, 102–119.
113. Zou, Q.; Du, X.; Liu, Y.; Chen, H.; Wang, Y.; Yu, J. Dynamic Path Planning and Motion Control of Microrobotic Swarms for Mobile Target Tracking. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 2454–2468. [CrossRef]
114. Zhang, H.; Gong, J.; Jiang, Y.; Xiong, G.; Chen, H. An Iterative Linear Quadratic Regulator Based Trajectory Tracking Controller for Wheeled Mobile Robot. *J. Zhejiang Univ. Sci. C* **2012**, *13*, 593–600. [CrossRef]
115. Zhang, Y.; Gao, F.; Zhao, F. Research on Path Planning and Tracking Control of Autonomous Vehicles Based on Improved RRT* and PSO-LQR. *Processes* **2023**, *11*, 1841. [CrossRef]
116. Cutler, C.R.; Ramaker, B.L. Dynamic Matrix Control—A computer control algorithm. In *Joint Automatic Control Conference*; American Automatic Control Council: Evanston, IL, USA, 1980; Volume 17, p. 72.
117. Richalet, J.; Rault, A.; Testud, J.L.; Papon, J. Model Predictive Heuristic Control. *Autom. J. IFAC* **1978**, *14*, 413–428. [CrossRef]
118. Pezzato, C.; Salmi, C.; Trevisan, E.; Spahn, M.; Alonso-Mora, J.; Corbato, C.H. Sampling-Based Model Predictive Control Leveraging Parallelizable Physics Simulations. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2750–2757. [CrossRef]
119. Xin, J.; Wu, X.; D'Ariano, A.; Negenborn, R.; Zhang, F. Model Predictive Path Planning of AGVs: Mixed Logical Dynamical Formulation and Distributed Coordination. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 6943–6954. [CrossRef]
120. Li, J.; Wang, A.; Lyu, Y.; Ding, Z. Global Path Guided Model Predictive Path Integral Control: Applications to GPU-Parallelizable Robot Simulation Systems. *Comput. Electr. Eng.* **2024**, *120*, 109645. [CrossRef]
121. Li, J.; Sun, J.; Liu, L.; Xu, J. Model Predictive Control for the Tracking of Autonomous Mobile Robot Combined with a Local Path Planning. *Meas. Control* **2021**, *54*, 1319–1325. [CrossRef]
122. Amertet, S.; Gebresenbet, G.; Alwan, H.M. Optimizing the Performance of a Wheeled Mobile Robots for Use in Agriculture Using a Linear-Quadratic Regulator. *Robot. Auton. Syst.* **2024**, *174*, 104642. [CrossRef]

123. Li, B.; Ji, Z.; Zhao, Z.; Yang, C. Model Predictive Optimization and Terminal Sliding Mode Motion Control for Mobile Robot with Obstacle Avoidance. *IEEE Trans. Ind. Electron.* **2025**, *72*, 9293–9303. [CrossRef]
124. Xin, J.; Qu, Y.; Zhang, F.; Negenborn, R. Distributed Model Predictive Contouring Control for Real-Time Multi-Robot Motion Planning. *Complex Syst. Model. Simul.* **2022**, *2*, 273–287. [CrossRef]
125. Osher, S.; Sethian, J.A. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *J. Comput. Phys.* **1988**, *79*, 12–49. [CrossRef]
126. Zhang, X.-G.; Zhang, W.; Li, H.; Liu, M.-Q.; Lyu, S. Level Set Based Path Planning Using a Novel Path Optimization Algorithm for Robots. *Int. J. Precis. Eng. Manuf.* **2018**, *19*, 1331–1338. [CrossRef]
127. Li, X.; Zhao, G.; Li, B. Generating Optimal Path by Level Set Approach for a Mobile Robot Moving in Static/Dynamic Environments. *Appl. Math. Model.* **2020**, *85*, 210–230. [CrossRef]
128. Stagg, G.; Peterson, C.K. Multi-Agent Path Planning for Level Set Estimation Using B-Splines and Differential Flatness. *IEEE Robot. Autom. Lett.* **2024**, *9*, 4758–4765. [CrossRef]
129. Sethian, J.A. A Fast Marching Level Set Method for Monotonically Advancing Fronts. *Proc. Natl. Acad. Sci. USA* **1996**, *93*, 1591–1595. [CrossRef]
130. Liu, J.; Zhang, Y.; Bian, S.; Song, R.; Zhang, C.-H.; Wei, Y.; Hua, C. P-RT-BFMT: A Prediction-Based Real-Time Bidirectional Fast Marching Tree for Robot Motion Planning in Dynamic Environments. *IEEE Trans. Ind. Electron.* **2025**, *72*, 13627–13636. [CrossRef]
131. Wu, Z.; Chen, Y.; Liang, J.; He, B.; Wang, Y. ST-FMT*: A Fast Optimal Global Motion Planning for Mobile Robot. *IEEE Trans. Ind. Electron.* **2022**, *69*, 3854–3864. [CrossRef]
132. Silveira, J.; Cabral, K.; Givigi, S.; Marshall, J.A. Real-Time Fast Marching Tree for Mobile Robot Motion Planning in Dynamic Environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: London, UK, 2023; pp. 7837–7843.
133. Giang, T.T.C.; Binh, H.T.T.; Luong, H.V.D.; Hoang, N.H.; Huy, D.Q. Fast Marching Firework Method for Multi-Goal Mobile Robot Path Planning in Complex Obstacle Maps. *Intell. Serv. Robot.* **2025**, *18*, 1467–1484. [CrossRef]
134. Xu, B.; Stilwell, D.J.; Kurdila, A.J. Fast Path Re-Planning Based on Fast Marching and Level Sets. *J. Intell. Robot. Syst.* **2013**, *71*, 303–317. [CrossRef]
135. Alvarez, D.; Gomez, J.V.; Garrido, S.; Moreno, L. 3D Robot Formations Path Planning with Fast Marching Square. *J. Intell. Robot. Syst.* **2015**, *80*, 507–523. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.