Journal of Soft Computing and Computer Applications

Manuscript 1027

A Comprehensive Review of the A* Algorithm: Evolution, Applications, and Future Trends in Path Planning

Saleel H. Abood

Hussein M. H. Al-Khafaji

Mohanned M. H. Al-Khafaji

JOURNAL OF SOFT COMPUTING AND COMPUTER APPLICATIONS 2026;3:1027

REVIEW

A Comprehensive Review of the A* Algorithm: Evolution, Applications, and Future Trends in Path Planning

Saleel H. Abood [a,*], **Hussein M. H. Al-Khafaji** [b], **Mohanned M. H. Al-Khafaji** [c]

[a] University of Technology – Iraq, College of Mechanical Engineering, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq
[b] University of Technology – Iraq, College of Mechanical Engineering, Department of Mechanical Engineering, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq
[c] University of Technology – Iraq, College of Production Engineering and Metallurgy, Department of Computer-Aided Design and Manufacturing Engineering, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq

ABSTRACT

Despite being a fundamental problem to autonomous robotics and intelligent navigation systems, path planning is still a challenge. The A* algorithm is often used among search-based techniques for optimal search performance, as it's a tradeoff of computation. The above techniques have been developed for various applications as many versions of A* Dynamic A* (D*), D* Lite, Hybrid A*, and Anytime A* are suggested to deal with dynamic environments, real-time constraints, and kinematic restrictions. This paper comprehensively and structurally reviews the A* algorithm and its major extensions, encompassing historical development, methodological improvements, practical applications, and upcoming research trends. This review differs from the descriptive surveys conducted in this book, comparing classical and hybrid approaches (to include semi-quantitative analysis) on different computational complexity, re-planning mechanisms, scalability, and applicability to robotic systems. It lays out the current challenges, implementation frameworks, and practical libraries as well, giving a comprehensive view linking theory evolution with real-world applications with robots. The goal of the review is to be a reference framework for researchers concerned with how this trend of A*-based planning is transforming into dynamic, cooperative and intelligent navigation systems.

Keywords: A* algorithm, Heuristic search, D* lite, Hybrid A*, Dynamic path planning, Autonomous navigation

Received 29 December 2025; revised 16 March 2026; accepted 20 April 2026.
Available online 16 June 2026

* Corresponding author.
E-mail addresses: me.20.29@grad.uotechnology.edu.iq (S. H. Abood), hussein.m.hussein@uotechnology.edu.iq (H.. M. H. Al-Khafaji), mohanned.m.hussein@uotechnology.edu.iq (M. M. H. Al-Khafaji).

https://doi.org/10.70403/3008-1084.1027

1. Introduction

The problem of path planning for mobile robots is the search for an efficient way to travel effectively and safely between a starting point and the end point and overcome obstacles. This process is divided into global planning (building a full path in a known environment) and local planning (shifting the route according to dynamic and real-time environmental surroundings). Some algorithms, including A*, Rapidly-exploring Random Tree (RRT), Dijkstra's algorithm, Genetic Algorithms (GA), and neural network algorithm, have been applied that predict the path according to the environment (static or dynamic). The ultimate objective of this process is to reduce travel time, avoid collisions, and reach autonomous robot motion. Although advances have been made in this area, the path planning problem is computationally demanding and has attracted extensive research [1]. Several surveys have reviewed classical and intelligent path planning algorithms for autonomous navigation [2–4]. One of those algorithms, A* algorithm, this algorithm is the most used and studied in this area. Its low complexity, the structural ability of the solution to balance between the cost (G(n)) and the heuristic estimate of the remaining cost (H(n)). This equilibrium allows the characteristic completeness and optimal optimization characteristic, as long as the heuristics to be computed are acceptable, to be taken from A*. From memory requirements, time needs on the environment, dynamic environments – all of these techniques of optimization as time progressed still further more improvements were made [5, 6]. While a number of more advanced algorithms have emerged in recent years including probabilistic search algorithms, machine learning techniques etc., the A* algorithm is still a reference algorithm in this area. Emerging developments also generalize and expand over the A*-based planning mechanisms to apply cooperative multi-agent systems, distributed decision-making architectures and multi-objective optimization models in autonomous surface and underwater vehicles. Such progress can be seen through the transformation from conventional path planning principles into a cooperative and adaptive approach for navigation in dynamic and complicated environments [7–9]. These representational studies are introduced in order that, in terms of the modern approaches, the heuristic search laws continue to motivate the modern day cooperative and multiobjective autonomous navigation. Therefore the goal for this paper is to analyse on A* the algorithm at its theoretical roots, its main components, its practical applications to various domains and to present the A* reference framework, which connects A* evolution with cutting-edge Artificial Intelligence (AI) through the use of practical solutions in smart systems in current systems. With this in mind, the core work in this paper is to synthesize and merge the primary A* algorithm development processes in this paper into comprehensive and current review. Unlike existing surveys, this approach provides a more structured critical analysis that is based on heuristic design, algorithmic variants, and hybrid AI-based extensions all in a comparative context. It explores how various researchers adapted the algorithm to dynamically, statically, or real time environments, with an analytical treatment of both objectives and performance trade-offs of each variant. Moreover, the proposed paper covers the applicability of robotics, autonomous driving, Unmanned Aerial Vehicle (UAV) navigation, and gaming to a variety of domains to focus on cross-domain trends and constraints. Finally, in the last part of the review challenges published recently are discussed and modern learning based techniques are critically examined at the level of improving heuristic quality, adaptability and practical deployment. The methodology of this paper is presented in the following sections: Section 2 presents the techniques that allowed us to study the recruitment and screening of relevant studies. The basis of the algorithm, evolution, extension, applications, and comparison, as well as some disadvantages are presented in Section 3. Discussion of your findings is done in Section 4. Section 5 examines

the key research opportunities and future directions of models based on A*. This review is concluded in Section 6.

2. Search methodology

To ensure that this review is both timely and comprehensive in A* and many further A* derivative works, an appropriate structured search process was used. We conducted our search, referring to several peer-reviewed scientific databases: IEEE Xplore, Scopus, ScienceDirect, SpringerLink, and Google Scholar. Due to being the source of most relevant literature until now in robotics, AI and path planning, they were chosen as the most comprehensive literature sources. Primary and secondary keywords were used to search for the studies related to a question. These keywords were: “A* algorithm,” “A-star path planning,” “heuristic search,” “hybrid A*,” “D*,” and “grid-based search.” Relevant keywords were “mobile robots,” “autonomous vehicles,” “UAV navigation,” and “dynamic environment.” Boolean operators (e.g., AND/OR) were applied to narrow or extend the search (e.g., “A* algorithm” AND “autonomous navigation”). Since the implementation of real-time re-planning techniques and learning-based heuristics, new strategies for planning paths were developed quickly and defined the search between 1998 and 2025. For these, the search consisted of only peer-reviewed journal articles, conference papers, and high-quality review papers that specifically described the A* algorithm and its variants, algorithmic developments, or practical applications. We excluded unpublished theses, non-English studies, duplicate studies, and documents with ambiguities or incomplete means. The review was a multi-step screening and screening process. Early in the process, titles and abstracts were selected to consider the suitable ones. This time, and in step two, full texts were screened for contribution, amount of experiment, experiment depth, or theory or background. A combination of classical and recent implementations of path planning by means of A* to select relevant, credible studies was performed. The chosen final set of studies serves as the supporting context for this rich review considering these existing works and algorithmic advances and also novel applications, such as robotics, autonomous systems, and intelligent navigation for the continued generation of A*.

3. Background

3.1. Theoretical overview

A* is one of the most popular path planning algorithms in AI and robotics because of its remarkable efficiency and accuracy in obtaining the optimal path [1, 10, 11]. The algorithm assumes that a heuristic function estimates the residual distance to the goal, limiting the exploration as much as possible, due to improved decision-making through heuristic and informed searching, allowing a significant reduction in the number of nodes explored compared to previous search methods like Dijkstra [12, 13]. A* is suitable for multiple dynamic and stationary environments, and is flexible- scalable for large datasets. This algorithm depends on the selection of an appropriate heuristic function to enhance efficiency while still achieving the accurate navigation behavior of the agent in the environment [14, 15]. Several studies have proposed improved A* variants to enhance computational efficiency and energy consumption in mobile robot navigation [16, 17]. To clarify the operation of the algorithm, its structure and implementation steps are examined below in the following paragraph:

The algorithm contains a network of nodes that are essentially the path. They are divided into edges, an open list representing those nodes which have been discovered but not yet processed; a closed list representing those that have been processed or expanded and an evaluation function (f) that evaluates each node with a function known as F(n) where:

$$F(n) = G(n) + H(n) \tag{1}$$

where G(n) represents the actual cost from the starting point to node n, and H(n) represents the estimated cost from node n to the goal. This estimation is obtained using a heuristic function such as Euclidean or Manhattan distance.

The effectiveness of the A* algorithm depends on the selected heuristic function used to guide the search process. Therefore, a critical analysis of commonly used heuristics is essential to assess their impact on performance, optimality, and computational efficiency as listed in Table 1.

Recent studies have explored adaptive heuristic strategies to improve real-time performance in unknown environments [18]. The heuristic comparisons listed in Table 1 are reported from previous studies [14, 15, 19, 20], as well as recent learning-based heuristic approaches [21, 22].

Traditional admissible heuristics, such as Manhattan and the Euclidean distance, exhibit optimality but do not achieve efficient scaling of their algorithms in complex or dynamic environments as the analysis shows. The weighted heuristic achieves quick computational efficiency at the price of solution optimality. Currently, adaptive and efficient learning-based and hybrid heuristic methods are applied in complex settings. But they are dependent on large training data and lack a guarantee of admissibility, making them unsuitable for

Table 1. Important comparison of methods used in A* path planning.

Heuristic function	Admissibility	Relative execution time	Relative number of explored nodes	Path optimality	Environment suitability	Critical limitations
Manhattan distance	Admissible	Low	High	Optimal	Grid-based, 4-connected maps	Performs poorly in diagonal or continuous environments
Euclidean distance	Admissible	Low - moderate	Moderate	Optimal	Continuous and geometric spaces	Ignores obstacles and terrain complexity
Diagonal distance	Admissible	Low	Moderate - low	Optimal in 8-connected grids	8-connected grid maps	Limited improvement in dense obstacle environments
Weighted heuristic	Non-admissible	Very low	Low	Suboptimal	Large-scale or time-critical systems	Sacrifices optimality for speed
Learning-based heuristic	Often non-admissible	Moderate - high	Low - moderate	Near - optimal	Dynamic and complex environments	Requires training data and lacks theoretical guarantees
Hybrid heuristic (A* + Machine Learning (ML))	Adaptive / often non-admissible	High	Very low	Near - optimal	Autonomous vehicles and intelligent robots	High computational and training cost

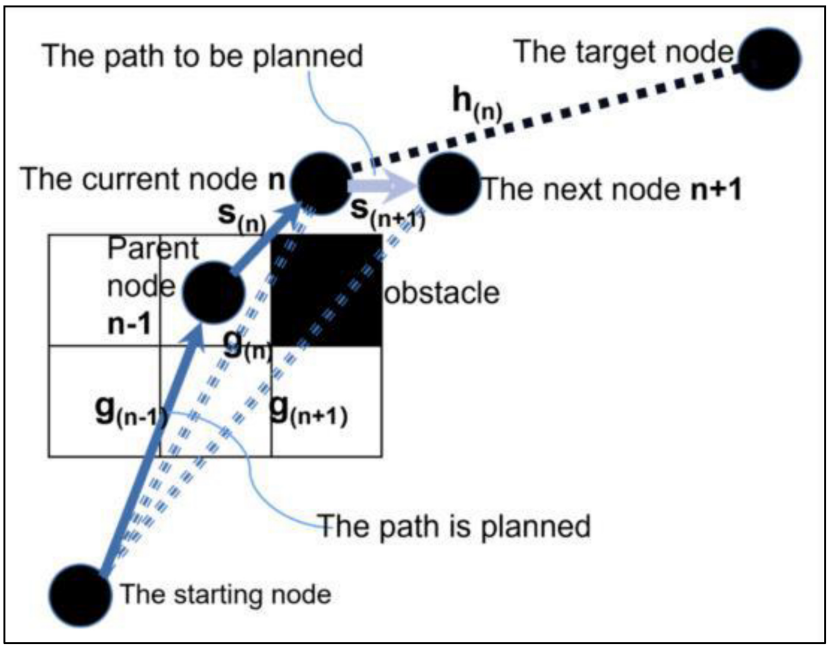

Fig. 1. Schematic representation of the mechanism of the A* algorithm [23].

use in safety-critical applications. Therefore, studies related to adaptive heuristics able to trade-off efficiency, optimality, and robustness are gaining prominence. This algorithm starts off with an open list containing the initial node. At each iteration, the lowest value of the first node F(n) is selected from the open list. The next node (neighboring nodes) of the processed node is checked. If the neighbor is not on the closed list, it can be added into the open list. G(n) and H(n) values are updated according to the new path that is being evaluated, repeating until the target node is reached [13]. Fig. 1 shows the general structure and mechanisms of the A* algorithm in path search. The evaluation function of each node is computed from Eq. (1). G(n) is the total cost from the starting node all the way to the current node. For the average required distance to be achieved from the goal, H(n) denotes the heuristic estimate of the remaining distance to the goal. The arrows show the direction of search between the nodes, and the black area is the barriers that the algorithm skips to get the best path in the solution.

*3.2. Traditional and recent approaches based on A*s*

The first A* formulations known as classical or traditional A* were designed to perform a mapping algorithm for static and grid-based environments [14]. Fixed heuristic concepts and exhaustive expansion of nodes led to these approaches which caused a lot of computationally expensive and RAM-consuming solutions, especially in large-scale environments. Recently, A*-based methods have concentrated on computational efficiency that are better adapted and scalable. Some of the recent variants, such as hybrid A*, learning-enhanced A* and real-time re-planning methods include dynamic heuristics, environment awareness and the incorporation with other methods such as Dynamic A* (D*), D* Lite and ML models [24, 25]. Fast re-planning, low node exploration, optimized performance in dynamic and partially known surroundings and less complexity make them better used as a real robot model approach. In terms of performance, contemporary work uniformly demonstrates

Table 2. Comparison between traditional and recent A*-based approaches.

Aspect	Traditional/Classical A*	Recent A*-based Approaches
Environment	Static, known maps	Dynamic or partially known environments
Heuristic type	Fixed, admissible	Adaptive, weighted, or learned
Re-planning ability	Limited	Strong, especially in D* and D* Lite
Path realism	Grid-constrained	Better for kinematic and smooth paths
Node exploration	Usually high in large maps	Reduced through improved guidance
Execution speed	Moderate to slow in complex maps	Faster in many real-time cases
Memory usage	High in large search spaces	Often improved, but depends on method
Main limitation	Poor scalability in dynamic settings	Higher complexity, possible loss of optimality

better execution time, memory cost and path tuning than classical A* [17, 20]. They attain these performance upgrades by eliminating unnecessary node expansions, by more appropriately guiding the search and by allowing for online re-planning in the presence of alterations of the environment. Table 2 shows the primary methodological and experimental differences between classical and newer A* approaches. Classical A* is most suitable for grid-like, known environments where fixed admissible heuristics can provide solutions that make optimal decisions. However, if the environment is large, dynamic, or, in some cases, poorly defined, the limitations of classical A* are glaringly apparent: intensive memory usage, larger node expansion, and slower computation time. Moreover, with the rise of Hybrid A*, D* Lite, and learning-enabled A* schemes, all of these introduce incremental re-planning, or adaptive heuristics/movement restrictions for greater flexibility. These methods typically minimize redundant exploration and improve response time in real-time applications. Some of those advances do involve trade-offs, not least an added burden on implementation, extra expenses on parameter tuning or algorithm training, or sacrificing some level of optimality guarantees. From the fact of both methods we can observe that conventional A* is suitable for restricted and controlled search space, but the modern approach of A* based search is suitable for dynamic large-scale and real world autonomous exploration related problems.

3.3. Historical development and variants

The A* algorithm has evolved a lot since it was introduced in 1968 [14]. The D* method was introduced by Stentz in 1994 where robots are able to reconfigure paths automatically in response to the discovery of a new obstacle, and thus to be used in dynamic situations [26]. D* Lite was introduced in 2002 for a more efficient and computationally lightweight algorithm [27]. Then algorithms like Anytime Repairing A* (ARA*) and Anytime A* were introduced, which provide fast first responses that improve through time [28]. For instance, in the Theta* algorithm, such as what was proposed previously in 2007, the path smoothness can be improved by adding straight lines between nodes in place of the grid movement [29]. This has been primarily the aim for recent studies in hybrid A*-level versions with machine learning-based approaches and neural network-based tools [30] to tackle the more challenging aspects of the environment. Fig. 2 illustrates the classification for path planning algorithms.

This history shows how the A* algorithm changed from a basic fixed model to a flexible and efficient system that can handle the needs of today's robotics, self-driving navigation, and smart toys. Table 3 presents the development and different versions of the A* algorithm.

The A* algorithm is a familiar algorithm as it yields suitable path-prediction at a moderate rate even in challenging scenarios. It is an essential algorithm in smart navigation

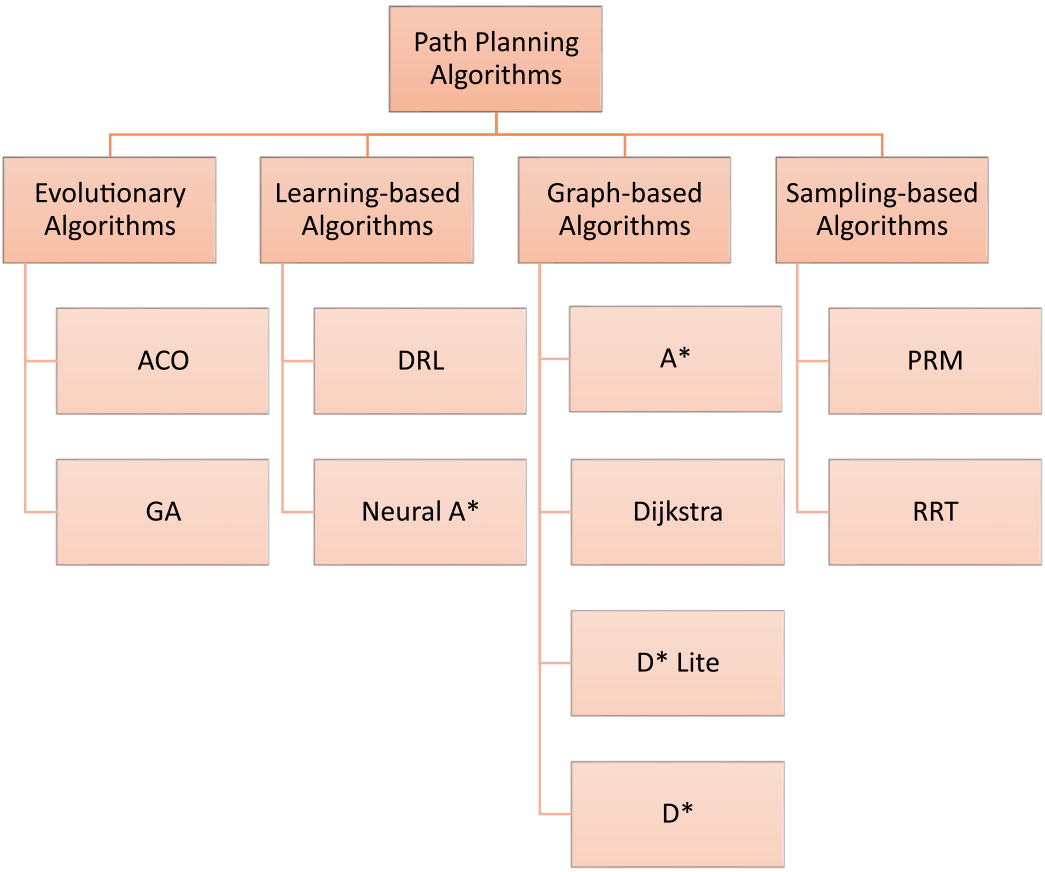

Fig. 2. Classification of path planning algorithms.

Table 3. Time evolution and improvements in the A* algorithm.

Year	Algorithm/Version	Basic Idea	Improvements/Features	Reference
1968	A*	A Search that uses a heuristic method to estimate the remaining distance	The method Finds the optimal path with a balance between actual and heuristic costs	[14]
1994	D*	Automatic re-planning in dynamic environments	The method deals with new obstacles without recalculating the entire path	[26]
2002	D* Lite	An improved and lighter version of D*	The method has a higher computational efficiency and ease of implementation	[27]
2003	ARA* / Anytime A*	Obtaining a fast, incrementally improvable solution Anytime	Introducing a new algorithm for improving solutions over time	[28]
2007	Theta*	Smoothly path planning using straight lines between nodes	Minimizes mesh movements and makes the path smoother	[29]
2015	Hybrid A* + ML	Combining the A* algorithm with machine learning techniques	Ability to adapt to complex environments, improve planning efficiency, predict obstacles, and produce smarter paths	[30]
2022	Hybrid Swarm-Based Neural Network (HSBNN)	Combining A* with bio-inspired neural networks and GA	Improve performance in planning dynamic paths, adapt to environmental changes, and update neural activity values for robot guidance	[31]

systems, robotic robot systems as well as video games, so the accurate and autonomous path-planning is one of the most important factors for the accuracy of these algorithms. The classical A* algorithm has been reported to do well in well-defined and static situations, for example it's shown in one study can run time 229 ms, nevertheless, despite this satisfactory speed, classical A* cannot accommodate dynamic re-planning; any change in the environment has to start from the entire route, which raises the computation load for real-time applications. Classical A* is also characterized by expensive node exploration and high memory usage with increasing volume of search space. Distinguishing from it, D* Lite is mainly designed for dynamic or partially unknown situations in which it enhances search efficiency by regenerating only the parts of the path previously computed that are affected from the previous search [32]. Consequently, D* Lite has lower re-planning cost, less re-expansion of nodes, increased responsiveness in changing environments with moderate memory consumption. Hybrid A*, on the other hand, generalizes the classical A* model to kinematic limitations and to produce more realistic and smoother vehicle and robot trajectories for autonomous vehicles and mobile robots, but they have a penalty: increased computation complexity, longer running-time, larger number of nodes, and more memory usage since the algorithm must consider additional motion states (heading angle and turning radius). Thus, comparing these options revealed that no type is better than another in every case all the while: Classical A* is always optimal in static path-planning tasks, considering its simplicity and optimality, D* Lite excels in dynamic problems demanding fast re-planning, and Hybrid A* is preferable for applications that demand kinematically feasible and smooth paths, even though the computational expense is higher. Comparing the A* algorithm main parameters, Table 4 makes clear the most relevant and crucial trade-offs between the optimality, computational expense, and environmental fit of the A* algorithm.

Table 4 critically compares the key variables in a model A* algorithm's path planning. The comparison and the analysis were previously indicated based on the evidence and the studies presented [14, 24–27, 29, 33]. In addition, with quantitative data, Table 5 summarizes representative experimental performance metrics found in existing literature to provide the comparison discussion with tangible support. Classical A* is still appealing because it guarantees optimality when an admissible heuristic is used and performs well in static, structured environments. Nevertheless, lack of dynamic re-planning restricts the applications in changing environments. While D* Lite avoids this problem by reducing repeated search effort and improving responsiveness, Hybrid A* improves the feasibility of a forward trajectory under motion constraints at the cost of more expensive computation and memory. Compared with other approaches, A* is in the middle ground, as it is usually more efficient than uninformed graph search, but isn't as flexible as incremental planners, and can be less readily scalable than some sampling-based approaches. This provides evidence that no one algorithm is the best in every case and that the choice of algorithmic method depends on the tradeoffs between optimality, responsiveness, scalability, and computational cost.

The numbers shown in Table 5 come directly from separate experiments. Even though these tests were done in different conditions, the results clearly show that classical A* works well in unchanging environments, D* Lite makes re-planning faster in changing situations, and Hybrid A* uses more computing power and memory because of movement limits.

3.4. Applications of A* algorithm

Due to their safety and minimal collision, A* and the variants thereof have found significant use since their introduction in the mobile robotics and autonomous system

Table 4. Critical comparison between the most important variables of A* algorithm.

Algorithm	Environment Type	Optimality	Computational Cost	Memory Consumption	Main Strength	Critical Limitation	Suitable Applications
Standard A*	Static known	Optimal with admissible heuristic	High in large maps	Very high	Guarantees optimal path with simple implementation	Poor performance in dynamic or large-scale environments	Grid-based mobile robots, games
D*	Dynamic, partially known	Optimal	Very high	High	Real-time re-planning capability	Computationally expensive with frequent updates	Outdoor mobile robots
D* Lite	Dynamic, unknown	Near optimal	Lower than D*	Moderate	Efficient re-planning with reduced complexity	Still sensitive to map size	Autonomous navigation
ARA*	Time constrained	Suboptimal - Optimal	Moderate	Moderate	Fast initial solution with progressive improvement	Not ideal for strict real-time systems	Real time robotic systems
Theta*	Static, grid-based	Near-optimal	Moderate	Moderate	Produced smoother and shorter path	Limited benefit in high cluttered environments	Simulation, games
Hybrid A*	Continuous, kinematic-constrained	Near-optimal	High	High	Handles vehicle kinematic and smooth trajectories	High computational burden	Autonomous vehicles
AI-based hybrid A*	Dynamic, complex	Near-optimal	Very high	Very high	Adaptive and predictive planning	Training cost and lack of generalization	Intelligent robot, UAVs

Table 5. Reported quantitative performance metrics from independent experimental studies.

Algorithm	Execution Time	Memory	Node Expansion	Dynamic Re-planning	Reference
A*	229 ms	Low	Not reported	No	[34]
D* Lite	Reduced runtime vs repeated A*	Moderate	Reduced re-expansion	Yes	[35]
Hybrid A*	Higher than classical A*	High	Increased due to kinematic constraints	Limited	[36]

paradigms. With similar principles in ground robotics, derivatives such as Field D* and improved A*-based approaches have been used to enhance the smoothness of trajectories and navigation reliability in complex terrains [37, 38]. Hybrid A* combines discrete search within vehicle kinematic constraints for the autonomous driving paradigm, making trajectory generation attainable and smooth in the urban environment. A* 3D extensions have been used in aerial robots to map UAV path and avoid obstacles in complex airspace [39, 40]. In simulation and gaming settings, performance is important, and A* is applied where it runs in real time [41, 42]. These cases illustrate that A*-based methods can be

Table 6. Provides a comparative analysis between A* and other representative path planning algorithms based on commonly reported performance indicators.

Algorithm	Optimality Guarantee	Scalability Level	Dynamic Environment Support	Real-Time Capability	Relative Computational Load	Typical Use Cases
A*	Optimal	Medium	No	Limited	High	Grid-based planning
Dijkstra	Optimal	Low	No	No	Very high	Graph shortest path
D* Lite	Near-optimal	Medium–High	Yes	Yes	Medium	Dynamic re-planning
RRT	Probabilistic	High	Yes	Yes	Low	High-dimensional spaces
PRM	Probabilistic	High	Partial	Limited	Medium	Multi-query planning
GA	Suboptimal	Moderate	Partial	Low	Very high	Flexible optimization; slow for real-time
DRL-based methods	Suboptimal	High	Excellent	High	Very high	Adaptive control needs large training datasets

effectively applied to static, dynamic, and continuous planning scenarios. A*-based planning is supported in several programming libraries. Python libraries (NetworkX, PythonRobotics for graph-search implementation) and Robot Operating System (ROS) navigation stacks that combine A* and D* variants for physical robotics systems are also present. Tools such as Gazebo are able to support experimental verification from simulation platforms. This makes deployment and enhances practical applicability with these tools.

3.5. Comparison with other algorithms

Several path planning approaches have been designed to overcome the shortcomings of classical A*. The variations in these methods are optimality, scalability, dynamic adaptability, and computational requirements. To structure these trade-offs.

The comparison in Table 5 is critical and comparable to other researches in [43–48]. A* combines optimality with practicality, as can be seen in Table 6, and is more efficient in search for applications based in a static grid than naive algorithms with advice heuristics; for example Dijkstra. Importantly, unlike sampling-based methods such as RRT, PRM, etc., this is more brittle to dynamic environments than D* Lite, and it is not suitable for high-dimensional continuous spaces. Although the contributions made by this paper, and particularly its broad conclusions and suggestions regarding enhancing current RQ schemes and approaches to efficient grid architecture, are important in showing the inherent biases of this approach. Unlike classical search algorithms, learning-based and bio-inspired methods are usually also more flexible but computationally expensive, and typically do not include the deterministic optimality guarantees from the classical search algorithms in their algorithms. These differences confirm that in structured scenarios, we can still obtain a very high efficiency using A* and that the remaining two approaches are more desirable when we must choose between flexibility or scalable capabilities. The time complexity of the classical A* algorithm is usually O(bd) from a computational techniques perspective whereby b is the branching factor and d the search depth. Such complexities

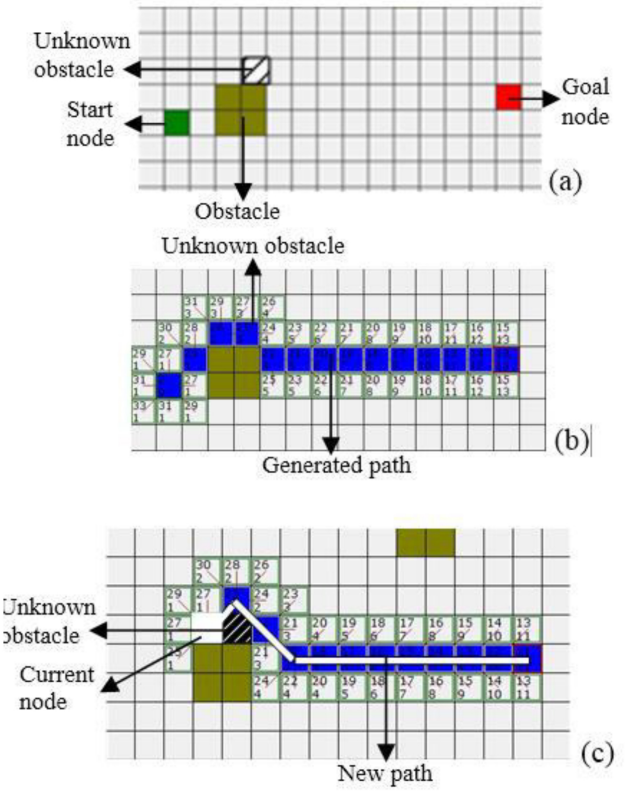

Fig. 3. Simulation results of A* path re-planning algorithm [48].

have notable impacts on planning time for mapping robotics, especially regarding large-area/high-resolution maps on-device. This results in extra memory consumption for nodes as the size of the environment increases, where expansion is exponentially higher. As nodes increase, this results in delayed response as the response time takes longer. For example, D* Lite is an incremental algorithm that ensures redundant computations are eliminated; by not doing such calculations and changing the environment only once, it only updates the affected nodes, which enhances real-time behavior of the algorithm users for dynamic navigation of the robot. Such a complexity is not only a conceptual characteristic, but also an important problem regarding real-world robot path planning applications. Experimental research is performed by [48] to illustrate the dramatic difference in the re-planning mechanism of the A* algorithm and the D* Lite algorithm against obstacle dynamization. Fig. 3, modified from [48], shows that A* works on the condition of an incoming obstacle in a predicted path. Here, the algorithm drops all prior search data and recalculates from where the robot currently operates to this target. Even though this guarantees a positive fix, it is redundant. By comparison, D* Lite, visualized in Fig. 4, exhibits a more effective incremental strategy. When the obstacle is observed, the algorithm keeps search data and only updates node costs directly influenced by the change, and finally only deploys this update to reverse the path. This greatly minimizes redundant computing time and exemplifies a novel design philosophy which makes D* Lite viable for dynamic complex environments.

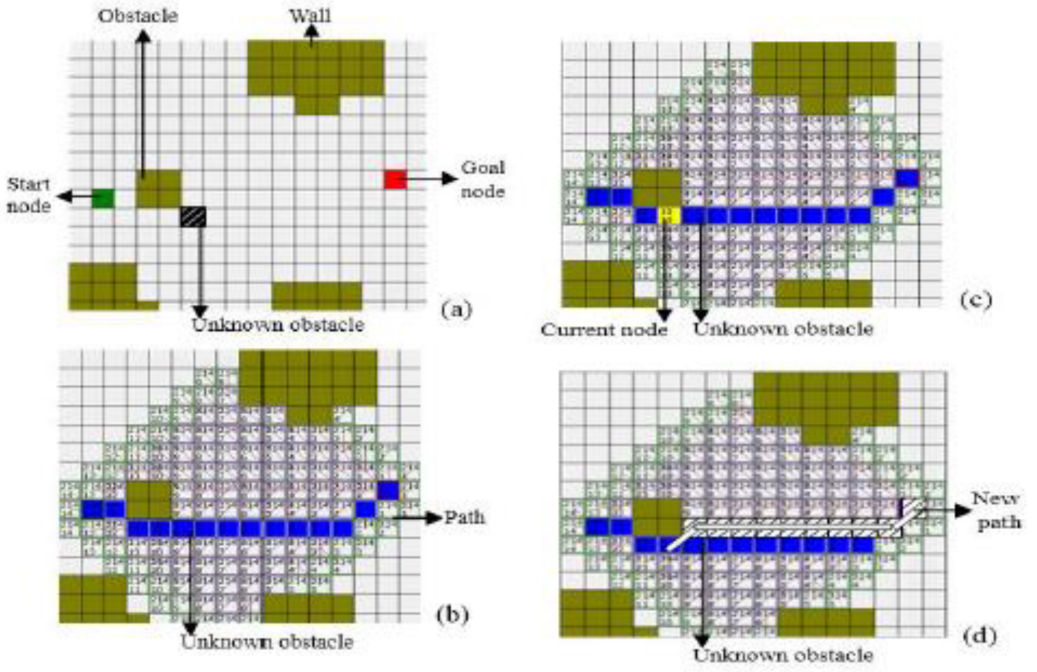

Fig. 4. Simulation results of D* Lite path re-planning algorithm [48].

This comparison clearly shows the main difference between full re-planning and step-by-step re-planning methods, giving a clear reason why D* Lite works better in changing robot tasks.

3.6. Challenges and critical limitations of A variants*

While the A* algorithm and its related algorithms have been widely applied for decades, there are still some major limitations to both the simple A* algorithm and advanced A* algorithm. These limitations are determined by memory consumption, scalability, real-time responsiveness, environmental assumptions, and heuristic dependencies. In addition, A* is memory-bounded as for classical A*, with open and closed lists in the search, it implies there is considerable computational overhead at scale [49]. This is especially hard to scale, in large dimensionality, or very connected search spaces, which indicates search spaces for which many computations are needed and also this is very expensive. Also, its most notable criticism is its assumptions about a well-defined system — a static (and fully known) environment. Standard A* has also not held for dynamic or partially described use cases. D* and D* Lite are also available and use incremental re-planning to handle environment updates, but they are expensive to compute due to frequent updates or increasing map complexity. Responsiveness is another challenge. Algorithms such as ARA* give faster answers in first versions and much better optimization, but at the same time, they can only be used for strict real-time systems (as far as we know). Additionally, Hybrid A* imposes vehicle movement restrictions in order to create feasible mobility for autonomous driving applications, but this realistic feature leads to higher search complexity and computational cost, particularly in unstructured networks. Also, how effective the heuristic functions are greatly affects the overall performance of the A*–based methods. A heuristic would

similarly result in low efficiency and performance in search optimization with poor design [51]. More sophisticated versions mitigate some of the drawbacks of classical A*, yet trade-off of optimal/constrained system, computational efficiency/capacity, scale, and dynamic adaptability remains an open source gap in intelligent navigation systems. It is known from the literature to enhance the classical A*, such as that the classical A* was not always able to tackle the problem. Utilization of hierarchical planning and graph pruning, as well as using implementations with memory constraints such as Simplified Memory Bounded A* (SMA*) can reduce memory consumption. Multi-resolution grids and sampling-based hybrid frameworks mitigate scalability concerns. In the case of dynamic environments, incremental search algorithms like D* and D* Lite remove the overall re-computation effort. Real-time performance should be improved using Anytime algorithms such as ARA* that give quick initial solutions followed by a stepwise refining process. In addition, algorithms for the heuristic calculation by learning have been previously introduced, for their ability to get better search performance without affecting the feasibility of a route. These approaches illustrate how much of A*'s constraints can be removed through adaptive and hybrid planning modes.

4. Discussion

Quantitative (Tables 3 to 5) data identify various and relevant performance trade-offs that guide an A*-based variant for individual applications. In static and unchanging environments, Classical A* performs well; however, this advantage is accompanied by 'High' memory usage (Table 3) and the absence of dynamic re-planning capabilities (Table 5), which restricts its applicability to well-defined static situations. D* Lite is a direct solution to this limitation, allowing real-time re-planning to give a 'Reduced runtime vs repeated A* execution time' (Table 4). It is not easy to explain, but this adaptability does not come without somewhat more computational overhead: this requires a compromise: providing environmental responsiveness at the cost of algorithmic complexity. The pricing of the actual physics is most evident in the cost of fitting A* in hybrid A* is best illustrated with hybrid A*; it is necessary to make possible the trajectories necessary for the real physics of autonomous driving, however, it is expensive in terms of 'Higher than classical A* execution time' and 'High' memory usage (Table 4), directly as a by-product of extending the search into a continuous state space. Taken together, these results support the fact that there is no one single algorithm in general where there is no ultimate preference of a variant, the best one is a tradeoff between optimality, computation cost and environment adaptivity. The review of present contemporary techniques presents a typical picture of trends in research: the requirement that systems be smart and adaptive. To overcome the tough trade-offs mentioned above, increasingly the research community is trying to combine both A* and AI models. A comparative perspective on these promising hybrid methods is provided in Table 7.

The comparative critique shown in Table 7 is accompanied by the recent literatures, which discuss hybrid and AI-integrated A* schemes [2, 10, 26, 31, 33, 34, 51]. The qualitative comparison in Table 7 (not that there are objective numerical parameters to represent, as they are experimental) shows a clear, yet important pattern: high adaptivity in more complex environments is costly. For example, although A* with reinforcement learning achieves 'High' adaptability, it suffers from computational costs of 'Very high' and has no optimality guarantees. Similarly, the use of Large Language Models (LLM) approaches provides innovative context-based guidance but still are computationally costly and experimentally immature. This means that there is really more to the project in terms

Table 7. Critical comparison of hybrid A* and AI-based path planning approaches.

Hybrid approach	AI technique used	Main improvement achieved	Adaptability to dynamic environments	Computational cost	Critical limitation
Hybrid A* + neural network	Supervised, deep neural network	Improved heuristic estimation and smoother path	Moderate-high	high	Requires large labeled datasets and retraining for new environments
A* + Reinforcement learning	Deep Reinforcement learning	Adaptive decision-making and online re-planning	High	Very high	Training instability and lack of optimality guarantees
A* + learning-based heuristic	Heuristic learning model	Reduced node expansion and faster convergence	Moderate	High	Heuristic admissibility is not guaranteed
A* + Bio Inspired neural networks	Bio-inspired, spiking neural network	Improved real-time response and robustness	High	High	Complex implementation and limited real-world validation
Hybrid A* + Evolutionary algorithms	GA, ACO	Enhanced global exploration capability	Moderate	Very high	Slow convergence and poor real-time suitability
A* + large language models	Large language models	Context-aware heuristic guidance	Moderate	Very high	High computational demand and experimental maturity

of how to make AI not only highly effective, but also to have the strongest theoretical foundations which can actually operate. The attention needs to move away from just incorporating AI but rather to balance the use of AI with the real system's limitations to perform in realistic conditions such as computing load, requirements of training data and requirements to have confidence in guarantees for safety. Therefore, the future of path planning must produce hybrid systems that can dynamically select according to live operation requirement, which allows them to handle the fundamental trade-offs involved in the development of the A* algorithm. While the admissible heuristic guarantees the optimality of A*, in practice, the performance is often restricted by high memory consumption and high levels of node expansion in large or dynamic networks. Published results demonstrate that these shortcomings can be drastically reduced due to problem-specific enhancements. However, for large-scale grids, search-space reduction methods have a significant effect — such as the one employed in a recent large-map work on 10,000×10,000 grids, where optimized A*-based methods could offer a 93% reduction of search time and a 60% reduction in memory consumption compared to ordinary A* while upholding path optimality. In dynamic environments, incremental replanning methods provide clear advantages over replanning from scratch. It has been reported that D* Lite outperformed repeated A* by more than a factor of four in vertex expansions on nonuniform terrains [7], and also noted prior evidence that D*-type replanning can outperform repeated A* by about one order of magnitude or more in unknown terrain. Additional evidence showed that Delayed D*, a refinement of D*-style replanning, achieved equivalent paths while requiring about half the computation time of D* [6]. For real-time applications, anytime and bounded-suboptimal search can further reduce computational burden: in replanning experiments, truncated incremental/anytime search variants reported about 4× to 11× runtime speedup over

the best corresponding optimal methods, depending on the change rate. Hybrid methods also improve practical performance in complex robotic scenarios. For example, a recent improved A* Dynamic Window Approach (A*-DWA) framework reported 19.1% lower computation time in simple environments and 20.1% lower execution time in complex environments, while also increasing robot speed by about 24-25%. These results show that the problems of the classic A* algorithm can be greatly reduced when the algorithm is adjusted to fit the search area, how often it updates the path, and the computing power available.

5. Future research directions

Despite the progress implemented by the A* algorithm and its derivatives over the decades, there is still room for improvement, more so in dynamic and complex environments. Recent works show the need to integrate advanced AI methods, like deep learning, to improve A* performance and enable adaptation to real-time environmental changes. Using adaptive learning models, we can learn helpful rules from past experience and update decisions quickly. Further work is expected to improve search speed by using smaller data formats and multiple processors simultaneously to search large maps faster and use less memory. A potential next trend would be to pursue hybrid A* versions, where the algorithm can be combined with methods, such as random exploration techniques, e.g. RRT or D* with directed search, to leverage their complementary advantages. Moreover, the rapid development of swarm robotics and autonomous vehicles may enhance cooperative path planning, enabling modified A* algorithms to coordinate the movements of multiple agents in dynamic multi-robot environments, that is, modified A* algorithms can provide coordination of movement of several robots in multi-agent environments simultaneously. Therefore, to develop future A* methods with more intelligent flexibility, a system that learns to adapt in real time, plan and so on, while maintaining their core strengths in accuracy and computational efficiency represents the next step.

This paper proposes a future-proof intelligent framework for the A* algorithm that combines the classical A* algorithm with deep learning models. In this framework, the exploratory function is dynamically learned and updated by a neural network using real-time environmental feedback, enabling adaptive and predictive path planning. This integration represents a step toward self-learning navigation systems while retaining A*'s core strengths in accuracy and computational efficiency.

6. Conclusion

It has been a long time that the A* algorithm and its variants, including the A* search algorithm, had been around, so this study has looked at the A* algorithm at its original and extended variants from the beginning with a critical lens. Based on this analysis together with cross-sectional studies on the quantitative and qualitative fronts, an overall trend was established: There is no such thing as a one-size-fits-all A* Algorithm. Rather, evolution involves constructing specialized "toolboxes," in which a particular version receives better scores on different path-planning tasks of interest. The conventional A* will still be the best design under simple conditions, while under dynamic conditions it's better—as the incremental re-planning principle is driven by D* Lite. Hybrid A*, however, is essential for use cases for which certain paths need to be performed in continuous space, and is computationally intensive. In this review, hybrid approaches emerge as the path-planning mechanism of choice for the future. Evolution is no longer simply

about replacing old algorithms with new ones, but smartly combining those two. An optimal path might be to integrate systematic searches' advantage with the adaptability and flexibility of ML to dynamic scenarios. Algorithm selection is strategic and calls heavily to the trade offs between optimization, computational overhead, flexibility, and physical system limitations. However, huge challenges persist despite progress. A huge challenge is the high computational burden and data dependence in AI-enhanced models, restricting an extensive usage both in a deep learning context and in embedded/real time systems. In addition, many learning-based approaches fail to offer theoretical protections (optimization and completeness, for example) which prevents their integration into critical tasks where complete security is needed. Accordingly, the adaptive models would be developed from now to develop adaptive frameworks for selection of the best planning strategy based on real-time operating conditions in future for the smarter, more modular and efficient navigation systems that can cope with problems of real-world complexity.

Acknowledgment

None.

Authors' contribution

Saleel H. Abood: Conceptualization, literature review, article selection, methodology design, visualization, and writing original draft. Hussein M. H. Al-Khafaji: Validation, critical review, writing, review, and editing. Mohanned M. H. Al-Khafaji: Methodology guidance, writing, review, and editing.

Conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Data availability

No data has been used in this paper.

References

1. A. Madridano, A. Al-Kaff, D. Martin, and A. de la Escalera, "Trajectory planning for multi-robot systems: Methods and applications," *Expert Systems with Applications*, vol. 173, Jul. 2021, Art. no. 114660, doi: 10.1016/j.eswa.2021.114660.
2. S. Meng, Y. Wang, C.-F. Yang, N. Peng, and K.-W. Chang, "LLM-A*: Large language model enhanced incremental heuristic search on path planning," in *Proc. Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, Florida, USA, pp. 1087–1102, doi: 10.18653/v1/2024.findings-emnlp.60.
3. Y. Yan, "Research on the A star algorithm for finding shortest path," in *Proc. 6th Int. Conf. on Mechatronics, Control, and Electronic Engineering (MCEE 2023)*, Dallas, USA, pp. 154–161, doi: 10.54097/hset.v46i.7697.
4. E. Papadopoulos and D. A. Rey, "The force-angle measure of tipover stability margin for mobile manipulators," *Vehicle System Dynamics*, vol. 33, no. 1, pp. 29–48, Aug. 2010, doi: 10.1076/0042-3114(200001)33:1;1-5;FT029.

5. P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968, doi: 10.1109/TSSC.1968.300136.
6. A. Stentz, "The D* algorithm for real-time planning of optimal traverses," The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-94-37, 1994.
7. S. Koenig and M. Likhachev, "Improved fast re-planning for robot navigation in unknown terrain," in *Proc. 2002 IEEE Int. Conf. on Robotics and Automation*, Washington, DC, USA, pp. 968–975, doi: 10.1109/ROBOT.2002.1013481.
8. M. Likhachev, G. J. Gordon, and S. Thrun, "ARA*: Anytime A* with provable bounds on sub-optimality," in *Proc. 17th Annual Conf. on Neural Information Processing Systems (NIPS 2003)*, Vancouver, Canada, pp. 767–774.
9. K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *Journal of Artificial Intelligence Research*, vol. 39, pp. 533–579, Oct. 2010, doi: 10.1613/jair.2994.
10. I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4–5, pp. 705–724, Mar. 2015, doi: 10.1177/0278364914549.
11. S. Chen, T. Feng, J. Li, and S. X. Yang, "Research on intelligent path planning of mobile robot based on hybrid symmetric bio-inspired neural network algorithm in complex road environments," *Symmetry*, vol. 17, no. 6, May 2025, Art. no. 836, doi: 10.3390/sym17060836.
12. H.-M. Zhang, M.-L. Li, and L. Yang, "Safe path planning of mobile robot based on improved A* algorithm in complex terrains," *Algorithms*, vol. 11, no. 4, Apr. 2018, Art. no. 44, doi: 10.3390/a11040044.
13. Y. Jin, M. Yue, W. Li, and J. Shangguan, "An improved target-oriented path planning algorithm for wheeled mobile robots," *Journal of Mechanical Engineering Science*, vol. 236, no. 22, pp. 11081–11093, Jun. 2022, doi: 10.1177/09544062221111.
14. D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," in *Proc. of the 1st Int. Symp. on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)*, Chicago, USA, Jul. 13–14, 2008, pp. 32–37.
15. M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, Aug. 2009, doi: 10.1177/027836490934044.
16. Y. Jianhua and L. Shang, "UAV 3D path planning and obstacle avoidance method," *Information and Control*, vol. 50, no. 1, pp. 95–101, 2021, doi: 10.13976/j.cnki.xk.2021.0095.
17. C. Lim, B. Li, E. M. Ng, X. Liu, and K. H. Low, "Three-dimensional (3D) dynamic obstacle perception in a detect-and-avoid framework for unmanned aerial vehicles," in *Proc. 2019 Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, 11–14 Jun., 2019, pp. 996–1004, doi: 10.1109/ICUAS.2019.8797844.
18. Y. Sazaki, A. Primanita, and M. Syahroyni, "Pathfinding car racing game using dynamic pathfinding algorithm and algorithm A*," in *Proc. 2017 3rd Int. Conf. on Wireless and Telematics (ICWT)*, Palembang, Indonesia, pp. 164–169, doi: 10.1109/ICWT.2017.8284160.
19. E. R. Firmansyah, S. U. Masruroh, and F. Fahrianto, "Comparative analysis of A* and basic theta* algorithm in android-based pathfinding games," in *Proc. 2016 6th Int. Conf. on Information and Communication Technology for The Muslim World (ICT4M)*, Jakarta, Indonesia, pp. 275–280, doi: 10.1109/ICT4M.2016.063.
20. Z. You, K. Shen, T. Huang, Y. Liu, and X. Zhang, "Application of A* algorithm based on extended neighborhood priority search in multi-scenario maps," *Electronics*, vol. 12, no. 4, Feb. 2023, Art. no. 1004, doi: 10.3390/electronics12041004.
21. D. Rachmawati and L. Gustin, "Analysis of Dijkstra's algorithm and A* algorithm in shortest path problem," in *Proc. 4th Int. Conf. on Computing and Applied Informatics 2019 (ICCAI 2019)*, Medan, Indonesia, pp. 1–8, doi: 10.1088/1742-6596/1566/1/012061.
22. J. Liu, S. Feng, and J.-H. Ren, "Directed D* algorithm for dynamic path planning of mobile robots," *Journal of ZheJiang University (Engineering Science)*, vol. 54, no. 2, pp. 291–300, Mar. 2020, doi: 10.3785/j.issn.1008-973X.2020.02.010.
23. A. Choudhary, "Sampling-based path planning algorithms: A survey," 2023, *arXiv:2304.14839*.
24. A. H. Al-Beaty, N. Al-Azzawi, and A. R. J. Almusawi, "Survey on path planning of mobile robot with multi algorithms," *American Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 90, no. 1, pp. 161–174, Oct. 2022.
25. Y. Zhu, W. Z. W. Hasan, and H. R. H. Ramli, "Deep reinforcement learning of mobile robot navigation in dynamic environment: A review," *Sensors*, vol. 25, no. 11, May 2025, Art. no. 3394, doi: 10.3390/s25113394.
26. L. Siag, S. Shperberg, A. Felner, and N. R. Sturtevant, "On parallel external-memory bidirectional search," in *Proc. Int. Symp. On Combinatorial Search*, Kananaskis, Alberta, Canada, 6–8 Jun., 2024, pp. 283–284, doi: 10.1609/socs.v17i1.31582.
27. Z. Xing, X. Zhu, and D. Dong, "DE-SLAM: SLAM for highly dynamic environment," *Journal of Field Robotics*, vol. 39, no. 5, pp. 528–542, Feb. 2022, doi: 10.1002/rob.22062.

28. K. Honda, R. Yonetani, M. Nishimura, and T. Kozuno, "When to replan? An adaptive replanning strategy for autonomous navigation using deep reinforcement learning," in *Proc. 2024 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Yokohama, Japan, pp. 6650–6656, doi: 10.1109/ICRA57147.2024.10611474.
29. S. S. Menon and A. D. Jagtap, "Anant-Net: Breaking the curse of dimensionality with scalable and interpretable neural surrogate for high-dimensional PDEs," *Computer Methods in Applied Mechanics and Engineering*, vol. 447, Dec. 2025, Art. no. 118403, doi: 10.1016/j.cma.2025.118403.
30. E. Futuhi and N. R. Sturtevant, "Learning admissible heuristics for A*: Theory and practice," 2025, *arXiv:2509.22626*.
31. Z. Yusuf, S. Mohamad, and W. S. W. Ibrahim, "Performance comparison of A* and Dijkstra algorithms with Bézier curve in 2D grid and OpenStreetMap scenarios," *Journal of Applied Engineering Design and Simulation*, vol. 5, no. 2, pp. 79–89, Sep. 2025, doi: 10.24191/jaeds.v5i2.94.
32. D. Numeroso, D. Bacciu, and P. Velickovic, "Learning heuristics for A*," 2022, *arXiv:2204.08938v1*.
33. Q. Wu *et al.*, "Real-time dynamic path planning of mobile robots: A novel hybrid heuristic optimization algorithm," *Sensors*, vol. 20, no. 1, Dec. 2019, Art. no. 188, doi: 10.3390/s20010188.
34. S. Meng, Y. Wang, C.-F. Yang, N. Peng, and K.-W. Chang, "LLM-A*: Large language model enhanced incremental heuristic search on path planning," 2024, *arXiv:2407.02511*.
35. S. Koenig and M. Likhachev, "D* lite," in *Proc. of the 18thNational Conf. on Artificial Intelligence*, Edmonton, Alberta, Canada, Jul. 28–1 Aug., 2002, pp. 476–483.
36. Y. Liu, X. Li, and Y. Tong, "Improved A* algorithm for mobile robot path planning," in *Proc. 6th Int. Conf. on Computer Information Science and Application Technology (CISAT 2023)*, Hangzhou, China, pp. 169–174, doi: 10.1117/12.3004049.
37. Z. Zhang, H. Fu, J. Yang, and Y. Lin, "Deep reinforcement learning for path planning of autonomous mobile robots in complicated environments," *Complex & Intelligent Systems*, vol. 11, no. 6, May 2025, Art. no. 277, doi: 10.1007/s40747-025-01906-9.
38. M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, Jan. 2014, doi: 10.1109/ACCESS.2014.2302442.
39. J. Huanf, Z. Liu, X. Chi, F. Hong, and H. Su, "Search-based path planning algorithm for autonomous parking: Multi-heuristic hybrid A*," in *Proc. 2022 34thChinese Control and Decision Conf. (CCDC)*, Hefei, China, pp. 6248–6253, doi: 10.1109/CCDC55256.2022.10033530.
40. X. Fu, Z. Huang, G. Zhang, W. Wang, and J. Wang, "Research on path planning of mobile robots based on improved A* algorithm," *PeerJ Computer Science*, vol. 11, Feb. 2025, Art. no. e2691, doi: 10.7717/peerj-cs.2691.
41. T. R. Wan, H. Chen, and R. Earnshaw, "Real-time path planning using adaptive heuristics in unknown environments," in *Proc. of Theory and Practice of Computer Graphics 2003*, Birmingham, UK, pp. 138–145, doi: 10.1109/TPCG.2003.1206941.
42. S. Venu and M. Gurusamy, "A comprehensive review of path planning algorithms for autonomous navigation," *Results in Engineering*, vol. 28, Dec. 2025, Art. no. 107750, doi: 10.1016/j.rineng.2025.107750.
43. W. Liao, F. Zhang, X. Wu, and H. Li, "Multi-Objective Path Planning for USVs Considering Environmental Factors," *J. Mar. Sci. Eng.*, vol. 13, no. 9, Sep. 2025, Art. no. 1705, doi: 10.3390/jmse13091705.
44. T. Takahashi, H. Sun, D. Tian, Y. Wang, "Learning heuristic functions for mobile robot path planning using deep neural networks," in *Proc. of the 29thInt. Conf. on Automated Planning and Scheduling*, Berkeley, California, USA, 11–15 Jul. 2019, pp. 764–772, doi: 10.1609/icaps.v29i1.3545.
45. S. Yin and Z. Xiang, "Multi-objective collaborative path planning for heterogeneous autonomous underwater vehicles in cluttered environments," *Swarm Evolutionary Computation*, vol. 100, Jan. 2026, Art. no. 102251, doi: 10.1016/j.swevo.2025.102251.
46. S. Yin and Z. Xiang, "Adaptive collision avoidance strategy for USVs in perception-limited environments using dynamic priority guidance," *Advanced Engineering Informatics*, vol. 65, May 2025, Art. no. 103355, doi: 10.1016/j.aei.2025.103355.
47. I. A. Bartsiokas, C. Ntakolia, G. Avdikos, and D. Lyridis, "Intelligent multi-objective path planning for unmanned surface vehicles via deep and fuzzy reinforcement learning," *J. Mar. Sci. Eng.*, vol. 13, no. 12, Nov. 2025, Art. no. 2285, doi: 10.3390/jmse13122285.
48. D. H. Kim, N. T. Hai, and W. Y. Joe, "A guide to selecting path planning algorithm for automated guided vehicle (AGV)," in *Proc. of the Int. Conf. on Advanced Engineering Theory and Applications (AETA 2017)*, Ho Chi Minh, Vietnam, pp. 587–596, doi: 10.1007/978-3-319-69814-4_56.
49. F. Duchoň *et al.*, "Path planning with modified A star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014, doi: 10.1016/j.proeng.2014.12.098.
50. M. Przybylski and B. Putz, "D* extra lite: A dynamic A* with search-tree cutting and frontier-gap repairing," *International Journal of Applied Mathematics and Computer Science*, vol. 27, no. 2, pp. 273–290, Jun. 2017, doi: 10.1515/amcs-2017-002.
51. Y. Chen, Y. Liu, and W. Xu, "Improved hybrid A* algorithm based on lemming optimization for path planning of autonomous vehicles," *Appl. Sci.*, vol. 15, no. 14, Jul. 2025, Art. no. 7734, doi: 10.3390/app15147734.