Review of Hybrid Path Planning Techniques for mobile robots: Integration between AI Techniques and Traditional Methods in known environments

Submitted: 15th July 2025; accepted: 1st October 2025

Mohamed Abdelghafar, Hazlina Selamat, Nurulaqilla Binti Khamis, Anas Aburaya, Mohd Taufiq Muslim

DOI: 10.14313/jamris-2026-017

Abstract:
Mobile robots require effective and secure path planning, especially in complex and dynamic environments. Traditional algorithms such as A, Dijkstra, and Rapidly-exploring Random Trees (RRT) offer dependable and mathematical solutions but face challenges regarding scalability, adaptability, and processing requirements in real-time applications. On the other hand, artificial intelligence (AI) techniques, such as reinforcement learning (RL) and neural networks (NNs) provide flexibility and quick decision-making but face challenges such as data dependency, optimal solution, and computational overhead. This review analyzes hybrid path planning methodologies that integrate traditional algorithms with AI techniques, utilizing the advantages of both to overcome their limitations. Hybrid approaches improve scalability, collision avoidance, and re-planning efficiency by integrating the accuracy and reliability of traditional techniques with the adaptability and learning capabilities of AI. This review categorizes and analyzes research to identify significant gaps and suggests future paths for enhancing hybrid path planning, offering insights for the development of more robust and intelligent navigation systems for mobile robots and autonomous platforms.*

Keywords: *AI, Robotics, Path planning, mobile robots*

1. Introduction

Mobile robots have gained significant attention in areas such as surveillance, agriculture, disaster management, and logistics. An important aspect of mobile robot activities is path planning, which is defined as identifying an optimal pathway from a start point to a destination while minimizing costs in terms of time, energy, and computational resources, and avoiding obstacles. The efficiency of mobile robot path planning is especially crucial in known environments where detailed 3D maps are available, allowing for more precise navigation and optimization.

Traditional methods of path planning, such as the A* algorithm, Dijkstra's algorithm and Rapidly-exploring Random Trees (RRT), have been widely used due to their reliability and mathematical accuracy. These algorithms have proven effective in producing collision-free paths in 2D and 3D environments. However, they often struggle with high computational complexity and lack adaptability in dynamic and complex scenarios [1].

Artificial Intelligence (AI) methods, such as machine learning, deep learning, and reinforcement learning, can enhance path planning performance. AI-based methods offer several advantages, such as learning from data, adaptation to evolving environments, and multi-objective optimization. Despite these strengths, AI methods alone may require extensive training data, high computational power, and may lack interpretability compared to traditional algorithms [2].

The combination of AI and traditional methods for mobile robot path planning presents a promising hybrid approach that leverages the strengths of both paradigms. By integrating AI's adaptability and learning capabilities with the robustness and interpretability of traditional algorithms, researchers aim to develop more efficient and versatile path planning solutions. This hybrid approach has the potential to address the limitations of both AI and traditional methods, particularly in known environments where comprehensive 3D maps provide a structured framework for navigation.

This review aims to explore the current state of research on the combination of AI and traditional methods for mobile robot optimal path planning in known environments. The review will focus on identifying the key AI techniques and traditional algorithms employed, examining their performance, and highlighting the challenges and opportunities associated with their integration. By combining the findings from existing literature, this review aims to provide a comprehensive understanding of the hybrid approaches to mobile robot path planning.

1.1. Fundamentals of Mobile Robot Path Planning

Mobile robot path planning is critical for autonomous moving, as it provides a secure, effective, collision-free path from the starting point to the destination. The difficulty of this task depends on factors such as the environment in operation (urban or rural), obstacles (static or dynamic), energy limitations, and the specific mission goals (surveillance, delivery, mapping, etc.).

The path planning sequence has three phases, environmental modeling, path searching and optimization, and, finally, path execution and adaptation. Environmental modeling is when a 3-dimensional map of the area is developed and obstacles are modeled similarly using occupancy grids and Voronoi diagrams; path search and optimization is when algorithms identify paths based on parameters such as length and safety; in path execution and modification, mobile robots need to change the path accordingly based on random obstacles and environmental changes [1].

Mobile robot path planning algorithms, both traditional, AI-based or hybrid, are crucial to enabling mobile robots to navigate safely and efficiently in heterogeneous environments. The algorithms are developed for path optimization by balancing objectives such as obstacle avoidance, energy efficiency, and mission-specific requirements that vary based on the use of the mobile robot. Path planning algorithms combine mathematical precision with adaptive functionality as they allow mobile robots to deal with static and dynamic obstacles, environmental changes, and accurate navigation of all types of missions. This holistic consideration of path planning stresses the need for algorithms to develop and adapt as new problems arise in autonomous movement.

1.2. Global and Local Path Planning

Path planning algorithms can be classified into global path planning and local path planning based on their operational scope and planning procedure [2]. Global path planning algorithms, such as Dijkstra's, A*, D* Lite, and Cell Decomposition, are engineered to calculate a path from a start point to the target point for a defined environment. These approaches use a complete or partial representation of the environment to determine an optimal or near-optimal path, accounting for all detected obstacles. These methods are especially efficient in static environments where all information of the environment is known ahead of time and all of what has been learned is used for complete planning [2].

Conversely, local path planning techniques focus on immediate navigation and avoidance of obstacles. Local path planning includes both traditional algorithms, such as Dynamic Window Approach (DWA) and Artificial Potential Field (APF), as well as AI methods, such as Machine Learning (ML) and Reinforcement Learning (RL). Local path planning algorithms are specifically designed to focus on the ability to make quick decisions, from the current state of the mobile robot and its sensors, without requiring all the environmental knowledge. Local path planners can be highly effective in dynamic or partially known environments where the mobile robot must respond quickly to an unexpected change or moving obstacle that was not known prior to encountering [2].

In practical mobile robot operations, the integration of global and local path planning methodologies is frequently beneficial [2]. The global planner will provide a planned long path to the target, and then the local planner can modify the mobile robot's trajectory considering immediate hazards and other changes in the environment. This hybrid method guarantees both optimality and adaptability, improving the mobile robot's overall navigation efficiency.

2. Research Methodology

Previous review papers have focused on algorithm reviews in general or on broad integrations of techniques such as [3–5]. However, this paper specifically reviews the integration between traditional path planning algorithms and artificial intelligence (AI) methods. Papers were retrieved exclusively from Scopus to ensure a comprehensive review. A combination of keywords and Boolean operators was used to maximize the search results. The keywords such as:

- "Mobile robot" AND "path planning" AND "known environment" AND "hybrid" AND "artificial intelligent" OR "AI"
- "Mobile robot" AND "path planning" AND "known environment" AND "combine" AND "artificial intelligent" OR "AI"
- "Mobile robot" AND "path planning" AND "known environment" AND "integrate" AND "artificial intelligent" OR "AI"
- "Robot" AND "path planning" AND "known environment" AND "hybrid" AND "artificial intelligent" OR "AI"

The search range was limited to papers published from January 2019 to December 2024. Using these keywords, a total of 105 papers were retrieved. By applying inclusion and exclusion criteria, 23 papers were identified that fit the requirements. Figure 1 illustrates the number of publications over the years. Some papers focused on the hybrid between two AI methods which fall outside the scope of this review.

The quality of each paper was evaluated based on clarity of objectives and methodology, relevance to hybrid path planning for mobile robots, and robustness of experimental or simulation results. Inclusion criteria ensured the review synthesized the studies discussing the hybrid methods integrating traditional and AI-based methods, and studies of mobile robot research in the known environment. Exclusion criteria ensured studies focused solely on either AI or traditional methods without integration, papers without any experimental or simulation results, and non-English papers. Overall, the review ensures a systematic and unbiased exploration of hybrid path planning methods, offering valuable insights for researchers and practitioners in the field.

3. Hybrid Approaches: Combining AI and Traditional Methods

Traditional path planning algorithms provide reliable and often optimal solutions in structured environments but face scalability issues and struggle with real-time adaptability in dynamic settings [2, 3]. In contrast, Artificial Intelligence (AI) techniques enhance adaptability and decision-making, allowing robots to learn from data and

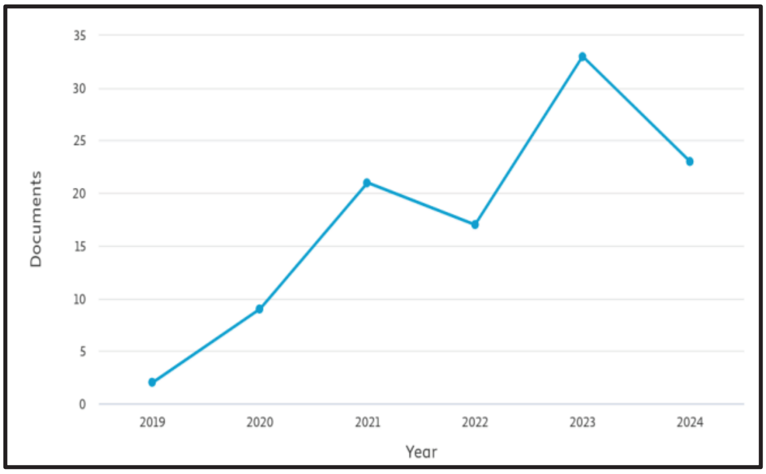

Figure 1. Line graph of number of papers vs. year published

respond to uncertainties, though they remain limited by computational demands, reliance on large datasets, and difficulty in guaranteeing global optimal paths [1, 3].

Hybrid approaches have emerged to combine the strengths of both paradigms. In such frameworks, traditional algorithms are employed as global planners to generate efficient baseline routes in known or partially known maps, while AI methods serve as local planners to adaptively handle real-time changes, dynamic obstacles, and uncertainties. This integration balances the efficiency and optimality of traditional methods with the adaptability and intelligence of AI, resulting in more robust and reliable path planning for mobile robots across diverse environments. Different hybrid models have been discussed, integrating different AI techniques with traditional methods to address the limitations of individual approaches. In the reviewed literature, 11 papers utilized RL as a local planner, including 3 that employed DRL, 8 papers used NN, and 4 utilized Fuzzy Logic, all combined with different traditional algorithms.

3.1. Integration with Reinforcement Learning (RL)

Reinforcement Learning (RL) enhances traditional algorithms by allowing mobile robots to adapt to dynamic environments through learning from interactions. RL is particularly effective for local path optimization, complementing traditional algorithms, which handle global path planning. In [6–8], A* is used with RL because A* provides optimal paths in static environments, but RL helps refine those paths in real time when dynamic obstacles or sudden changes occur. This integration is useful in cases where environmental conditions are semi-known, but real-time changes require constant path adjustment.

On the other hand, in [9, 10] RRT with RL is used because RRT excels in exploring large, unstructured environments, but it lacks refinement in finding optimal paths. RL complements RRT by improving the path's quality, particularly in environments where the mobile robot needs to avoid dynamic obstacles or continuously adapt to unforeseen changes in real-time [7]. RRT's fast exploratory nature is particularly suitable for global exploration, while RL refines local decisions, making this hybrid system ideal for navigating cluttered and rapidly changing environments.

Likewise, in [11, 12], PRM was combined with RL to manage navigation in large-scale, known environments. PRM generates the global path based on the roadmap, and RL optimizes the local trajectory as conditions evolve. PRM's suitability for structured environments makes it an ideal candidate for global path planning, while RL adapts to dynamic elements, ensuring that the mobile robot can handle changing conditions without fully recalculating its route [13].

3.2. Integration with Deep Reinforcement Learning (DRL)

Deep Reinforcement Learning (DRL) further extends RL by incorporating deep learning models to process complex, high-dimensional data, enabling mobile robots to adapt in real-time. In [19], DRL was integrated with traditional methods like A* or Dijkstra to overcome the challenge of high uncertainty in environments with dynamic obstacles. A* and Dijkstra provide efficient global path planning, but DRL enables the system to process large amounts of sensor data and optimize the mobile robot's local movements. This hybrid model is particularly advantageous in high-complexity environments, where mobile robot must make frequent adjustments based on real-time feedback [15].

The decision to use DRL with A* and Dijkstra stems from DRL's ability to handle vast datasets and improve real-time decision-making, making it well-suited for dynamic settings where frequent adaptations are required.

3.3. Integration with Neural Networks

Neural Networks (NNs) enhance traditional path planning methods by learning from previous experiences and processing real-time sensor data. Integrating NNs with RRT*, as demonstrated in [16–19], accelerate the selection of sampling points and improves the overall path efficiency. The decision to combine NNs with RRT* is based on RRT*'s ability to generate near-optimal paths through re-wiring, and NNs' strength in refining the path selection by learning from prior experiences. This integration significantly reduces the computational cost of re-planning and improves the mobile robot's ability to navigate highly dynamic environments.

Similarly, integrating NNs with A* enhances the mobile robot's ability to adjust its global path based on real-time sensor data [20]. The proposed Learning Heuristic A* (LHA*) algorithm uses a neural network to model the heuristic function, ensuring faster exploration while maintaining a suboptimality bound. Also, Neural A* reformulates the A* algorithm into a differentiable module, allowing it to be integrated into a neural network and trained end-to-end, which bridges the gap between traditional search algorithms and deep learning [21]. This hybrid approach is particularly effective in environments with dynamic obstacles, where NNs help the mobile robot process sensor inputs and refine the computed path without needing constant recalculation.

Additionally, Deep Neural Networks (DNNs) further improve traditional path planning methods by learning heuristic functions to guide the search process more efficiently. As demonstrated in [22], DNNs can be trained to optimize the search cost in algorithms like A*, reducing computational overhead while maintaining accuracy. Similarly, in [23], DNN-based approach using max-pooling layers efficiently solves large-scale path planning problems without requiring training, demonstrating the power of DNNs in optimizing pathfinding in complex environments.

3.4. Integration with Fuzzy Logic

Fuzzy Logic complements traditional algorithms by providing a flexible decision-making framework in uncertain environments. When integrated with RRT, as seen in [24], Fuzzy Logic allows mobile robots to adjust their path based on imprecise or noisy sensor data, which RRT alone cannot manage effectively. The hybrid approach improves navigation in environments with high uncertainty, such as rescue missions or exploration of unknown areas. The decision to integrate Fuzzy Logic with RRT stems from the need to handle imprecise sensor data, where Fuzzy Logic's ability to manage uncertainty complements RRT's exploratory power.

When integrated with A*, as demonstrated in [25], Fuzzy Logic enhances the efficiency of the search process by dynamically adjusting the heuristic function in response to obstacle density, distance to the goal, and the number of visited nodes. While A* guarantees optimality, it is computationally expensive and memory-intensive in large or maze-like environments. The hybrid Fuzzy A* overcomes these limitations by reducing unnecessary state expansions and memory consumption, while still maintaining near-optimal paths, making it highly effective for large-scale and complex maps. In a similar manner, when combined with Dijkstra, as reported in [26], Fuzzy Logic provides the adaptability required for real-time navigation in partially known environments. Dijkstra ensures a globally optimal path offline but cannot handle unexpected obstacles during execution. Fuzzy Logic fills this gap by enabling reactive obstacle avoidance and smooth control adjustments based on sensor feedback. This integration preserves Dijkstra's reliability in static environments while extending its capability to dynamic scenarios, ensuring safe and efficient navigation where the environment cannot be fully known in advance

3.5. Summary

The reviewed studies focused on integration traditional algorithms with AI techniques for mobile robot path planning. The analysis highlighted how these hybrid approaches combine the systematic efficiency of traditional methods with the adaptability and intelligence of AI to address challenges in dynamic and complex environments. The key findings from the reviewed papers are summarized in Table 1, highlighting the strengths of each approach. Furthermore, the applications of these hybrid methods suggest their suitability for different operational contexts. For instance, A* combined with Reinforcement Learning could be well-suited for long-run navigation in complex outdoor environments and multi-robot coordination, such as autonomous transportation systems, since A* ensures a reliable global path while RL adapts to dynamic traffic or obstacle changes [6–8]. PRM integrated with RL may be advantageous for indoor ground robot navigation and outdoor UAV operations, such as warehouse automation or aerial delivery systems, because PRM efficiently explores high-dimensional spaces while RL refines safe passage in cluttered or dynamic zones [11]. Likewise, A* combined with Deep Reinforcement Learning appears suitable for highly complex and dynamic environments, such as disaster response or search-and-rescue missions, as DRL enhances adaptability to unpredictable hazards while A* provides a stable baseline path [14]. PRM combined with DRL could be applied to large-scale outdoor navigation, with potential uses in autonomous exploration in agriculture or environmental monitoring, where PRM maps vast spaces and DRL manages environmental uncertainty [22]. Meanwhile, RRT* integrated with Neural Networks shows potential for robotic arm manipulation tasks and target tracking systems, for example, in industrial assembly or surveillance applications, since RRT* generates an optimal trajectory while NN supports precision control and adaptive tracking [17]. These insights underline the flexibility of hybrid approaches in tailoring path planning solutions to diverse domains of robotic systems

4. Comparative Analysis

This section evaluates hybrid path-planning methodologies, focusing on two critical aspects: performance and computational complexity. The performance is evaluated based on the method's ability to produce optimal and smooth paths, crucial for effective and secure navigation, whereas

Table 1. Summary of hybrid approaches methods for mobile robot path planning

Paper	Global Planner Algorithm	Local Planner Algorithm	Strength
[6]	IADA*	Reinforcement Learning (RL)	- IADA* algorithm can re-plan paths efficiently in dynamic environments without recalculating the entire path when an obstacle is encountered.
[15]	—	Deep Reinforcement Learning (DRL)	- The human-in-the-loop (HL) training speeds up the convergence of the DRL algorithm, reducing the time required to learn complex navigation policies.
[7]	A*	Reinforcement Learning (RL)	- The RL is allowing the robot to adapt its policy through trial-and-error interactions with its surroundings.
[13]	PRM	Reinforcement Learning (RL)	- PRM is triggered using an updated probabilistic roadmap, if RL fails to find a valid path due to obstacles detected.
[10]	RRT	Reinforcement Learning (RL)	- RL learns to select optimal actions that lead to collision-free paths, while RRT generates collision-free states.
[8]	A*	Reinforcement Learning (RL)	- The approach can be scaled to multi-robot systems without centralized control.
[9]	RRT	Reinforcement Learning (RL)	- RL helps the RRT tree grow toward target point, avoiding computationally expensive steering functions.
[11]	PRM	Reinforcement Learning (RL)	- PRM-RL combines the strengths of PRMs for long-range planning with RL agents that handle short-range.
[12]	PRM	Reinforcement Learning (RL)	- PRM-RL is designed to be robust against sensor noise and unmodeled dynamic environments.
[14]	A*, Dijkstra	Deep Reinforcement Learning (DRL)	- The DRL is specifically trained to navigate around humans, predicting their movements and adjusting robot trajectories accordingly to avoid close encounters.
[17]	RRT*	Back Propagation (BP) Neural Networks (NNs)	- BP-RRT* method uses neural networks to predict the optimal number of samples required in each phase of the search, making it faster and more efficient. - reducing the computational process by optimizing the node selection process
[18]	A*, RRT*	Neural Networks (NNs)	- RNN continuously learns from the environment, making it adaptable and faster in generating paths. - RNN allows it to operate in a relatively constant time regardless of environmental complexity.
[19]	A*, RRT*	Neural Networks (NNs)	- Limitation learning from pre-calculated optimal paths making this method unique on real-time calculations. - The use of R-CNN allows for fast computation by leveraging offline-trained models, reducing the need for heavy real-time computations
[24]	RRT	Fuzzy Logic	- fuzzy logic is computationally light and well-suited for real-time operations. - An extended Kalman filter (EKF) is employed to minimize cross-track errors during path following, ensuring smooth and accurate navigation along the planned trajectory
[22]	PRM	Deep Reinforcement Learning (DRL)	- PMR-Dueling DQN utilizes prioritized replay and dueling networks, improving the learning process by focusing on more critical learning events and better approximating state-action values.
[20]	—	Deep Neural Network (DNN)	- DNN is used to optimize the heuristic function, allowing it to maintain the strengths of traditional search algorithms while improving efficiency.
[27]	A*	Neural Networks (NNs)	- Learning Heuristic A* (LHA*) algorithm uses a neural network to model the heuristic function. - The neural network reduces the number of unnecessary vertex expansions in a graph, speeding up the search process.
[21]	A*	Neural Networks (NNs)	- Neural A* combines learning and search into a unified framework, which allows for both task optimization and improved performance.
[23]	—	Deep Neural Network (DNN)	- OMAP does not require large datasets or neural network training, making it a simple yet powerful alternative for solving complex path-planning problems.
[28]	DWA	Fuzzy logic	- Important points on the global path are selected as key sub-target sites for the local motion planning phase.
[25]	A*	Fuzzy logic	- Significantly reduces computation and memory usage in large, complex environments while maintaining near-optimal paths.
[26]	Dijkstra	Fuzzy logic	- Ensures globally optimal offline planning with adaptive real-time obstacle avoidance in partially known environments

computational complexity analyzes convergence rate, scalability to complex environments, and re-planning efficacy in dynamic situations. Due to the varied experimental configurations and uses in different investigations, direct comparisons are difficult. Therefore, this comparison emphasizes general concepts and trade-offs, providing insights into the strengths of each method in relation to its intended application.

4.1. Performance

The performance of hybrid path planning approaches is evaluated based on two key metrics: (i) optimal and smooth path and (ii) success rate, which together reflect the quality and reliability of the paths generated by these methods.

4.1.1. Optimal and Smooth Path

Creating paths that are optimal in length and smooth in execution is essential for assuring efficient and safe navigation while reducing energy consumption and mechanical wear on robotic systems. Numerous hybrid methodologies proficiently achieve this balance by utilizing the advantages of traditional algorithms alongside AI techniques.

In [14], the methodology was evaluated across many contexts and scenarios, showing that the combination of traditional algorithms with Deep Reinforcement Learning (DRL) always produces optimal paths compared to independent methods. In [15] RRT was integrated with RL, utilizing RRT as the sampling method and RL to link the sampled points in order to formulate the shortest and most secure path. This hybrid methodology utilizes the exploration efficacy of RRT and the flexibility of RL to produce both optimal and smooth paths.

In [16], RRT* was combined as a global planner alongside a neural network as a local planner, significantly improving path-finding efficiency. Experiments comparing this approach with RRT* and improved RRT* (IRRT*) showed that while path length remained comparable, integrating neural networks reduced the number of nodes and computation time required to reach the optimal path. Similar results were observed in [15, 16], where the integration of neural networks refined the sampling and path selection processes, producing optimal solutions more efficiently than RRT* and Improved RRT* as demonstrated in Figure 2.

4.1.2. Success Rate

The success rate is an essential indicator of the reliability of path planning systems, representing the proportion of paths successfully completed without failures or collisions across many scenarios. Hybrid methodologies that combine traditional algorithms with AI techniques have shown considerable enhancements in success rates, especially in dynamic and intricate contexts.

In [11], 50 tests were performed to assess the integration of Anytime Dynamic A* (iADA*) with various reinforcement learning techniques, comparing it with standalone iADA* and iADA* integrated with DQN and DDPG. The results showed that both iADA* with DQN and iADA* with DDPG achieved high success rates in both dynamic and static environments: 94% and 90% in dynamic environments, compared to 74% for standalone iADA*, and 100% in static environments, compared to 94% for standalone iADA* as shown in Table 2.

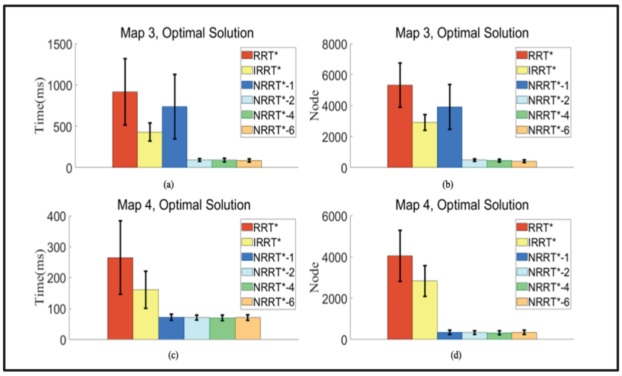

Figure 2. Comparison of the optimal solutions between the integration of RRT with Neural Network, RRT, and Improved RRT* [16]

Table 2. Comparison of the success rates between the integration of iADA with AI techniques and standalone iADA [6]

Static environment	Algorithms		
	IADA*	IADA*+DQN	IADA*+DDPG
Success	47	50	50
Failure	3	-	-
Success rate	94%	100%	100%
Dynamic environment	Algorithms		
	IADA*	IADA*+DQN	IADA*+DDPG
Success	37	47	45
Failure	13	3	5
Success rate	74%	94%	90%

Likewise, in [14], success rates were emphasized when integrating traditional methods with Deep Reinforcement Learning (DRL). Experiments in various obstacle-laden environments demonstrated that the hybrid technique consistently navigated paths without failure, surpassing traditional algorithms. In [12], Reinforcement Learning was utilized as a global Guide, integrating global planning with localized RL modifications, and attained elevated success rates in complex environments. [22] also exhibited strong performance by achieving higher success rates in both static and dynamic scenarios.

4.2. Computational Complexity

The Computational Complexity of hybrid path planning methods is a crucial factor in assessing their efficiency and practicality, particularly for real-time and resource-constrained applications.

4.2.1. Convergence Speed

One of the key aspects of computational complexity is Convergence Speed, which measures how quickly a method can compute a solution or learn an effective policy.

Table 3. Comparison of convergence time between PRM+DQN, DQN, DDQN, and Q-learning [22]

Method	Algorithm comparison on environment E-2	
	Success rate	Convergence Time/min
Q-learning	26.7	268
DQN	49.3	139
DDQN	56.4	116
PMR-Dueling DQN	84.6	107
Method	**Algorithm comparison on environment E-3**	
	Success rate	Convergence Time/min
Q-learning	21.4	276
DQN	32.6	161
DDQN	41.5	143
PMR-Dueling DQN	79.6	122

Methods that combine reinforcement learning (RL) or deep reinforcement learning (DRL) with traditional algorithms significantly improve convergence speed by enhancing decision-making and minimizing unnecessary computations. [9] employs Globally Guided Reinforcement Learning, which improves convergence by utilizing global planning to guide the RL agent's attention towards pertinent regions of the environment. This reduces the search space and enables the approach to compute optimal pathways more quickly, even in dynamic and complex environments. Similarly, [22] integrates Dueling Deep Q-Networks (Dueling DQN) with PRM, achieving superior performance compared to standalone Q-learning, DQN, and DDQN algorithms. In experiments, the integrated method reduced computation times by significant margins, outperforming the other approaches by 9 and 21 minutes across different test environments, demonstrating its efficiency and scalability as shown in Table 3.

The integration of neural networks (NNs) with traditional algorithms also significantly improves convergence speed by reducing computational overhead. [17] integrates RRT* with a neural network, enhancing the efficiency of sampling nodes and focusing calculations on important regions. Experimental results indicate that this integration achieved a calculation time of 12.6 seconds, in contrast to 21.04 seconds for RRT, 32.31 seconds for RRT*, and 16.07 seconds for improved RRT* (IRRT*). This demonstrates the integration's ability to not only reduce computational time but also minimize the number of nodes required for pathfinding as shown in Table 4.

4.2.2. Scalability

The scalability of hybrid path planning approaches indicates their ability to maintain efficiency and performance as environmental complexity increases, such as larger workspaces, higher obstacle density, or more dynamic environments. Scalability is essential for practical applications requiring dependable navigation in various and challenging environments.

Table 4. Comparison of convergence speed between BP-RRT, RRT*, RRT, and IRRT* [17]

Algorithm name	Avarage search time/s	Avarage number of nodes samples
RRT	21.04	5293.6
RRT*	23.31	1366.7
Improved P-RRT*	16.70	1482.5
BP-RRT*	12.76	1044.7

In [6], the system was tested and evaluated based on success and failure rates in both static and dynamic environments. The test scenario included 17 obstacles of varying sizes, with 7 static and 10 dynamic obstacles moving randomly at velocities ranging from 5 and 25 m/s in unrestricted directions. These conditions simulate complex, dynamic environments, and the method consistently achieved success rates of 90% and above, underscoring its ability to scale effectively in highly dynamic scenarios.

Similarly, [10] showcased the scalability of their method through experiments conducted in six different environments with varying random obstacle distributions. In Environment 5, there were two paths to the destination, whereas in Environment 6, the UAV could reach the goal only by navigating along a specific side of the maze. These experiments demonstrated the method's capability to adapt to both open and constrained environments, maintaining robust performance across all scenarios.

[22] emphasized scalability by evaluating the method in three distinct environments: a regular map, a random map, and a free map. The regular map mimics warehouse environments with a mix of static and dynamic obstacles, while the random map includes randomly distributed static and dynamic obstacles at varying densities. The free map focuses exclusively on dynamic obstacles. Dynamic obstacles were modeled as uncontrollable robots that could move one cell per step in any direction. These experiments demonstrated the method's ability to efficiently navigate environments of varying complexity and obstacle configurations.

4.2.3. Re-planning Efficiency

Re-planning Efficiency evaluates the ability of path-planning techniques to adjust existing paths or create new ones in response to dynamic environmental changes, including the emergence of new obstacles or changes in the goal position. This metric is essential for real-time applications in unstable environments.

[6] and [8] showcase strong re-planning efficiency by integrating Anytime Dynamic A* (iADA*) and reinforcement learning. The iADA* algorithm in [6] can re-plan paths efficiently in dynamic environments without recalculating the entire path when an obstacle is encountered. Instead, it updates only the affected portion of the path, significantly reducing computational overhead. Similarly, in [8] re-planning efficiency had

been enhanced by using a globally guided reinforcement learning approach that employs a Moving Cost metric.

$$Moving\ Cost = \frac{N_s}{\left|C_{goal} - C_{start}\right|\ Ls} \qquad (1)$$

Where N_s is the number of steps taken, and $\left|C_{goal} - C_{start}\right|\ Ls$ is the Manhattan distance between the start and goal cells. This metric indicates the ratio of actual moving steps to the ideal number of steps. The naive reward-based approach achieved success rates ranging between 68% and 89%, ensuring faster re-planning with shorter paths even in complex and dynamic environments.

In [9, 10, 12], sampling-based methods (RRT, RRT*, and PRM, respectively) were utilized to establish key nodes, while AI techniques connect these nodes to generate optimal paths. In dynamic environments, these established nodes provide a robust framework for quick re-planning. When obstacles appear or configurations change, the AI component adjusts the connections between nodes, allowing for efficient path recalculations without starting from scratch.

4.3. Summary

The assessment of hybrid methods illuminated both the strengths and the limitations of each approach, particularly when examined across key dimensions such as re-planning efficiency, scalability, convergence speed, success rate, and path optimality. Reinforcement Learning (RL) integrations demonstrated superior performance in terms of path smoothness, success rate, and adaptability to dynamic environments, making them highly effective for tasks requiring continuous re-planning. However, this advantage comes at the cost of extensive training times and significant data requirements, which limit their practicality in scenarios where fast deployment is necessary [6, 17].

Neural Networks (NN) offered strong performance in processing high-dimensional sensor data and enhancing obstacle detection. Yet, in dynamic and uncertain settings, their reliance on pre-trained models constrained adaptability, often resulting in reduced path smoothness and lower success rates compared to RL-based hybrids. This contrast suggests that NN integrations are better suited for relatively structured environments with predictable features, while RL hybrids remain stronger in highly variable contexts [17].

Deep Reinforcement Learning (DRL) extended the adaptability of RL by handling more complex decision spaces, but its computational burden and slower convergence made it less viable for real-time applications. While DRL may excel in offline or simulation-rich domains where training time is less restrictive, it underperforms in fast-changing environments that require immediate responsiveness. Fuzzy Logic, in contrast, provided resilience in uncertain scenarios due to its rule-based flexibility, but it struggled with scalability and path optimization, limiting its effectiveness in large-scale or multi-robot systems [7].

Overall, the comparative analysis indicates that no single hybrid method provides a universally optimal solution. Each combination exhibits strengths in certain conditions while revealing clear weaknesses in others. This underscores the need for future work to focus not only on improving adaptability and computational efficiency, but also on systematically benchmarking hybrids across diverse scenarios to identify where each approach succeeds or fails.

5. Research Challenges and Future Directions

Although hybrid path-planning techniques have advanced substantially in recent years, many barriers remain that limit their usability and performance in applied settings, particularly in dynamic and uncertain environments. One of the main limits relates to collision avoidance in dynamic scenarios. In [6, 9], due to the local data from LiDAR or sensors, the system does not consider the prediction of future states of moving obstacles [15]. Similarly, it has limitations arising from the Q-learning equation, thus running into potential problems with convergence within dynamic environments, limiting the potential for any true obstacle avoidance strategies. In addition, the need for extensive datasets and training via reinforcement learning (RL) requires considerable computational resources and time to effectively train to act in complex environments [7]. Additionally, Noise in sensor data poses additional challenges, as seen in [12], where environmental uncertainty from noisy inputs impacts the accuracy and reliability of the path planning process. Lastly, combining neural networks with traditional techniques simultaneously in varying applications specifically in high-dimensional environments and concomitant dynamic obstacles [16, 18], can often only be done with considerable computational requirements, limiting real-time adaptability.

Future research prioritizes addressing challenges related to obstacle avoidance in dynamic environments and sensor noise. Collision avoidance could also focus on the application of predictive models to calculate projected movements and integrate global and local data for overall situational awareness [6]. In addition, future research may investigate the combination of traditional algorithms, such as D* and D*Lite, with AI models in controlled scenarios to assess their performance in dynamic environments. Another promising direction is hybrid path planning in unknown environments, where prior maps are unavailable or incomplete. In such cases, traditional algorithms can provide exploratory global strategies, while AI techniques—particularly RL and DRL—can adaptively refine navigation based on real-time sensor feedback. This combination could be valuable for applications such as autonomous exploration, planetary rovers, or search-and-rescue missions in disaster zones, where the environment is partially or entirely unknown.

6. Conclusion

This review paper examined various studies published that utilized hybrid models for path planning. Hybrid models leverage the systematic reliability of traditional algorithms while leveraging AI's adaptability to address complex navigation problems in both static and dynamic environments. We concluded with various advantages of each approach including the increase of path optimality, smoothness, scalability, and re-planning. The comparative analysis showed that RL outperformed other methods in path smoothness and successful attempts, especially when complemented by graph-based approaches like A*, which incorporated flexibility to achieve optimal paths. Neural Networks and Deep Reinforcement Learning offered significant adaptations and decision-making capabilities but incurred computational cost. Fuzzy logic was robust to uncertainty but was not successful in path optimization and scalability, which are the main advantages of other associated methods.

Overall, despite additional advancements in hybrid methodology, future work will focus on addressing specific challenges on collision avoidance in highly dynamic environments, reducing issues related to sensor noise, and minimizing computational cost for real-time navigation in autonomous applications. Future research should continue to develop hybrid models that combine automated path-planning with collision avoidance, robust optimization to manage sensor noise, and lightweight analytics to limit computational cost. Future exploration of combinations of AI and traditional methods, using RL with adaptive graph-based approaches, would enhance the adaptability and reliability of hybrid path-planning models. This paper offers an in-depth analysis of hybrid path-planning techniques, providing essential insights for researchers seeking to improve autonomous navigation systems for practical applications.

AUTHORS

Mohamed Abdelghafar – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: mohamedalyabdelghafar@gmail.com.

Hazlina Selamat* – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: hazlina@utm.my.

Nurulaqilla Binti Khamis – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: nurulaqilla@utm.my.

Anas Aburaya – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: ihanas2002@graduate.utm.my.

Mohd Taufiq Muslim – Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Skudai, 81300, Johor, Malaysia, e-mail: mohdtaufiqmuslim@utm.my.

*Corresponding author

ACKNOWLEDGEMENTS

This research received funding under Research Grant Vote 00Q18 by Universiti Teknologi Malaysia and the Malaysian Ministry of Higher Education.

References

[1] S. Aggarwal and N. Kumar, "Path Planning Techniques for Unmanned Aerial Vehicles: A Review, Solutions, and Challenges" Computer Communications, vol. 149, pp. 270–299, Jan. 2020, doi: 10.1016/j.comcom.2019.10.014.10.1016/j.comcom.2019.10.014.

[2] J. R. Sánchez-Ibáñez, C. J. Pérez-Del-pulgar, and A. García-Cerezo, "Path Planning for Autonomous Mobile Robots: A Review," Sensors, vol. 21, no. 23, 2021, doi: 10.3390/s21237898.

[3] C. Cheng, Q. Sha, B. He, and G. Li, "Path Planning and Obstacle Avoidance for AUV: A Review," Ocean Engineering, vol. 235, 2021, doi: 10.1016/j.oceaneng.2021.109355.

[4] K. Katona, H. A. Neamah, and P. Korondi, "Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot," Sensors, vol. 24, no. 11, 2024, doi: 10.3390/s24113573

[5] B. B. K. Ayawli, R. Chellali, A. Y. Appiah, and F. Kyeremeh, "An Overview of Nature-Inspired, Conventional, and Hybrid Methods of Autonomous Vehicle Path Planning," Journal of Robotics, 2018, doi: 10.1155/2018/8269698.

[6] A. A. Maw, M. Tyan, T. A. Nguyen, and J. W. Lee, "IADA-RL: Anytime Graph-Based Path Planning With Deep Reinforcement Learning for an Autonomous UAV*," *Applied Sciences*, vol. 11, no. 9, 2021, doi: 10.3390/app11093948.

[7] K. Cai, C. Wang, J. Cheng, S. Song, C. W. de Silva, and M. Q-H Meng, "Mobile Robot Path Planning in Dynamic Environments: A Survey," "arXiv preprint arXiv:2002.09197, 2020".

[8] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, Oct. 2020, doi: 10.1109/LRA.2020.3026638

[9] H. T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "RL-RRT: Kinodynamic Motion Planning via Learning Reachability Estimators from RL Policies," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4298–4305, Oct. 2019, doi: 10.1109/LRA.2019.2931199.

[10] P. Gao, Z. Liu, Z. Wu, and D. Wang, *A Global Path Planning Algorithm for Robots Using Reinforcement Learning*. IEEE, 2019.

[11] A. F. K. Oslund, O. R. A. Francis, L. Tapia, and M. F. J. Davidson, *PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning*. [IEEE], 2018.

[12] A. Francis et al., "Long-Range Indoor Navigation with PRM-RL," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1115–1134, Aug. 2020, doi: 10.1109/TRO.2020.2975428.

[13] V. Bisht, "Visibility-Constrained Path Planning for Unmanned Aerial Vehicles," M.S./Ph.D. thesis, Year 2023.

[14] R. Guldenring, M. Gorner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning Local Planners for Human-Aware Navigation in Indoor Environments," in *IEEE International Conference on Intelligent Robots and Systems*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 6053–6060. doi: 10.1109/IROS45743.2020.9341783.

[15] S. Zhang, Y. Li, F. Ye, X. Geng, Z. Zhou, and T. Shi, "A Hybrid Human-in-the-Loop Deep Reinforcement Learning Method for UAV Motion Planning for Long Trajectories with Unpredictable Obstacles," *Drones*, vol. 7, no. 5, May 2023, doi: 10.3390/drones7050311.

[16] J. Wang, W. Chi, C. Li, C. Wang, and M. Q. H. Meng, "Neural RRT∗: Learning-Based Optimal Path Planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, Oct. 2020, doi: 10.1109/TASE.2020.2976560.

[17] Q. Gao, Q. Yuan, Y. Sun, and L. Xu, "Path Planning Algorithm of Robot Arm Based on Improved RRT* and BP Neural Network Algorithm," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, Sep. 2023, doi: 10.1016/j.jksuci.2023.101650.

[18] S. Chehelgami, E. Ashtari, M. A. Basiri, M. Tale Masouleh, and A. Kalhor, "Safe Deep Learning-Based Global Path Planning Using a Fast Collision-Free Path Generator," *Rob Auton Syst*, vol. 163, May 2023, doi: 10.1016/j.robot.2023.104384.

[19] Y. Liu, Z. Zheng, F. Qin, X. Zhang, and H. Yao, "A Residual Convolutional Neural Network Based Approach for Real-Time Path Planning," *Knowledge Based Systems*, vol. 242, Apr. 2022, doi: 10.1016/j.knosys.2022.108400.

[20] T. Takahashi, H. Sun, D. Tian, and Y. Wang, "Learning Heuristic Functions for Mobile Robot Path Planning Using Deep Neural Networks," in Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

[21] R. Yonetani, T. Taniai, M. Barekatain, M. Nishimura, and A. Kanezaki, "Path Planning using Neural A* Search," 2021. [Online]. Available: https://omron-sinicx.github.io/

[22] D. A. Deguale, L. Yu, M. L. Sinishaw, and K. Li, "Enhancing Stability and Performance in Mobile Robot Path Planning with PMR-Dueling DQN Algorithm," *Sensors*, vol. 24, no. 5, Mar. 2024, doi: 10.3390/s24051523.

[23] T. Kulvicius, S. Herzog, M. Tamosiunaite, and F. Wörgötter, "Generation of Paths in a Maze using a Deep Network without Learning," 2020. [Online]. Available: https://arxiv.org

[24] Z. Liu, Y. Zhang, C. Yuan, L. Ciarletta, and D. Theilliol, "Collision Avoidance and Path Following Control of Unmanned Aerial Vehicle in Hazardous Environment," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 95, no. 1, pp. 193–210, Jul. 2019, doi: 10.1007/s10846-018-0929-y.

[25] G. Airlangga, "Fuzzy A* for optimum Path Planning in a Large Maze," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 4, pp. 455–466, Dec. 2023, doi: 10.12928/biste.v5i4.9394.

[26] S. Sahloul, D. Ben Halima Abid, and C. Rekik, "An Hybridization of Global-Local Methods for Autonomous Mobile Robot Navigation in Partially-Known Environments," Jul. 01, 2021, *Department of Agribusiness, Universitas Muhammadiyah Yogyakarta*. doi: 10.18196/jrc.2483.

[27] S. Kim and B. An, *Learning Heuristic A*: Efficient Graph Search using Neural Network*, 2020. doi: 10.0/Linux-x86_64.

[28] Y. Sun et al., "Local Path Planning for Mobile Robots Based on Fuzzy Dynamic Window Algorithm," *Sensors*, vol. 23, no. 19, Oct. 2023, doi: 10.3390/s23198260.