Research On Trajectory Planning Methods of Mobile Robots Based on SLAM Technology

Xindan Zhang, Haoyuan Zheng *
School of Automation, Fuzhou University, Fuzhou, China
* Corresponding Author Email: HAOYUAN.ZHENG.2022@MUMAIL.IE

Abstract. With the widespread adoption of mobile robots across various domains, the need for precise and efficient path planning has become paramount. Mobile robots, utilizing Simultaneous Localization and Mapping (SLAM) technology, achieve autonomous positioning and map construction, thereby laying a critical foundation for path planning. Consequently, this paper primarily focuses on path planning algorithms, encompassing SLAM, Dijkstra's algorithm, A* algorithm, and the Artificial Potential Field (APF) algorithm.The paper aims to provide a concise introduction to the fundamental principles of these algorithms, accompanied by a review of their practical applications across diverse fields. Additionally, it will discuss future developmental trends. Through an in-depth analysis of these algorithms, the paper seeks to elucidate their roles and advantages in mobile robot trajectory planning. Furthermore, it will objectively assess their limitations and disadvantages, thereby enabling readers to gain a comprehensive understanding of their real-world efficacy and potential constraints.Ultimately, the paper will undertake a comprehensive comparison and discussion of each algorithm, endeavoring to furnish readers with a thorough comprehension of mobile robot trajectory planning methods leveraging SLAM technology. It aspires to offer valuable insights and guidance for future research and practical endeavors in this domain.

Keywords: SLAM; Dijkstra; A* algorithm; APF; Path planning.

1. Introduction

In recent years, mobile robots have found widespread applications across various domains such as disaster relief, transportation, and medical care owing to their capability to autonomously navigate and execute intelligent tasks in diverse environments [1]. This trend holds significant importance not only in the present but also for the future. Over the past two decades, researchers have dedicated efforts to addressing path planning challenges and have proposed a multitude of methodologies and applications. Notably, in 2001, J. Blitch, a leading figure in disaster robotics at the University of South Florida, spearheaded CRASAR (Center for Robot-Assisted Search and Rescue) to partake in the search and rescue operation at the World Trade Center [2]. Subsequently, in 2006, Rentschler et al. introduced a wheeled micro-mobile robot tailored for laparoscopic support, engineered to furnish wireless multi-angle views of the intra-abdominal surgical field [3]. Additionally, in 2015, the American iRobot company unveiled the Roomba980 cleaning robot, equipped with precise self-positioning and navigation capabilities, effectively streamlining cleaning operations [4].

Nevertheless, a significant challenge persists: the autonomous localization of robots in unfamiliar environments poses difficulty, potentially resulting in prolonged and uncertain paths. Hence, a pivotal technology—Simultaneous Localization and Mapping (SLAM)—becomes indispensable. Robots can be outfitted with various sensors to implement SLAM algorithms and construct maps, facilitating autonomous positioning. Leveraging SLAM technology in conjunction with path planning methodologies enables the selection of optimal paths, empowering robots to execute diverse tasks efficiently and safely. Consequently, this article introduces and scrutinizes a range of path planning algorithms based on SLAM technology, intertwining SLAM technology with path planning principles, elucidating the merits and drawbacks of distinct methodologies, and offering guidance on selecting the most suitable approach for task completion. We aspire to furnish more dependable and efficient references and support for future research endeavors pertaining to the intelligent operation of mobile robots across varied environments.

The subsequent sections of this article are structured as follows: Section 2 delves into SLAM technology; Section 3 delineates the fundamental tenets of Dijkstra's algorithm and provides an overview of its application in path planning; Section 4 expounds upon the basic principles of the A* algorithm and its applicability in path planning; Section 5 elaborates on the fundamental principles of the artificial potential field algorithm and its utilization in path planning; Section 6 conducts a comparative analysis of the advantages and disadvantages of the three methodologies, offering recommendations; and finally, Section 7 furnishes a summary.

2. SLAM Section Headings

In mobile robot systems, autonomous navigation is regarded as a fundamental challenge. Achieving autonomous navigation entails the robot's ability to traverse within a mapped area, discern direction, and devise routes between points—essentially requiring precise positioning and mapping capabilities. To address these requirements, Simultaneous Localization and Mapping (SLAM) technology emerges as the predominant solution. Over recent years, SLAM technology has witnessed widespread adoption across various domains, encompassing autonomous vehicles, robot navigation, drone navigation, augmented reality (AR), and virtual reality (VR), among others. This section aims to introduce the foundational principles of SLAM technology and elucidate the associated challenges.

2.1. Basic Principle

SLAM, or Simultaneous Localization and Mapping, is a technology utilized to concurrently deduce the movement trajectory of sensors and reconstruct the spatial layout of surveyed areas [5]. By means of data association and landmark association, SLAM solutions can execute loop closure, mitigating pose uncertainty within the map and thereby enhancing the solution [5].

SLAM constitutes an estimation problem that concurrently estimates variables pertaining to the robot's trajectory and variables representing landmark locations within the environment. Detailed formulations are provided in [5]. SLAM methodologies predominantly employ probabilistic techniques, such as particle filters and Kalman filters (Extended Kalman Filter, EKF) [6]. Initially, solving the SLAM mapping problem involved extending the Kalman filter [5]. Figure 1 illustrates the EKF-SLAM process. At time t, when external data is received, equation (1) is utilized to forecast the robot's state, and the identified features are compared with those in the map [5]. The results of these matches can then be utilized to update the state and map utilizing equation (2). In cases where no match is found, attempts are made to initialize and incorporate the feature into the map. This process operates recursively throughout, continuously iterating. Further details are available in equation (3) [5].

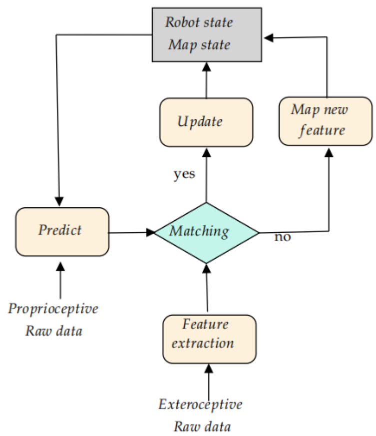

Figure 1. Process flow diagram for EKF-SLAM [5]

$$
\begin{aligned}
&P(x_{k|k-1z}, M_{k|k-1} \mid Z_{0:k-1}, U_{0:k}) = \dots \\
&\int P(x_{k|k-1} \mid x_{k-1|k-1}, u_k) \times \dots \\
&P(x_{k-1|k-1}, M_{k-1|k-1} \mid Z_{0:k-1}, U_{0:k-1}) dx_{k-1|k-1}
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
&P(x_{k|k}, M_{k|k} \mid Z_{0:k}, U_{0:k}) \propto \dots \\
&P(z_{i,k} \mid x_{k|k-1}, M_{k|k-1}) \times \dots \\
&P(x_{k|k-1}, M_{k|k-1} \mid Z_{0:k-1}, U_{0:k})
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
&P(x_{k|k}, M_{k|k} \mid Z_{0:k}, U_{0:k}) \propto \dots \\
&P(z_{k,i} \mid x_{k|k-1}, M_{k|k-1}) \times \dots \\
&\int P(x_{k|k-1} \mid x_{k-1|k-1}, u_k) \times \dots \\
&P(x_{k-1|k-1}, M_{k-1|k-1} \mid Z_{0:k-1}, U_{0:k-1}) dx_{k-1|k-1}
\end{aligned} \tag{3}
$$

2.2. Development and Challenge

Although the 2D SLAM problem is considered tractable, there remains a pressing need for enhancement in its accuracy, speed performance, and ability to handle large datasets [6]. The pursuit of high-quality and robust 3D visual SLAM remains a formidable challenge, particularly in outdoor, unstable, and dynamic environments encountered by drones, autonomous vehicles, and other applications [6]. Localization and map construction stand as pivotal components of SLAM, intricately interdependent, hence termed the "chicken and egg" problem [6]. Improving the odometry model holds promise for enhancing positioning accuracy and map construction, with the latter playing a pivotal role in path planning, obstacle avoidance, and other functionalities [6]. In summary, the advancement of SLAM technology necessitates addressing challenges related to perceptual instability, accuracy, and speed [6].

3. Dijkstra's Algorithm

Dijkstra's algorithm plays a pivotal role in mobile robot trajectory planning, offering a well-defined and straightforward approach for navigation in intricate environments. Its practicality has led to widespread adoption across diverse fields, prominently in traffic information systems such as Google Maps [7]. The crux lies in resolving the conundrum of finding the optimal path from a designated starting point to a desired endpoint. Particularly when integrated with SLAM technology, it furnishes a comprehensive autonomous navigation solution for robots, enabling precise self-positioning and optimal path planning in unfamiliar or dynamic environments. This section will proceed to elucidate the fundamental principles of Dijkstra's algorithm, its application in mobile robot trajectory planning, a concise analysis of its merits and demerits, and a succinct introduction to recent advanced algorithms predicated on it.

3.1. Basic Principle

Dijkstra's method aims to identify the shortest route connecting two nodes within the graph. which is achieved by considering the distance and connection relationship between each node. First use graph theory to model, a directed graph G = <V, E, W>, where V represents the collection of vertices (including the source vertex s), E denotes the set of edges, and W signifies the weights assigned to the edges [7]. and then start from the original vertex to find a set of points with the smallest sum of weights that can reach the target position. The algorithm process is as follows:

Algorithm: Dijkstra (G, d [], s)

// Typically, the graph G is defined as a global variable. The array d represents the shortest path length from the source point to each respective point, with s denoting the starting point.

```
Initialization: Set the shortest path length of the source point s to 0, that is, d[s] = 0; initialize the shortest path length of all other points to infinity, indicating unknown. That is, d[v] = INF for all vertices v; mark all vertices as unvisited.
for (loop n times)
    u = the label of the vertex that minimizes d[u] but has not yet been visited.
    Remember that u has been visited.
    for (all vertices v that can be reached from u)
        if (v has not been visited && using u as the intermediate point makes the shortest path d[v] from s to vertex v more optimal)
            Optimize d[v];
```

3.2. Dijkstra Algorithm in Path Planning

In the realm of robot path planning, Dijkstra's algorithm stands as a cornerstone due to its status as a classic shortest path algorithm, paving the way for subsequent enhancements and advancements in path planning methodologies [1]. Widely employed in global navigation-type path planning, Dijkstra's algorithm serves as a graph search method utilized in offline path planning modes to derive the shortest path by identifying the optimal route with non-negative edge path costs [1].

The Ant Colony Optimization (ACO) algorithm finds application in the path planning of mobile robots navigating through complex environments [1]. In a study by [8], a fusion approach combining Dijkstra's algorithm with the Ant Colony System (ACS) was proposed to ascertain a suboptimal collision-free path, effectively resolving mobile robot path planning challenges.

Furthermore, an improved variant of Dijkstra's algorithm was introduced in [7], merging global Dijkstra algorithmic principles with local iteration techniques. Leveraging ultrasonic sensor data for trajectory updates, this method achieves global optimal path planning while averting collisions, successfully applied in DIY mobile robot path planning and experimentation. This exemplifies the efficacy and practicality of Dijkstra's algorithm in addressing path planning intricacies within complex environments [7].

Furthermore, a relaxed version of Dijkstra's algorithm, designed to achieve real-time path planning and improve the speed of mobile robots in large, obstacle-rich environments, was introduced in [9], utilizing grid environment modeling. This approach constructs a Manhattan distance potential field and employs eight-neighborhood search, delivering rapid computations and reduced path length errors compared to traditional Dijkstra's algorithm. However, it currently remains in the software simulation phase, necessitating further validation of its effectiveness [9].

In a study conducted by [10], a path planning method rooted in Dijkstra's algorithm is introduced to address the issue of path drift arising from positioning errors in GPS-free environments. By estimating error characteristics via statistical mileage analysis, the method recalibrates paths to minimize cumulative errors. Leveraging a global error map, this approach swiftly identifies the path with the least error, thereby enhancing the accuracy and safety of path planning [10].

In summation, Dijkstra's algorithm assumes a pivotal role in robot path planning endeavors. Research demonstrates its efficacy when seamlessly integrated with various algorithms to effectively tackle challenges inherent in complex mobile robot path planning environments, thereby bolstering accuracy and safety. Moreover, the demonstrated efficiency and practicality of Dijkstra's algorithm in navigating robot path planning dilemmas offer valuable insights for the future evolution of the field.

4. Astar algorithm

As an extension and improvement of Dijkstra's algorithm, the Astar algorithm also plays an important role in the field of path planning. A* functioning as a heuristic algorithm, employs a heuristic function to assess the cost from any point within the graph to the target position, aiding in determining the optimal search direction. This section will next discuss the basic principles of the Astar algorithm and briefly introduce its specific applications.

4.1. Basic Principle

We first need to understand one of the core concepts of the Astar algorithm-the comprehensive cost value:

$$\mathrm{f}(n) = h(n) + g(n) \tag{4}$$

Where n represents the current node, h(n) represents the estimate of the minimum cost from the current node to the target node, and g(n) represents the actual measurement value from the starting node to the current node. f(n) is regarded as a heuristic comprehensive cost value, which is used to guide the A* algorithm to select the optimal path during the search process. There are usually three ways to determine the heuristic function: Manhattan distance (dm), Euclidean distance (de) and diagonal distance (dd), which are expressed as follows [11]:

$$\mathrm{d}_m = |x_i - x_j| + |y_i - y_j| \tag{5}$$

$$\mathrm{d}_\mathrm{d} = \max(|x_i - x_j|, |y_i - y_j|) \tag{6}$$

$$\mathrm{d}_e = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{7}$$

A smaller heuristic function can improve the search accuracy but will increase the search time; while a larger heuristic function can speed up the search but may not find the optimal solution. As a heuristic function, Manhattan distance is close to the actual minimum cost and can effectively guide the search, so it is often used in path planning of the A* algorithm.

4.2. A-star Algorithm in Trajectory Planning for Mobile Robots

4.2.1 Application of A-star algorithm in SLAM

In the domain of SLAM (Simultaneous Localization and Mapping), the A* algorithm assumes a pivotal role, particularly in path planning, environment modeling, and obstacle avoidance during the robot localization process, thereby offering significant application value.

Within SLAM, the A* algorithm leverages the known global map for path planning, facilitating mobile robots in efficiently determining safe and cost-optimal paths from the starting point to the target destination [12]. Acknowledged as an effective static map path planning algorithm, A* is widely esteemed as a proficient method for solving shortest path dilemmas in static maps [12]. In [2], the A* algorithm is utilized to plan paths for mobile robots, aiming to ascertain the shortest path utilizing the SLAM approach, augmented with ultrasonic sensors and rotary encoders. Additionally, in [13], the A* algorithm is fused with SLAM technology to address the global dynamic path planning challenges encountered by unmanned vehicles. Through the integration of the DWA algorithm and the construction of an enhanced evaluation function, both global optimality and local optimality in path planning are achieved, thereby enhancing the efficiency and accuracy of autonomous navigation for unmanned vehicles within indoor environments.

However, notwithstanding its proficiency in optimizing shortest paths, the A* algorithm encounters challenges when confronted with a substantial number of nodes, leading to prolonged search times and issues such as excessive inflection points, uneven paths, and elongated distances [11, 14]. To enhance path planning efficiency in complex environments, researchers are dedicated to refining methodologies based on the A* algorithm.

4.2.2 Improved A* algorithm applications

In a study conducted by [11], the BB-A algorithm (Bidirectional Search-Binary tree A* algorithm) was introduced as an enhancement to the A* algorithm. By modeling the road network environment, incorporating bidirectional search strategy, and implementing binary tree optimization, the BB-A algorithm effectively tackles constrained path planning problems, thereby improving path planning efficiency and achieving global path planning under constraints. The BB-A algorithm demonstrates superior computational speed and adaptability compared to the traditional A* algorithm. While the obtained path may not always be optimal, it offers valuable insights for the global path planning of autonomous driving systems [11].

Furthermore, an enhancement to the A* algorithm was proposed by increasing the number of search directions to 12 [14]. This modification, compared to the traditional A* algorithm, results in shorter paths, smoother trajectories, and reduced inflection points. Successfully applied to path planning for simulated moving vehicles, this improvement addresses the issue of excessive inflection points encountered by the traditional A* algorithm, ensuring compliance with actual physical constraints [14].

Additionally, in [12], an enhanced version of the A* algorithm, ASL-DWA (A* Leading Dynamic Window Approach), was introduced to tackle the path planning problem of indoor cleaning robots. Through the integration of a new hybrid heuristic function and the inclusion of the global path yaw angle, this method significantly decreases the search node count, eliminates polyline routes, navigates around local unidentified obstacles, and reduces the likelihood of getting trapped in local optima.

In summary, the A* algorithm amalgamates the shortest path search capability of Dijkstra's algorithm with the heuristic search approach of the greedy algorithm, resulting in high search efficiency and low time complexity. Widely utilized in robot path planning applications, the A* algorithm's flexibility and efficiency enable rapid determination of the best route from the robot's present location in relation to the desired destination, while accommodating various movement restrictions such as obstacle avoidance and terrain considerations.

5. APF Algorithm

The artificial potential field (APF) method stands as a widely utilized path planning algorithm renowned for its exceptional planning speed and commendable real-time performance [15]. By emulating an electric potential field and computing the potential energy at each position, this method guides the robot to navigate the environment efficiently. In this framework, the desired point exerts attraction, whereas impediments cause repulsiveness, enabling the robot to steer clear of obstacles and progress towards the target [15]. Nonetheless, despite its merits, APF is susceptible to drawbacks, notably the potential entrapment in local minima within complex environments. This section will commence by providing a succinct overview of the fundamental principles, advantages, and disadvantages of the APF method. Subsequently, it will delve into recent advancements and applications aimed at addressing its limitations and enhancing its efficacy in path planning scenarios.

5.1. Basic principle

APF is mainly divided into two parts: gravitational field and repulsive field. Every obstacle creates an opposing field to the gravitational field that the target point produces. The robot receives both gravitational and repulsive forces in the potential field and moves toward the target point. In the APF field, the robot always follows the movement from high potential to low potential. Its potential energy function can be expressed as follows [1]:

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{8}$$

Uatt(q) is the attractive potential function and Ureq(q) is the repulsive potential function.

5.1.1 Attractive potential function & attractive force

The gravitational field makes the robot move toward the target point. The strength of the gravitational field is usually inversely proportional to the separation between the robot and the intended location. That is, the closer it is to the target point, the greater the attractive force. In the APF field, the gravitational field can be represented by a mathematical function. When the robot approaches the target point, the intensity of the gravitational field gradually increases, pushing the robot closer to the target point. In this article, the attraction potential function can be expressed as follows [1]:

$$U_{\mathrm{att}}(q)=\frac{1}{2}\zeta\rho^{2}(q.q_{goal}) \tag{9}$$

where ζ is the gain and (q.qgoal) is the distance between the robot and the target point.

Attractive force is the negative gradient of the attractive potential, which is also represented by a mathematical function [1]:

$$F_{\mathrm{att}}(q)=-\nabla U_{att}(q)=\zeta(q_{goal}-q) \tag{10}$$

5.1.2 Repulsive potential function & repulsive force

The repulsive field keeps the robot away from obstacles. The strength of the repulsive field is usually proportional to the distance between the robot and the obstacle, that is, the closer it is to the obstacle, the greater the repulsive force. The repulsive potential function is usually expressed as [1]:

$$U_{\mathrm{rep}}(q)=\begin{cases}\frac{1}{2}\eta(\frac{1}{\rho(q.q_{obs})}-\frac{1}{\rho_0})^{2}\end{cases} \tag{11}$$

Where ρ0 is the distance threshold. When the distance to the obstacle is less than the threshold, the repulsive potential function is as shown above; when the distance to the obstacle is greater than the threshold, the repulsive potential function is 0.

The repulsive force is also the negative gradient of the repulsive force potential, which is expressed as follows [1]:

$$F_{rep}(q)=-\nabla U_{rep}(q)$$

$$=\begin{cases}\zeta(\frac{1}{\rho((q.q_{obs})-\frac{1}{\rho_0})}\frac{1}{\rho(q)^{2}}\frac{q-q_{obs}}{\rho(q)}\end{cases} \tag{12}$$

While the artificial potential field (APF) method exhibits certain advantages in path planning, it is prone to becoming ensnared in local optimal solutions. For instance, in scenarios where the robot becomes trapped in a location where multiple force fields nullify each other, it struggles to identify the optimal path to the goal. Moreover, APF encounters challenges in navigating complex environments. In instances characterized by numerous obstacles or constrained local spaces, the APF method may fail to generate a viable path, resulting in the robot either failing to reach the target or resorting to an unreasonable movement strategy.

5.2. APF Algorithm Improvement and Application Review

In the realm of robot motion planning, path planning methodologies leveraging artificial potential fields have garnered significant attention. Recognizing the inherent limitations of traditional APF algorithms, this section provides a brief overview of various improvement strategies explored in recent research endeavors, with the aim of furnishing guidance and support for future advancements in APF algorithms.

In a study outlined in [15], a novel approach termed DA-APF (Deterministic Annealing APF) is introduced, employing a deterministic annealing strategy for path planning. This algorithm incorporates temperature parameters to refine the potential field function, augment the influence of obstacles on the robot, mitigate the risk of local minima, and optimize initial path selection, thereby achieving a high success rate in path planning within complex environments. Leveraging simulated annealing and quenching processes, the DA-APF algorithm effectively addresses path planning challenges in static obstacle-laden environments, ensuring path smoothness, maintaining safe distances, and mitigating oscillations. The algorithm exhibits exceptional performance across various complex environments [15].

Furthermore, an APF-based mobile robot navigation path planning algorithm (P-APF algorithm) centered on deadlock avoidance is proposed to address the local minima issue inherent in traditional APF methods [16]. By integrating information regarding the robot's movement direction and front obstacles, this algorithm adeptly circumvents deadlock scenarios, thereby enhancing the accuracy and reliability of path planning. Particularly in the context of navigating unknown 2D environments to locate target points, the P-APF algorithm effectively overcomes deadlock and unreachability challenges, ensuring smooth traversal to the destination. Through simulation experiments, significant improvements over the traditional T-APF method are observed, characterized by reduced driving distances and maximized safe distances between the robot and obstacles. These findings underscore the promising applications of the P-APF algorithm in the realm of mobile robotics [16].

6. Discussion

In the realm of path planning, the map data furnished by Simultaneous Localization and Mapping (SLAM) technology serves as a crucial input for path planning algorithms, facilitating the robot's comprehension of its surrounding environment and enabling more precise path planning decisions. Concurrently, the outcomes of path planning endeavors can be relayed back to the SLAM system to optimize map updates and enhance positioning accuracy, thereby fostering synergy and collaboration between path planning and SLAM. While SLAM technology undeniably holds significance in path planning, this article primarily focuses on the comparative analysis of diverse path planning algorithms, rather than delving deeply into SLAM. Consequently, our primary emphasis remains on the path planning algorithm itself, with SLAM regarded as an auxiliary technology that supplements and fortifies path planning endeavors.

The primary advantage of Dijkstra's algorithm lies in its simple structure, which renders it relatively straightforward to implement and comprehend. Additionally, the algorithm boasts high efficiency, particularly in scenarios devoid of negative weighted edges, where its time complexity stands at O(n^2), enabling swift identification of the optimal path in the majority of cases [7]. Moreover, Dijkstra's algorithm exhibits lower space complexity compared to alternative path planning algorithms, necessitating storage solely for a distance array and a predecessor node array. Owing to its favorable time complexity and efficiency, the Dijkstra algorithm finds application in real-time systems, offering substantial advantages in the domain of autonomous mobile robots. Furthermore, it proves suitable for dynamic programming quandaries, adept at recalibrating to identify the optimal path in the event of edge weight alterations. However, as the node network expands, the algorithm's complexity escalates, resulting in heightened response times and diminished efficiency [11]. Moreover, in the presence of negative weight edges, the Dijkstra algorithm may yield erroneous outcomes, rendering it unsuitable for deployment in such scenarios.

The A* algorithm amalgamates the shortest path exploration of Dijkstra's algorithm with the heuristic search of the greedy algorithm, affording it efficient search capabilities and low time complexity. Widely employed in robot path planning, it swiftly discerns the best route to take the robot from its current location to the desired destination. Particularly in static maps, A* is esteemed as an effective approach for resolving the shortest path quandary [12]. Furthermore, it aids mobile robots in efficiently navigating safe and cost-effective routes, showcasing significant utility when integrated with SLAM technology to achieve global and local path planning optimality. However, A* confronts several challenges. Firstly, in the presence of a plethora of nodes, search times may elongate, especially within intricate environments, potentially impeding efficiency [11]. Secondly, issues such as excessive turning points, uneven paths, and elongated distances may yield suboptimal path planning outcomes [14]. Lastly, the resultant path may occasionally fall short of optimality, predisposing to local optimum solutions. Hence, continual refinement and optimization of the A* algorithm are imperative to enhance its path planning efficiency and optimization capabilities within complex environments.

The APF method boasts swift and uncomplicated execution, rendering it suitable for diverse robot types, including mobile robots and drones. It serves as a straightforward, efficient, and dependable motion planner for real-world targets, offering outstanding planning speed and real-time performance, with results swiftly generated during path planning endeavors [15, 17]. However, this method is susceptible to encountering local minima, leading to potential entrapment in suboptimal solutions and resulting in robot deadlock states. Moreover, it may encounter challenges in dense obstacle environments, potentially culminating in planning failures due to the absence of viable paths [15, 16]. Furthermore, the path trajectory may deviate from the equilibrium position, exhibit oscillations, or recurrently close loops, thereby impinging upon path planning efficacy. Consequently, there is a requisite to refine the algorithm to mitigate inherent deficiencies, such as the incorporation of distance factors, jump strategies, fuzzy controllers, and other methodologies aimed at mitigating local optima and path closure issues [15, 17].

In order to handle the complexities of intricate settings and dynamic fluctuations, path planning algorithms should be improved upon and optimized in future research initiatives. For Dijkstra's algorithm, there is a need to investigate methods to bolster efficiency in intricate environments and expansive node networks, as well as strategies to navigate negative weight edges effectively. As for the A* algorithm, avenues for exploration include refining heuristic functions and search strategies to enhance the quality and expediency of path planning outcomes. Furthermore, in the realm of the APF method, emphasis should be placed on ameliorating the local minimum conundrum and devising novel strategies to fortify its resilience and stability within complex environments. Such endeavors hold promise for advancing the efficacy and applicability of path planning algorithms in diverse real-world scenarios.

7. Conclusion

This paper offers a comprehensive analysis of the fundamental principles, research applications, and advanced algorithms pertaining to SLAM, Dijkstra's algorithm, A* algorithm, and APF algorithm in recent years, shedding light on their significance and potential applications in mobile robot trajectory planning. SLAM technology furnishes mobile robots with real-time positioning and map construction capabilities, thereby playing a pivotal role in path planning endeavors. While Dijkstra's algorithm stands out as a classic path planning approach widely adopted, its efficacy may falter in complex and dynamic environments. As an evolutionary refinement of the Dijkstra algorithm, the A* algorithm effectively mitigates its limitations through heuristic search techniques, albeit encountering challenges such as inflection points and non-smooth paths, warranting further enhancements and optimizations. Meanwhile, the APF method, renowned for its versatility across various robot types and operational efficiency, is susceptible to local optimal solutions and deadlock predicaments. Hence, in practical applications, judicious selection of an appropriate algorithm or amalgamation of algorithms tailored to specific scenarios is imperative to attain optimal trajectory planning outcomes. Future research endeavors could explore algorithmic amalgamations, adaptability in dynamic environments, and multi-robot collaboration to elevate the trajectory planning technology of mobile robots and foster its widespread adoption across diverse domains.

Author Contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Gul F, Rahiman W, Nazli Alhady S S. A comprehensive study for robot navigation techniques[J]. Cogent Engineering, 2019, 6(1): 1632046.

[2] Kuswadi S, Santoso J W, Tamara M N, et al. Application SLAM and path planning using A-star algorithm for mobile robot in indoor disaster area[C]//2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA). IEEE, 2018: 270-274.

[3] Raj R, Kos A. A comprehensive study of mobile robot: History, developments, applications, and future research perspectives[J]. Applied Sciences, 2022, 12(14): 6951.

[4] Fan Haiting, Du Yungang. Research on Navigation Algorithm for Mobile Robots Based on Laser SLAM[J]. Machine Tool & Hydraulic, 2021, 49(14): 41-46.

[5] Debeunne C, Vivet D. A review of visual-LiDAR fusion based simultaneous localization and mapping[J]. Sensors, 2020, 20(7): 2068.

[6] Taheri H, Xia Z C. SLAM; definition and evolution[J]. Engineering Applications of Artificial Intelligence, 2021, 97: 104032.

[7] Alshammrei S, Boubaker S, Kolsi L. Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance[J]. Comput. Mater. Contin, 2022, 72: 5939-5954.

[8] Guan-Zheng T A N, Huan H E, Sloman A. Ant colony system algorithm for real-time globally optimal path planning of mobile robots[J]. Acta automatica sinica, 2007, 33(3): 279-285.

[9] Li X. Path planning of intelligent mobile robot based on Dijkstra algorithm[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 2083(4): 042034.

[10] Wang C, Cheng C, Yang D, et al. Path planning in localization uncertaining environment based on Dijkstra method[J]. Frontiers in Neurorobotics, 2022, 16: 821991.

[11] Xu Z, Liu X, Chen Q. Application of improved Astar algorithm in global path planning of unmanned vehicles[C]//2019 Chinese Automation Congress (CAC). IEEE, 2019: 2075-2080.

[12] Liu H, Zhang Y. ASL-DWA: An improved A-star algorithm for indoor cleaning robots[J]. IEEE Access, 2022, 10: 99498-99515.

[13] Li, S. and Li, W. (2023), P-2.10: Design and Implementation of Mobile Robot Path Planning System Based on SLAM Technology. SID Symposium Digest of Technical Papers, 54: 513-516.

[14] Zhang Z, Wang S, Zhou J. A-star algorithm for expanding the number of search directions in path planning[C]//2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT). IEEE, 2021: 208-211.

[15] Wu Z, Dai J, Jiang B, et al. Robot path planning based on artificial potential field with deterministic annealing[J]. ISA transactions, 2023, 138: 74-87.

[16] Weerakoon T, Ishii K, Nassiraei A A F. An artificial potential field based mobile robot navigation method to prevent from deadlock[J]. Journal of Artificial Intelligence and Soft Computing Research, 2015, 5(3): 189-203.

[17] Chen G, Liu J. Mobile robot path planning using ant colony algorithm and improved potential field method[J]. Computational intelligence and neuroscience, 2019.