An Improved A* Algorithm for UAV Path Planning Problems

Jinchao Chen, Mengyuan Li, Zhenyu Yuan
Department of Computer Science
Northwestern Polytechnical University
Xi'an, China 710072
Email:cjc@nwpu.edu.cn, 1403254606@qq.com,
sherlockyuan@mail.nwpu.edu.cn

Qing Gu
Kunming Shipborne Equipment Research and Test Center,
China State Shipbuilding Corporation Limited
Kunming, China 650051
Email:gm90327@sina.com

***Abstract*—Recently unmanned aerial vehicle (UAV) has been widely applied in military and civil fields due to its strong autonomic and adaptability. Compared with the manned vehicles, UAV has significant advantages in carrying out the dangerous work by keeping human life away from risks. Although UAV provides notable benefits to practical applications, it gives rise to a complex path planning problem. The optimal flying path of a UAV should be obtained such that the flight length and time cost can be reduced as much as possible. In this paper, we study the path planning problem and propose an improved A* algorithm to solve the problem. First, with the models of UAVs and regions, an exact formulation based on mixed integer linear programming (MILP) is introduced to completely search the solution space. Then, by improving the evaluation function and the node selection strategy, an improved A* algorithm is presented to produce an optimal flight path for UAVs. Experimental results show that the approach proposed is more effective to solve the path planning problem than the traditional algorithms.**

***Index Terms*—unmanned aerial vehicle, path planing, A* algorithm, node selection strategy**

I. INTRODUCTION

Due to the strong autonomic and adaptability, unmanned aerial vehicle (UAV) has been widely adopted in military and civilian applications where manned operation is either unnecessary or dangerous [1], such as remote sensing information [2], image recognition [3], and military surveillance [4]. Compared with the manned vehicles, UAV does place human life in risks, and has significant advantages in carrying out the dangerous work.

Path planning plays an important role in UAV flight mission, which includes the information acquisition and processing of the terrain and the enemy, the establishment of the threats model, the design of the planning algorithm and the path tracking control [5]. Researches show the UAV self-limiting conditions, partial information of environment and limited sensor capabilities bring UAV path planning and optimization more challenges [6]. A reasonable planning path is beneficial to reduce the flight path length and improve the endurance performance. It is necessary to design an intelligent path planning algorithm with low complexity, fast convergence speed and adaptability to complex battlefield environment.

Based on the different task requirements and battlefield environment, researchers carry out different path planning methods [7], [8], such as: A* algorithm [9], [10], genetic algorithm [11], [12], ant colony algorithm [5], [13], artificial potential field method [14], [15] and so on. Among them, A* algorithm is a simple, scalable, and high reliable algorithm in the path planning algorithm, but its search efficiency is often affected by the selection of evaluation function. The calculation amount increases with the expansion of the search range, which often consumes huge memory and occupies lots of time [16], [17]. And in practical applications, the current path calculated by A* algorithm is often not the theoretical optimal path. Many scholars have conducted in-depth research on A* algorithm and improved it to suit the path planning of robots in different fields [18].

The objective of this paper is to develop a methodology for discovering a path for the UAV that meets mission goals, avoids collision, is optimal in terms of path length. In this paper, aiming at the shortcoming of A* algorithm that takes more time in large place, we improve the evaluation function and simplify the search space, so as to strive to find the optimal path as early as possible. By calculating the cosine of the initial node and the target node, the current node and the target node, and filtering the expansion node of larger angle, the search process is more inclined to extend the connection between the initial vertex and the target vertex. After above steps, the number of extend nodes in the path search process is small and the efficiency is higher.

The rest of this paper is organized as follows. Section II introduces the related work. Section III gives the models used in this paper and presents an exact formulation. Section IV proposes an improved A* algorithm. Section V conducts the experiments and analyses the performance of our approach. Finally, Section VI gives a summary of this paper and the directions for future work.

II. RELATED WORK

In order to realize the intelligent autonomous flight of UAVs, the track planning system should take arrival time, threat and path smoothness into consideration [19]. At present, researchers have proposed a variety of different UAV flight path planning algorithms. According to planning decisions, they

978-1-7281-4390-3/20/$31.00 ©2020 IEEE

can be divided into traditional mathematical programming solutions and modern intelligent optimization algorithms. Mathematical programming solutions include linear programming methods, dynamic programming methods, while intelligent optimization algorithms include heuristic search algorithm, genetic algorithm, swarm intelligence methods.

Linear programming (LP) is an important mathematical theory and method widely used in various fields [20]. In [21], The authors formulated the UAV path planning problem in NAS with sense and avoid capability in the framework of MILP. They proposed a method using the mathematical paradigm of mixed integer linear programming and provided a solution strategy in the dynamic sense. However, it is quite difficult for the expansion of model under the MILP framework. Moreover, LP does not have practical significance in path search because of the uncertainty of constraint conditions [22].

The ant colony algorithm in the intelligent optimization algorithm is a new bionic algorithm, which has good versatility, robustness and strong parallelism. The authors in [23] proposed a UAV path planning algorithm by using a multi-colony ACO algorithms to reach an optimal or near optimal solution quickly. In [24], the path searched by traditional ACO is trimmed afterwards to make it more straightforward. Straightening the path to reduce some unnecessary routes could make the strength shorter and cost less. However, the ant colony algorithm is easy to fall into an excessive search for a certain area, and the solution speed is slow when the number of the formation is large.

A* algorithm is a classic optimal heuristic search algorithm that uses heuristic information to find the optimal path. It is a direct search algorithm and is therefore widely used in path planning problems. In [10], the authors proposed Bidirectional Sparse A* Search algorithm (BSAS), which expands the start point and goal point respectively and calculates whether there is a Line of Sight (LOS) between every two different expansion nodes to get UAV path. The authors in [25] combined the A* algorithm and the least squares policy iteration algorithm together to get better path planning results. Although the A* algorithm is efficient, the dependency on the heuristic function is too strong, and the amount of calculation becomes larger as the search range increases, often consuming huge memory and taking a lot of time.

In view of the shortcomings of the previous studies, an improved A* algorithm is introduced to enhance the optimal extent and the search success rate of the robot path. The proposed algorithms have a higher success rate of UAV path planning and can get a more optimized path.

III. SYSTEM MODEL AND EXACT FORMULATION

A. Notations and System Model

In a real flight environment, the characteristics of the drone itself can determine the length of its maximum flight path. In this paper, a UAV is characterized by a tuple $U = \{V, W\}$ where V is its speed and W is the scan width of its sensor on the speed of V. The UAV has a mission of searching m regions $R = \{R_1, R_2, , R_m\}$, and find the optimal path efficiently. In the course of flight path planning, the terrain information on the flight should be considered. When the UAV has a flying mission, it is inevitable to encounter some obstacles such as mountains. We use an array $B = \{B_i \mid i \in [1, m]\}$ to represent whether there are barriers in region R_i. If there are impassable barriers the UAV can not fly over in region R_i, we assume $B_i = 1$. Table I summarizes the basic notations used.

TABLE I
BASIC NOTATIONS USED IN THIS PAPER

Symbol	Description
U	the UAV
R	the set composed of all regions
m	the number of regions in R
V	the speed of U
W	the scan width of the sensor of U
R_i	the ith region in R
B_i	the symbol representing whether barriers are in R_i

B. Exact Formulation

In this subsection, based on mixed integer linear programming (MILP), we present an exact formulation of the path planning problem for single UAV on multiple regions. The path planning problem is dened as minimizing a linear objective function (the minimum time cost of a single UAV) subject to linear constraints (coverage constraints of regions). We use grid method to model the flight space and make the assumptions that each step occupies the entire grid in the planned path and the position and size of the obstacle are known and do not change during the flight of the drone. For region R_i, its coordinates in the grid are expressed as (x_i, y_i). And the path length between adjacent regions is 1.

We use an array $P = \{p_{i,j} \mid i \in [1, m], j \in [1, m]\}$ to describe the flight paths of the UAV. Each element $p_{i,j}$ is a Boolean variable, representing whether the UAV flies from R_i to R_j. If the UAV ies from R_i to R_j, $p_{i,j} = 1$; otherwise $p_{i,j} = 0$, i.e.,

$$p_{i,j} = \begin{cases} 1, & \text{if the UAV ies from } R_i \text{ to } R_j \\ 0, & \text{otherwise} \end{cases}$$

With the flight path array P, the MILP formulation for the path planning problem can be written as,

$$\text{minimize } T = \sum_{i=1}^{m} \sum_{j=1}^{m} p_{i,j}$$

subject to

$$\forall i, j \in [1, m], if\ p_{i,j} = 1, \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq 1 \quad (1)$$

$$\forall i \in [1, m], if\ B_i = 1, \sum_{j=1}^{m} p_{i,j} = \sum_{j=1}^{m} p_{j,i} = 0 \quad (2)$$

$$\sum_{i=1}^{m} p_{0,i} = \sum_{j=1}^{m} p_{j,m} = 1 \quad (3)$$

$$\forall j \in [1, m], if \sum_{i=1}^{m} p_{i,j} = 1, \sum_{k=1}^{m} p_{j,k} = 1 \quad (4)$$

This formulation guarantees that all regions are fully covered, and seeks optimal values for the elements in the flight path array to minimize the maximum time cost of all UAVs. Constraint (1) guarantees the UAV can only fly over adjacent regions. Constraint (2) shows that if there are obstacles in region R_i, the UAV cannot fly into or out of the region R_i. Constraint (3) ensures initial position and target position are within the flight path. Constraint (4) represents that the UAV can not stop during the flight path.

IV. THE IMPROVED A* ALGORITHM

A. Limitations of the Original A* Algorithm

The A* algorithm heuristically searches nodes to construct the optimal path by calculating the value of the evaluation function of the path continuously. We use F(n) to represent the evaluation function from the initial node to the target node via node n, G(n) is the actual cost from the initial node to the current node in the state space, and H(n) is the estimated cost from the current node to the target node. According to the above definition, the evaluation function of A* algorithm can be obtained by

$$F(n) = G(n) + H(n) \tag{5}$$

The accuracy of the evaluation function F(n) plays a decisive role in the path planning performance and efficiency of the A* algorithm. In the original A* algorithm, the node selection only relies on the minimum F value and lacks other constraints. Therefore, many extended node non-optimal path nodes lead to the increase in space and time cost. In order to improve the search efficiency, we need to reduce search nodes and make path planning directional.

B. Evaluation Function Optimization

In view of the problems arising from the A* algorithm evaluation function, we add directional restrictions on the selection of the extended nodes during the operation of the Open table. As shown in Fig. 1 below, we assume that the initial node is S and the target node is E. Node A and Node B are the expandable nodes of the current node, and the line of the initial node to the target node is a vector $\vec{SE}$, the lines of the extension node of the current node to the target node are vector $\vec{AE}$,$\vec{BE}$.

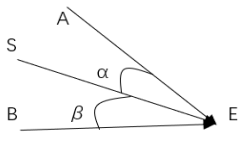

Fig. 1. Vector diagram

According to the cosine theorem of vector angle, we use formula (6) to calculate the angle α and β cosine values:

$$\cos\alpha = \frac{\vec{AE} \times \vec{SE}}{\left|\vec{AE}\right| \cdot \left|\vec{SE}\right|} \quad \cos\beta = \frac{\vec{BE} \times \vec{SE}}{\left|\vec{BE}\right| \cdot \left|\vec{SE}\right|} \tag{6}$$

In this paper, the angle range is from 0 to 90. The cosine function diagram is shown in Fig. 2, which illustrate its function value is monotonically decreasing. This ensures the monotonic constraint on the heuristic function. According to the vector diagram, the farther the distance from the vector $\vec{SE}$ is, the larger the included angle will be.

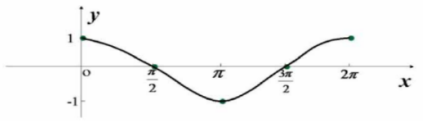

Fig. 2. Cosine function waveform

In order to reduce the investigation of useless nodes, the vector angle cosine value is introduced to improve the evaluation function, so that the selection of the extended node is more directional.

1) Construct a vector $\vec{SE}$ from the initial node to the ending node and a vector $\vec{AE}$ from the current node's extended node to the end node. Calculate the cosine of the angle between the two vectors $\cos\theta$.

2) Filter out the extended node whose cosine value is more than 0.5, so that the angle between $\vec{SE}$ and $\vec{AE}$ is less than 60.This optimization ensures that search path is closer to the destination.

3) Use ω for the weight, and the value can be adjusted according to the actual size of the map. Then we construct a new function named $H_1(n)$ the evaluation function of A* algorithm.

$$H_1(n) = H(n) - \omega\cos\theta \tag{7}$$

When the cosine function is added to the evaluation function, the extension node closer to the target node has a lower G value. Since the cosine function is a non-linear monotonic function, it can effectively constrain the search node range of the A* algorithm within a certain range. As a new heuristic function, our approach can better reduce the number of useless nodes and improve the efficiency of A* algorithm search time.

In the implementation of the algorithm, two linked lists are constructed. Set the $OPEN$ list to store the nodes to be accessed, and set the $CLOSED$ list to store the nodes that have been visited. The algorithm steps are as follows:

1) Put the starting node S into the $OPEN$ list.

2) Calculate the valuation function value of each node according to the improved cosine function optimization algorithm. Select the node with the lowest F value as the current position in the $OPEN$ list. Remove the node from the $OPEN$ list and put it into the $CLOSED$ list. Extend the current node to get four neighborhood nodes. The operation for the adjacent region is as follows:

(a) If the extended node is in the $CLOSED$ list or the obstacle area, skip the node. Otherwise turn to (b).

(b) If the extended node is not in the $OPEN$ list, add it into the $OPEN$ list; and have its parent pointer point to the location.

(c) If the node is already stored in the Open list, the G value is calculated first. If the G value is smaller than the original G value, the path satisfies the condition. Modify the parent pointer to the current position, and get the new F and G values, then reorder the F values in the OPEN list.

3) During the search, if any of the following situations occurs, the search is stopped. One case, the target location has been added to the $CLOSED$ list, which proves the shortest path has been found, then turn to (4). The other case, if the $OPEN$ list is empty but the target node is not found, there is no feasible path.

4) Save the trajectory of the search path. Return to the starting node from the target node by the parent pointer.

Algorithm 1: the improved A* algorithm proposed

Input: a starting node set S and a target node set E
Output: a flight path for the UAV

$OPEN \leftarrow \emptyset$, $CLOSED \leftarrow \emptyset$;
Put S into $OPEN$;
repeat
  $q \leftarrow$ the node with the lowest F value in $OPEN$;
  Remove q out from $OPEN$;
  Put q into $CLOSED$;
  if *q is the target node E* **then**
    break;
  end
  foreach *neighbor of q* **do**
    Compute F value according to Eq.(7);
    if *neighbor is in $CLOSED$* **then**
      break;
    end
    if *neighbor is in $OPEN$ and current F value is lower* **then**
      Update neighbor with the lower F value;
      Change neighbor's parent to q;
    end
    if *neighbor is not in both lists* **then**
      Put neighbor into $OPEN$;
    end
  end
until $OPEN \neq \emptyset$;

By improving the evaluation function of the A* algorithm, the search range is greatly reduced and the search efficiency of track points is improved, so as to ensure that UAVs can successfully complete.

V. EXPERIMENTS AND RESULTS

In this section we conduct experiments to analyze the performance of our approach. We mainly study the obstacle avoidance algorithm of UAVs in two-dimensional plane, which can make the drone fly autonomously. The algorithm gets rid of the limitation of the three-dimensional map and the influence of other factors of the path-finding algorithm, so that the experimental results are more accurate and clearer. In this experiment, the shortest flight distance searched by the breadth-first algorithm (BFS) is taken as the standard of path planning, and the original A* algorithm, cross product optimization A* algorithm (CP-A*) and ant colony optimization algorithm (ACO) are used as the control. In the 80*80 grid map, six groups of map obstacle rates are set as 10%, 15%, 20%, 25%, 30%, 35%. Each group carried out 100 simulation experiments, and the experimental results are averaged. The performance of the four algorithms is compared from the two aspects of search path accuracy and search path time.

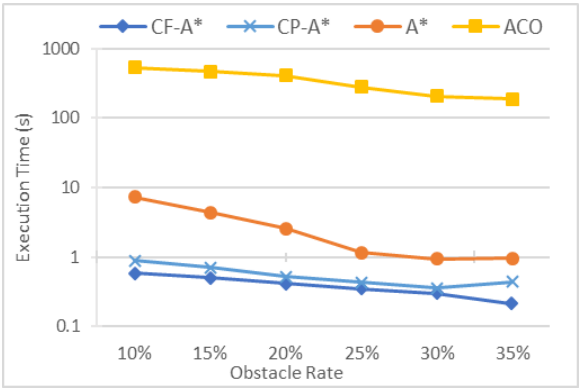

Fig. 3. Time costs obtained by our approach and other methods along with the increase of the obstacle rate

Fig. 3 shows that the average execution time required by different methods decreases along with the increase of the number of obstacles in random map. Among the four algorithms, the A* algorithm has obvious advantages over the ant colony algorithm in search time. The A* algorithm optimized by vector cross product can effectively reduce the search time in low-obstacle rate map. In the case of high obstacle rate, such as the obstacle rate of 35%, the average execution time of our approach is 77% less than that of original A* algorithm, which demonstrate the superiority of our approach.

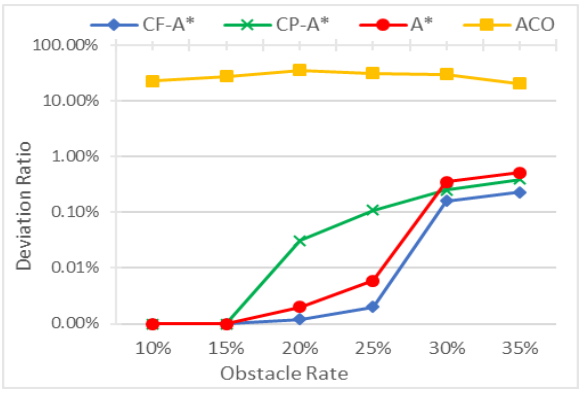

Fig. 4. Deviation ratio obtained by our approach and other methods along with the increase of the obstacle rate

Through varying the number of obstacles at random map, Fig. 4 shows the deviation rate of four methods path finding length. As can be seen in Fig. 4, the ant colony algorithm is not ideal in the larger map. The deviation rate of A* algorithm and its optimization algorithm could be controlled less than 1%. Among them, the A* algorithm optimized by vector cross product is efficient, but it sacrifices accuracy relatively. However, when the obstacle rate is 20%, the deviation of our approach is only 0.0012%, which ensures that the searched path is relatively optimal.

VI. CONCLUSIONS

As a heuristic search algorithm, A* algorithm has been widely used in solving the path planning problem. It improves the efficiency by reducing the search space, and plays an important role in real world applications. In this paper, we improve the A* algorithm, by changing evaluation function and improving the node selection strategy, in order to produce an optimal flight path for UAVs. Experimental results show that, the time cost and space utilization efficiency of our approach are better than the original algorithm, and our approach can efficiently to solve the UAV path planning problem.

VII. ACKNOWLEDGMENT

The work presented in this paper was supported in part by the National Key Research and Development Program of China No. 2017YFB1001900 and No. 2018YFB2101304, the Defense Industrial Technology Development Program No. JCKY2016607B006, the Special Civil Aircraft Research Program No. MJ-2017-S-39, and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] M. D. Flint, "Cooperative unmanned aerial vehicle (uav) search in dynamic environments using stochastic methods," 2005.

[2] E. Salami, C. Barrado, and E. Pastor, "Uav flight experiments applied to the remote sensing of vegetated areas," *Remote Sensing*, vol. 6, no. 11, pp. 11 051–11 081, 2014.

[3] P. Dan and L. Ichim, *Image Recognition in UAV Application Based on Texture Analysis*, 2015.

[4] M. A. Masum, M. K. Arrofi, G. Jati, F. Arifin, M. N. Kurniawan, P. Mursanto, and W. Jatmiko, "Simulation of intelligent unmanned aerial vehicle (uav) for military surveillance," in *International Conference on Advanced Computer Science & Information Systems*, 2013, pp. 161–166.

[5] C. Zhang, Z. Zhen, D. Wang, and M. Li, "Uav path planning method based on ant colony optimization," in *Control and Decision Conference (CCDC), 2010 Chinese*, 2010, pp. 3790–3792.

[6] A. Atyabi and D. Powers, "Review of classical and heuristic-based navigation and path planning approaches," *International Journal of Advancements in Computing Technology (IJACT)*, vol. 5, pp. 1–14, 01 2013.

[7] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, and P. Feyzeau, "Path planning: A 2013 survey," 2013, pp. 1–8.

[8] J. Chen, C. Du, X. Lu, and K. Chen, "Multi-region coverage path planning for heterogeneous unmanned aerial vehicles systems," in *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, 2019, pp. 3561–3565.

[9] H. T. Fan, T. T. Liang, C. H. Lee, D. C. Li, and C. C. Han, "A star search algorithm for civil uav path planning with 3g communication," in *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, 2014, pp. 942–945.

[10] B. Meng and X. Gao, "Uav path planning based on bidirectional sparse a* search algorithm," vol. 3, pp. 1106–1109, 2010.

[11] F. C. J. Allaire, M. Tarbouchi, G. Labonte, and G. Fusina, "Fpga implementation of genetic algorithm for uav real-time path planning," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1, pp. 495–510, 2009.

[12] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous uav," *Aerospace Science & Technology*, vol. 16, no. 1, pp. 47–55.

[13] U. Cekmez, M. Ozsiginan, and O. K. Sahingoz, "A uav path planning with parallel aco algorithm on cuda platform," 2014, pp. 347–354.

[14] J. Chen, C. Du, F. Xie, and Z. Yang, "Schedulability analysis of non-preemptive strictly periodic tasks in multi-core real-time systems," *Real-Time Systems*, vol. 52, no. 3, pp. 239–271, 2016.

[15] J. Chen, C. Du, F. Xie, and B. Lin, "Allocation and scheduling of strictly periodic tasks in multi-core real-time systems," in *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2016, pp. 130–138.

[16] L. H. O. Rios and L. Chaimowicz, "A survey and classification of a* based best-first heuristic search algorithms," 2010, pp. 253–262.

[17] J. Chen, C. Du, P. Han, and Y. Zhang, "Sensitivity analysis of strictly periodic tasks in multi-core real-time systems," *IEEE Access*, vol. 7, pp. 135 005–135 022, 2019.

[18] B. Fu, L. Chen, Y. Zhou, D. Zheng, Z. Wei, J. Dai, and H. Pan, "An improved a* algorithm for the industrial robot path planning with high success rate and short length," *Robotics and Autonomous Systems*, vol. 106, pp. 26–37, 2018.

[19] Z. Fu, J. Yu, G. Xie, Y. Chen, and Y. Mao, "A heuristic evolutionary algorithm of uav path planning," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11, 2018.

[20] J. Chen, C. Du, and P. Han, "Scheduling independent partitions in integrated modular avionics systems," *PloS one*, vol. 11, no. 12, p. e0168064, 2016.

[21] M. Radmanesh, M. Kumar, A. Nemati, and M. Sarim, "Dynamic optimal uav trajectory planning in the national airspace system via mixed integer linear programming," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 230, no. 9, pp. 1668 – 1682, 2016.

[22] J. Chen, C. Du, F. Xie, and B. Lin, "Scheduling non-preemptive tasks with strict periods in multi-core real-time systems," *Journal of Systems Architecture*, vol. 90, pp. 72 – 84, 2018.

[23] U. Cekmez, M. Ozsiginan, and O. K. Sahingoz, "Multi colony ant optimization for uav path planning with obstacle avoidance," in *International Conference on Unmanned Aircraft Systems*, 2016, pp. 47–52.

[24] Z. Qiannan, Z. Ziyang, G. Chen, and D. Ruyi, "Path planning of uavs formation based on improved ant colony optimization algorithm," 2014, pp. 1549–1552.

[25] L. Zuo, Q. Guo, X. Xu, and H. Fu, "A hierarchical path planning approach based on a* and least-squares policy iteration for mobile robots," *Neurocomputing*, vol. 170, pp. 257–266, 2015.