IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 21 July 2023, accepted 15 August 2023, date of publication 21 August 2023, date of current version 31 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307135

METHODS

The Dynamic Path Planning of Indoor Robot Fusing B-Spline and Improved Anytime Repairing A* Algorithm

MENGLONG CAO, BINGXIN LI, AND MENGGE SHI
College of Automation and Electronic Engineering, Qingdao University of Science Technology, Qingdao 266700, China
Corresponding author: Bingxin Li (347089154lb@gmail.com)

This work was supported in part by the Natural Science Foundation of Shandong Province under Grant ZR2020MF087, in part by the National Key Research and Development Program under Grant 2019YFA0706502-3.

ABSTRACT A path planning method fused B-spline curves, and an improved anytime repairing A* algorithm path planning method (BS-IARA*) is proposed to address the needs for timeliness, feasibility, and optimality in robot path planning in indoor dynamic environments. In this method, firstly, adaptive expansion factors and dynamic search step strategies are introduced in the initial stage and iterative process of the algorithm, respectively, to improve the low expansion factor preference defect and low search efficiency in the anytime repairing A* algorithm; Secondly, the resulting path is optimized using B-spline curves to enhance the smoothness and feasibility of the track. Simulation experiments in a static environment show that the introduced improved strategies can combine to enhance path quality and planning efficiency; simulation experiments in a dynamic environment show that the method can generate feasible paths quickly, with better path cost and smoothing than the standard anytime repairing A* and Randomized Weighted A* algorithms, and continuously improve the path optimality in a specified time to meet the needs of dynamic path planning.

INDEX TERMS Dynamic path planning, improved anytime repairing A* algorithm, B-spline curves.

I. INTRODUCTION

Path planning is a topical problem in indoor mobile robot research. The aim is for an indoor mobile robot to automatically plan an optimal and safe path from start to finish in an obstacle-laden working environment under unguided conditions. In dynamic environments, indoor mobile robots must react quickly to changes in the indoor environment, planning new and safe paths within a defined time frame to ensure collision-free operation while maximizing path optimization and planning efficiency to reduce the energy consumption of the mobile robot. Currently, the methods proposed for the path-planning problem of mobile robots are broadly classified into intelligent bionic algorithms [1], [2], [3], [4], [5], sampling-based planning algorithms [6], [7], and graph search algorithms [8], [9], [10], [11]. The intelligent bionic algorithm is difficult to apply in complex working scenarios containing dynamic obstacles due to its solution accuracy and optimization-seeking efficiency. Represented by the RRT algorithm, random sampling-based path planning methods have the ability to avoid falling into local optima and quickly plan safe paths, but the random nature of the sampling sacrifices robustness in the search for an optimum. Heuristic graph search algorithms are widely used in practical engineering because of their efficiency and stability. Still, it is challenging to solve the problem of generating better paths in complex dynamic environments within a specified time.

To solve the problem of low time efficiency and poor quality of traditional path planning algorithms in generating feasible paths in dynamic environments, many scholars have conducted research on heuristic anytime search algorithms [12], [13], [14], [15], [16]. These algorithms are capable of outputting the current optimal search result at any time after the start of the planning process, and the quality of the paths is continuously improved as the number of iterations increases, allowing them to solve dynamic path planning problems with a balance of planning timeliness and optimality of results. Likhachev et al. [17]proposed Anytime

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng H. Zhu.

Repairing A* (ARA*), which obtains better planning results by continuously reducing the expansion factor during the iterative process and making full use of the expansion information during each iteration to improve the efficiency of subsequent searches. Chakrabarti et al. [18] proposed Anytime Window A* (AWA*) to reduce the number of expansion nodes by searching iteratively within a restricted window region. It verified that it effectively improved the algorithm search efficiency when solving the backpack and TSP problems. Bhatia et al. [19] proposed a Randomized Weighted A* (RWA*) method to solve the benchmark function global search problem, which obtains higher search efficiency and robustness by randomizing the weighting factors in the immediate heuristic A* algorithm.

This paper proposes a fusion of B-sample curves and the Improved Anytime Repairing A* algorithm (BS-IARA*) for solving path-planning problems for mobile robots in dynamic environments. This method introduces an adaptive expansion factor and dynamic step size strategy to solve the issues of low expansion factor preference defect, algorithm efficiency sensitive to the scale of the planning area, and unsmooth planning path in the ARA* algorithm. Simulation results in a static environment show that the introduced improvement strategy effectively improves the optimality and smoothness of the generation path of the standard ARA* algorithm. The dynamic simulation results show that compared to the standard ARA* and RWA* algorithms, the BS-IARA* algorithm can generate feasible paths quickly, improve feasible path length and path smoothness effectively, and continuously improve path optimality within the specified time, meeting the needs of dynamic path planning.

The rest of the paper is organized as follows: Section II describes the standard ARA* and the improvement strategies introduced. Based on the B-spline curve path smoothing strategy is introduced in Section III. Experimental comparisons and results discussion are conducted in Section IV. Section V summarizes the article.

II. IMPROVED ANYTIME REPAIRING A* ALGORITHM

A. STANDARD ANYTIME REPAIRING A* ALGORITHM

The Anytime Repairing A* algorithm was proposed by Likhachev et al. [17], which performs the weighted A* algorithm multiple times by decreasing the expansion factor one at a time to obtain better planning results and to make full use of the expansion information during each iteration to improve the efficiency of subsequent searches. The general form of the cost function of the ARA* algorithm during node expansion can be expressed as:

$$key(s) = g(s) + \varepsilon \times h(s) \tag{1}$$

In equation (1), $g(s)$ denotes the true cost from the starting point to the current node s, $h(s)$ represents the estimated cost from the current node to the target node; ε is the expansion factor ($\varepsilon \geq 1$). During the iterative search of the path, ARA* simultaneously updates and saves three node chains, namely the OPEN list, the CLOSED list, and the INCONS list. When the ARA* algorithm search repeatedly extends to a node in the CLOSED list and causes the g value of that node to fall, the value of that node's generation is updated, and the node is placed in the INCONS list. The nodes in the INCONS list will not be expanded immediately but will be merged with the OPEN list and reordered for expansion during the next iteration of the search process. After completing a single path planning, if the planning time has not yet exceeded the limit, an expansion factor and node chain list update is performed, and this is used for the next weighted A* algorithm planning. The update of the expansion factor in the ARA* algorithm is expressed as a monotonically decreasing function:

$$\varepsilon = \varepsilon - \Delta\varepsilon \tag{2}$$

The empirical value of $\Delta\varepsilon$ from [17] is taken as 0.1 in (2).

In this section, the ARA* algorithm is improved to enable the generation of feasible paths quickly and the subsequent continuous optimization of existing paths based on the structure of the anytime repairing A* algorithm. By introducing the adaptive expansion factor, dynamic step size strategy, and smoothing path strategy to overcome the low expansion factor preference defect, low efficiency of the algorithm, sensitivity to the scale of the planning region, and unsmoothed planning path in the anytime repairing A* algorithm, the robustness and feasibility of the algorithm are improved. The following describes the improved ARA* algorithm process and improvement strategy.

B. ADAPTIVE EXPANSIVE FACTOR STRATEGY

The value of the expansion factor in the ARA* algorithm is closely related to the actual situation and can directly affect the results and time of the algorithmic operation. In the search process, the initial expansion factor determines the algorithmic efficiency and solution quality. A smaller initial expansion factor means a large number of node expansions, leading to a decrease in search efficiency; a more prominent initial expansion factor represents a high search efficiency but yields sub-optimal search results for the feasible paths. The choice of ε in the valuation function is a prerequisite for finding the optimal path in a finite amount of time, and the value of the expansion factor ε is usually set according to the different scales and complexity of the working environment of the mobile robot. In the improved algorithm proposed in this paper, the initial expansion factor can be varied with the scale of the working environment, which can balance the need of the algorithm for search efficiency and optimization results. As shown in (3):

$$\varepsilon = ceil(\frac{length + width}{2} \times \frac{1-c}{10}) \tag{3}$$

In equation (3), the function *ceil* ensures that the expansion factor $\varepsilon \geq 1$, *length* is the length of the environment map, *width* is the width of the environment map, and c represents the complexity of the working map, i.e., the density of static obstacles.

C. DYNAMIC SEARCH STEP STRATEGY

The search steps of the standard ARA* algorithm are set according to the actual requirements to ensure the mobile robot's safe operation. If the search step size is too large, the quality of the planned path could be better, while if it is too small, the solution's efficiency is affected, and it is easy to fall into local search. To balance the algorithm's initial path planning speed with the path's optimality, the algorithm in this paper uses a dynamic step strategy for planning in the iteration of the algorithm, i.e., a more significant search step is used at the beginning of the iteration. After the initial path is obtained, the search step is reduced iteratively to get a path with better optimality. In addition, the ARA* algorithm consumes a lot of computation time for cost calculation and sorting of OPEN list nodes. Using a more significant search step can effectively reduce the number of nodes in the OPEN list, improve the initial planning efficiency, and avoid the extension of the algorithm into local search in complex working areas. In the simulation environment, the maximum search step size l_{max} and the minimum search step size l_{min} are determined by considering the threat information and the planning performance of the mobile robot. This paper uses l_{max} as the initial extension step to improve the speed of the algorithm in planning possible paths. The search step size is reduced linearly as the number of iterations of the algorithm increases to enhance the quality of path planning and is not reduced when the step size is declined to l_{min}, as shown in (4):

$$l = l_{max} - \Delta l \tag{4}$$

In equation (4), Δl is the decreasing search step.

D. THE IMPROVED ANYTIME REPAIRING A* ALGORITH PROCESS

The pseudo-code of the improved ARA* algorithm is shown in Algorithm 1 and Algorithm 2, where Algorithm 2 is a sub-pseudo-code of Algorithm 1. The approximate steps of the improved ARA* algorithm are as follows:firstly determines according to equation (3) and plans a suboptimal path; next, it determines whether is reduced to 1 and exceeds the planning time, and if so, saves the result of the previous planning step, otherwise continue the search in the OPEN list and INCONS list obtained during the last planning step, reduce and according to equation (2) and equation (4) respectively, and repeat this step.

Algorithm 1 Improved ARA*

set $CLOSED = OPEN = INCONS = \emptyset$, $g(s_{start}) = 0$
insert s_{start} into $OPEN$ with $key(s_{start})$, $g(s_{goal}) = \infty$
determine ε according to (3)
while $\varepsilon \geq 1$ or NOT INTERRUPTED **do**
    ImprovePath
    get path
    save current suboptimal path
    decrease ε
    if $ExpandStep > l_{min}$ **then**
        decrease $ExpandStep$ according to (4)
    end if
    move states from $INCONS$ into $OPEN$
    update $OPEN$ according to (1)
end while

Algorithm 2 Improve Path

while $key(s_{goal}) > min_{s \in OPEN}(\text{key}(s))$ **do**
    remove s with the smallest $key(s)$ from $OPEN$
    $CLOSED = CLOSED \cup s$
    for each successor s' of s **do**
        if s' was not visited before **then**
            $key(s') = \infty$
        end if
        if $key(s') > g(s) + h(s, s')$ **then**
            $key(s') = g(s) + h(s, s')$
        end if
        if $s' \notin CLOSED$ **then**
            insert s' into $INCONS$
        end if
    end for
end while

III. BASED ON B-SPLINE CURVE PATH SMOOTHING STRATEGY

In practical indoor work scenarios, the resulting path should conform to the kinematic and dynamic constraints of the mobile robot. Therefore, the planned path should satisfy the requirement of smoothness, ensuring as far as possible that the planned path is identical to the actual movement trajectory. Path planning using the ARA* algorithm generates spikes at turns. Path planning using the ARA* algorithm generates spikes at turns, and the output of the currently planned optimal path requires smoothing of the global route to facilitate the smoothness of the mobile robot and to reduce unnecessary energy loss at path spikes. B-spline curves [20], [21], [22] are the most commonly used trajectory smoothing strategy in engineering practice and can effectively improve the smoothness and safety of paths.

Assuming that there are a total of n+1 control points, which are used to define the strike, the bounding range of the B-spline curve, the kth degree B-spline curve is defined as:

$$P(u) = \sum_{i=0}^{N} P_i B_{i,k}(u) \tag{5}$$

In equation (5), P_i is the control point, $B_{i,k}$ evaluated by equations (6) and (7), is the *ith* B-spline basis function of order k.

$$B_{i,0}(u) = \begin{cases} 1, u_i \leq u \leq u_{i+1} \\ 0, others \end{cases} \tag{6}$$

$$B_{i,k}(u) = \frac{u - u_k}{u_{i+k} - 1} B_{i,k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} B_{i+k,k-1}(u) \tag{7}$$

This paper chooses a third-order B-spline curve, i.e., $k = 3$, to balance the smoothness and computational complexity of the robot trajectory. The pseudo-code for path planning based on the improved ARA* algorithm and using B-spline curves are shown in Algorithm 3. First, create a map of the mobile robot's working environment and determine the starting and target points; Then, the globally optimal path from the starting position to the target position is planned using the improved ARA* algorithm; and finally generates a B-spline curve with planning curvature and obstacle avoidance safety distances using the global path coordinate points as control points, thus obtaining a continuous obstacle-free optimal feasible path.

Algorithm 3 BS-IARA* Algorithm

set s_{start} and s_{goal}
get the initial path through *Improved ARA* algorithm*
get the waypoints $(P_0, P_1, P_2, \ldots, P_n)$
$s_{start} = P_0, s_{goal} = P_n$
if $u < n-k+2$ **then**
  compute P_i according to (5)
  $u \leftarrow u + 1$
end if

IV. SIMULATION AND ANALYSIS

This section first validates the ARA* algorithm with different improvement strategies on maps of different scales and containing only static obstacles, comparing the computational timeliness and optimality of the algorithm from several aspects. Simulation experiments are then conducted in complex dynamic environments using the three algorithms, standard ARA*, RWA* [19], and BS-IARA*, respectively, to verify the path planning capability of the algorithms in dynamic environments and under limited planning time conditions.

A. STATIC ENVIRONMENTS SIMULATION VERFICATION AND ANALYSIS

The standard ARA* algorithm with various initial expansion factors and search steps and the ARA* algorithm applying different improvement strategies are used to maps with different scales to verify the effectiveness of the different improvement strategies. The map settings are shown in Table 1, and the path planning trajectories comparing various expansion factors, different step sizes, and using path smoothing strategies are shown in Figure 3(1), (2), and (3), respectively:

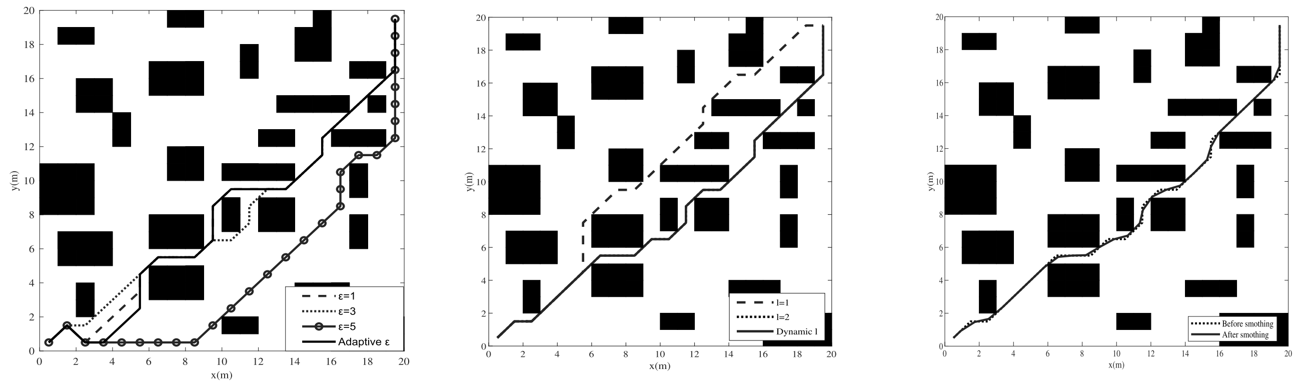
Table 2 counts the path planning time, path length, and the number of expansion nodes for the standard ARA* algorithm with different parameters and BS-IARA* algorithm in the environment built based on the information in Table 1, and the specific planning trajectory is shown in in Figure 1, Figure 2, and Figure 3. In particular, sub-schematic (a) shows the difference in the performance of different expansion factors in solving feasible paths. When the initial expansion factor of the standard ARA* is 1, the algorithm solves better paths, but the number of nodes in the OPEN table is high, and the efficiency of the solution cannot be guaranteed. The ARA* algorithm, which only introduces an adaptive initial expansion factor, does not ensure the optimal solution path. Still, it significantly reduces the solution time and improves the efficiency of planning feasible paths. Sub-schematic (b) compares the path planning results of the standard ARA* with different search step sizes and the ARA* algorithm with the sequential introduction of two improved strategies in different environments. The standard ARA* algorithm uses a fixed step size that does not balance the algorithm's search accuracy and solution efficiency. Although the improved ARA* algorithm does not ensure optimality in all metrics, the improved ARA* algorithm per-

TABLE 1. Information on static maps.

No.	Map size (m× m)	Complexity (%)	Start point	Target point
1	20×20	24.0	(1,1)	(20,20)
2	50×50	20.2	(1,1)	(50,50)
3	80×80	19.4	(1,1)	(80,80)

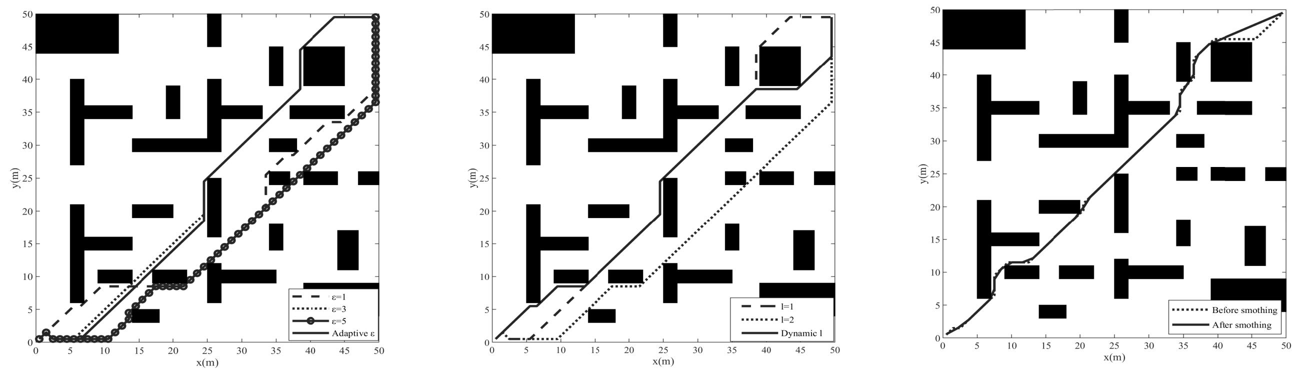
TABLE 2. Information on static maps.

No.	Different parameters or improvement strategies		Time to obtain the optimal path(s)	Path length(m)	Number of nodes in OPEN list
1	Using different ε	ε=1	0.179	29.79	49
		ε=3	0.041	32.97	44
		ε=5	0.037	35.9	41
		Adaptive ε	0.052	32.38	48
	Using different l	l=1	0.074	29.21	49
		l=2	0.054	29.80	34
		Dynamic l	0.068	29.80	47
	Applying B-spline curve		0.074	28.83	——
2	Using different ε	ε=1	0.195	76.57	165
		ε=3	0.129	78.67	164
		ε=5	0.133	78.33	157
		Adaptive ε	0.119	77.15	162
	Using different l	l=1	0.120	75.74	168
		l=2	0.099	77.74	158
		Dynamic l	0.115	76.57	160
	Applying B-spline curve		0.132	74.34	——
3	Using different ε	ε=1	0.312	121.68	277
		ε=3	0.319	122.51	280
		ε=5	0.199	123.68	266
		Adaptive ε	0.169	127.8	242
	Using different l	l=1	1.44	120.75	322
		l=2	0.206	130.12	251
		Dynamic l	0.260	122.51	261
	Applying B-spline curve		0.286	118.97	——



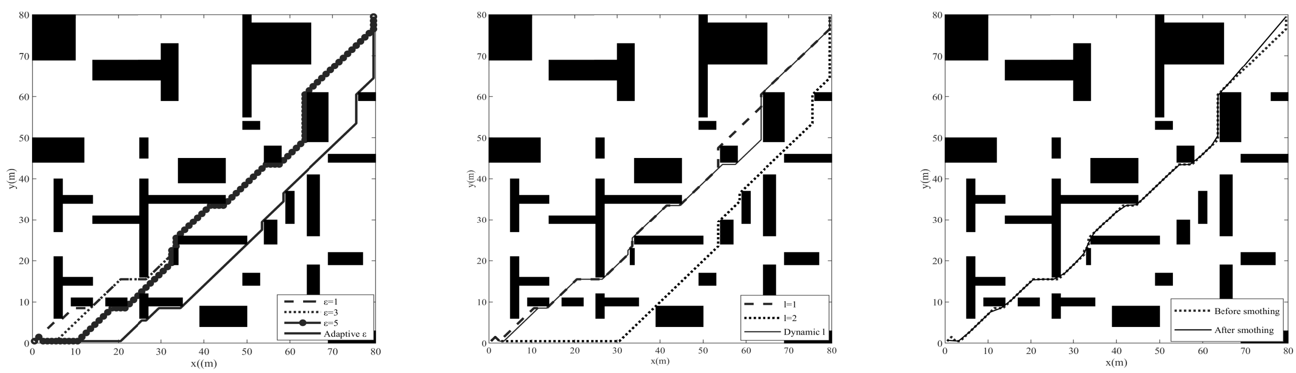
((a)) ((b)) ((c))

FIGURE 1. Diagram of different improvement strategies applied under Map 1. (a) is planning results with different expansion factors, (b) is planning results with different step sizes, (c) is a schematic of the comparison using the B-spline curve path smoothing strategy.



(a) (b) (c)

FIGURE 2. Diagram of different improvement strategies applied under Map 2. (a) is planning results with different expansion factors, (b) is planning results with different step sizes, (c) is a schematic of the comparison using the B-spline curve path smoothing strategy.



(a) (b) (c)

FIGURE 3. Diagram of different improvement strategies applied under Map 3. (a) is planning results with different expansion factors, (b) is planning results with different step sizes, (c) is a schematic of the comparison using the B-spline curve path smoothing strategy.

forms better in terms of combined solution time and quality of the solution. Sub-schematic (c) uses B spline curves for path smoothing. The number of nodes in the node chain table is not counted in Table 2 as the path smoothing strategy only optimizes the paths resulting from the improved ARA* algorithm planning. The spikes at the corners of the smoothed paths are well-optimized to ensure smoothness, while the path lengths are further reduced to meet practical requirements.

B. DYNAMIC ENVIRONMENTS SIMULATION VERFICATION AND ANALYSIS

The following experiments are conducted to verify the real-time path planning performance of the BS-IARA* algorithm in dynamic indoor environments. The following assumptions are first made for the mobile robot's operating environment:

1. Both dynamic and static obstacles are present in the indoor working environment.

((a)) ((b))

FIGURE 4. Dynamic obstacle moving vertically upwards at a rate of 4m/s. (a) 0∼10.5s dynamic obstacle moves vertically upwards. (b) 10.5∼21s dynamic obstacle encounters boundary and moves in the opposite direction.

TABLE 3. Table of dynamic obstacle information.

No.	Initial coordinates (m, m)	Radius (m)	Speed (m/s)	Direction
1	(0,25)	1	7	Horizontal to the right
2	(3,7)	1	10	Horizontal to the right
3	(50,20)	1	6	Horizontal to the left
4	(50,40)	1	9	Horizontal to the left
5	(15,50)	2	2	Vertical downward
6	(40,0)	2.5	4	Vertical upward

2. Moving obstacles in the working environment is simplified to outer circles.

3. The obstacle runs in a straight line and returns in the opposite direction when it reaches the boundary of the working environment.

4. Moving obstacles run at a known constant rate and in an available direction.

Dynamic obstacle movement is shown in Figure 4. Assume that the mobile robot moves at a rate of $5m/s$, the starting coordinates are $(0.5m,0.5m)$ and the ending coordinates are $(48.5m,48.5m)$, and the algorithm planning time is limited to $1s$.

Information on all dynamic obstacles in the working environment is shown in Table 3, and the three algorithm parameters are shown in Table 4:

The simulation results are shown in Figure 5, Figure 6, Figure 7, and Table 5. Figure 5, Figure 6, and Figure 7 show the schematic diagrams of the dynamic path planning process for the standard ARA* algorithm, the RWA* algorithm, and the BS-IARA* algorithm, respectively. Table 5 records the path cost, corner cost, and the number of extended nodes in the OPEN table for the three algorithms.

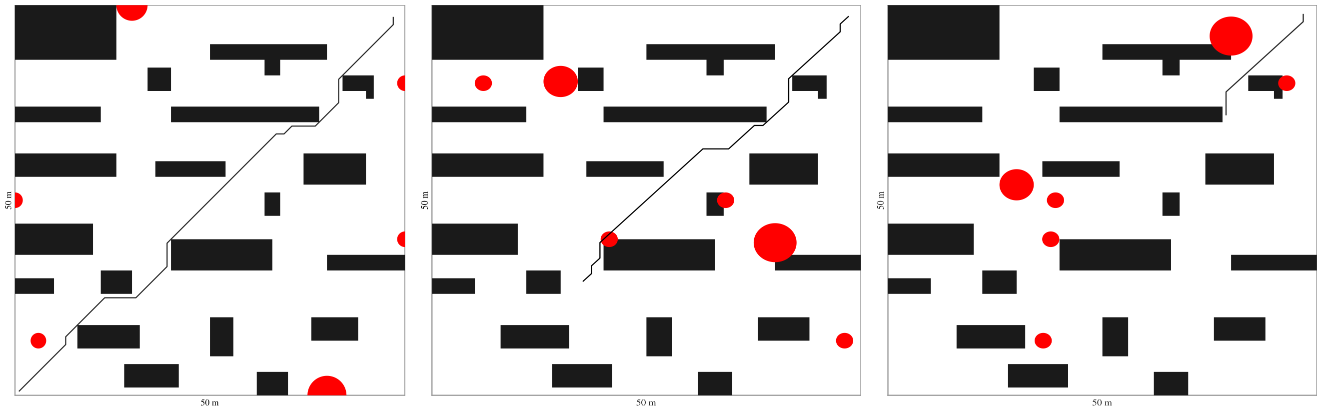
TABLE 4. Parameter settings.

Algorithms	Parameters	Set values
Standard ARA*	ε $\Delta\varepsilon$ l	3 0.1 1
RWA* [19]	w l	{1,1.5,2,3,4,5} 1
BS-IARA*	$\Delta\varepsilon$ l_{max} l_{min} Δl	0.1 2 1 0.2

TABLE 5. Table of dynamic obstacle information.

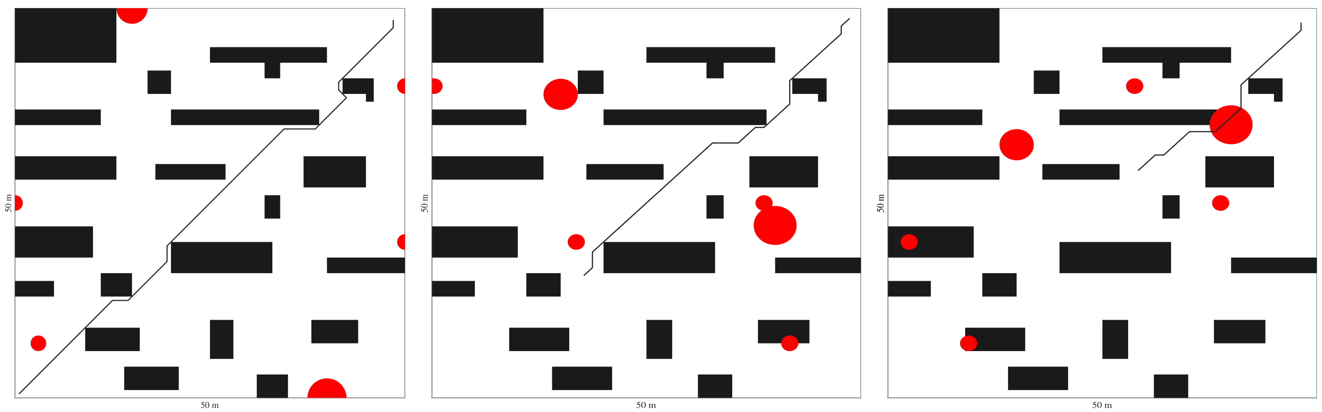
Merits	Standard ARA*	RWA*	BS-IARA*
Path length(m)	72	72	70.6
The Cost of the Corner(○)	720	540	252.81
OPEN list	184	198	163

As shown in Table 5, when using the BS-IARA* algorithm for dynamic path planning, the robot successfully avoids all obstacles in reaching the motion target (as shown in Figure 7), and the entire movement is 14.12s with a path cost of 70.6m. Compared to standard ARA* and RWA*, path lengths are 1.9% shorter, path corners are 64.9% and 53.19% lower, and expansion nodes are 11.41% and 17.68% lower, respectively. The above numerical simulation results show that the BS-IARA* algorithm proposed in this paper can find a feasible better path within the specified time limit and meet the path planning requirements in a dynamic environment. The path smoothing has been greatly improved with high planning efficiency and strong feasibility. Compared with the standard ARA* algorithm and RWA* algorithm, the BS-IARA* algorithm can plan feasible paths in a very



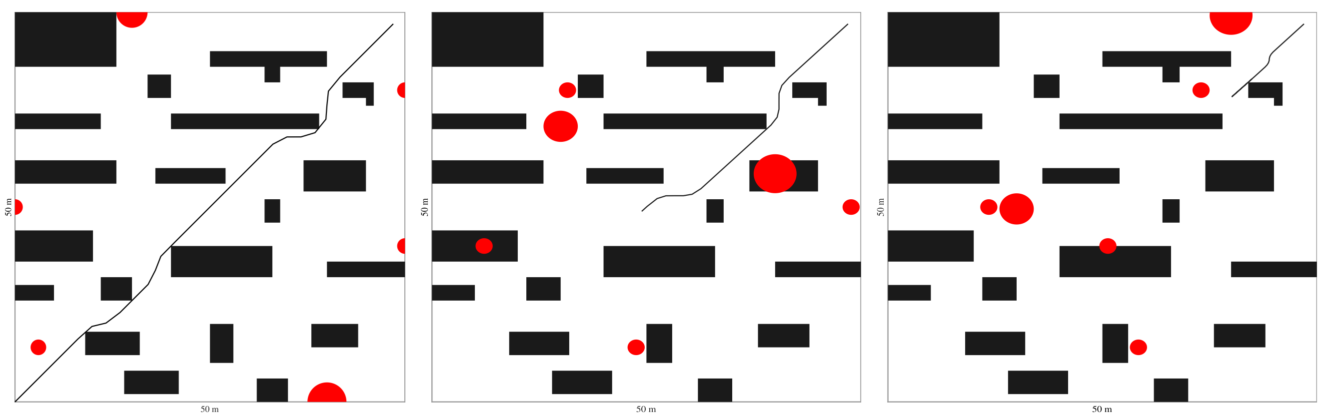
((a)) ((b)) ((c))

FIGURE 5. **Standard ARA* algorithm for planning paths in a dynamic environment. (a) ,(b) and (c) are schematic diagrams of the trajectories of the mobile robot at times 0s, 5s, and 7s, respectively.**



((a)) ((b)) ((c))

FIGURE 6. **RWA* algorithm for planning paths in a dynamic environment. (a) ,(b) and (c) are schematic diagrams of the trajectories of the mobile robot at times 0s, 5.3s, and 8.6s, respectively.**



(a) (b) (c)

FIGURE 7. **BS-IARA* algorithm for planning paths in a dynamic environment. (a) ,(b) and (c) are schematic diagrams of the trajectories of the mobile robot at times 0s, 7.2s, and 12s, respectively.**

short period and make full use of the planning time to continuously optimize the paths, which can search for better feasible paths in a static environment with higher efficiency and also adapt to the strict requirements of timeliness and optimality in a dynamic environment, with solid smoothness.

V. CONCLUSION

This paper proposes a fusion of B spline curves and an improved ARA* algorithm, which is able to plan feasible paths quickly and continuously improve the path quality as the planning time increases. An adaptive expansion factor and dynamic step size strategy are introduced to balance the quality and planning efficiency of the paths planned by the algorithm to address the shortcomings in the anytime algorithm. Simulation test results show that the planning efficiency of the BS-IARA* algorithm is higher than that of the RWA* and standard ARA* algorithms. It can instantly handle path planning problems in dynamic environments and has strong engineering practicality. This paper only considers the path-planning problem of robots in a two-dimensional environment. In the future, an attempt will be made to extend and apply the methodology of this paper to the path-planning problem of robots in a three-dimensional environment.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments, which significantly improved the paper.

REFERENCES

[1] J. Liu, J. Yang, X. Tian, M. Gao, and H. Liu, "An improved ant colony algorithm for robot path planning," *Soft Comput.*, vol. 21, no. 19, pp. 5829–5839, May 2016.

[2] C. Huang and J. Fei, "UAV path planning based on particle swarm optimization with global best path competition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 6, Jun. 2018, Art. no. 1859008.

[3] N. Lin, J. Tang, X. Li, and L. Zhao, "A novel improved bat algorithm in UAV path planning," *Comput., Mater. Continua*, vol. 61, no. 1, pp. 323–344, 2019.

[4] F. Li, X. Fan, and Z. Hou, "A firefly algorithm with self-adaptive population size for global path planning of mobile robot," *IEEE Access*, vol. 8, pp. 168951–168964, 2020.

[5] J. Liu, X. Wei, and H. Huang, "An improved grey wolf optimization algorithm and its application in path planning," *IEEE Access*, vol. 9, pp. 121944–121956, 2021.

[6] I. Noreen, A. Khan, and Z. Habib, "A comparison of RRT, RRT* and RRT*-smart path planning algorithms," *Int. J. Comput. Sci. Network Secur.*, vol. 16, no. 10, p. 20, 2016.

[7] Z. Tahir, A. H. Qureshi, Y. Ayaz, and R. Nawaz, "Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments," *Robot. Auton. Syst.*, vol. 108, pp. 13–27, Oct. 2018.

[8] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning A*," *Artif. Intell.*, vol. 155, nos. 1–2, pp. 93–146, 2004.

[9] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Proc. Eng.*, vol. 96, pp. 59–69, 2014.

[10] A. K. Guruji, H. Agarwal, and D. K. Parsediya, "Time-efficient A* algorithm for robot path planning," *Procedia Technol.*, vol. 23, pp. 144–149, Jan. 2016.

[11] T. XiangRong, Z. Yukun, and J. Xinxin, "Improved A-star algorithm for robot path planning in static environment," *J. Phys., Conf. Ser.*, vol. 1792, no. 1, Feb. 2021, Art. no. 012067.

[12] E. A. Hansen and R. Zhou, "Anytime heuristic search," *J. Artif. Intell. Res.*, vol. 28, pp. 267–297, Mar. 2007.

[13] M. Likhachev, D. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun, "Anytime dynamic A*: An anytime, replanning algorithm," in *Proc. ICAPS*, vol. 5, 2005, pp. 262–271.

[14] J. Benton, M. Do, and S. Kambhampati, "Anytime heuristic search for partial satisfaction planning," *Artif. Intell.*, vol. 173, nos. 5–6, pp. 562–592, Apr. 2009.

[15] V. Narayanan, M. Phillips, and M. Likhachev, "Anytime safe interval path planning for dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 4708–4715.

[16] A. A. Maw, M. Tyan, and J.-W. Lee, "IADA*: Improved anytime path planning and replanning algorithm for autonomous vehicle," *J. Intell. Robot. Syst.*, vol. 100, nos. 3–4, pp. 1005–1013, Dec. 2020.

[17] M. Likhachev, G. J. Gordon, and S. Thrun, "ARA*: Anytime A* with provable bounds on sub-optimality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2003.

[18] S. Aine, P. P. Chakrabarti, and R. Kumar, "AWA*—A window constrained anytime heuristic search algorithm," in *Proc. Int. Joint Conf. Artif. Intell.* Morgan Kaufmann, 2007, pp. 2250–2255.

[19] A. Bhatia, J. Svegliato, and S. Zilberstein, "On the benefits of randomly adjusting anytime weighted A," in *Proc. Int. Symp. Combinat. Search*, 2021, pp. 116–120.

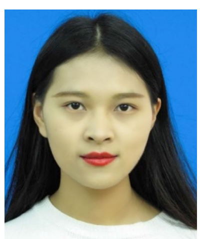
[20] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 167–172, Jan. 2010.

[21] C.-C. Tsai, H.-C. Huang, and C.-K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4813–4821, Oct. 2011.

[22] S. A. Eshtehardian and S. Khodaygan, "A continuous RRT*-based path planning method for non-holonomic mobile robots using B-spline curves," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 7, pp. 8693–8702, Jul. 2023.

MENGLONG CAO was born in Shandong, China, in 1971. He received the Ph.D. degree in navigation, guidance, and control theory from the Harbin Institute of Technology, Harbin, China, in 2009. He is currently a Professor with the College of Automation and Electronic Engineering, Qingdao University of Science and Technology, China, where he is a Principal Member of the Institute of Autonomous Navigation and Intelligent Control. He is currently the Vice Dean of the College of Automation and Electronic Engineering, Qingdao University of Science and Technology. His research interests include intelligent control, autonomous navigation, and information fusion.

BINGXIN LI received the bachelor's degree in electrical automation from the Qingdao University of Science and Technology, in 2020, where she is currently pursuing the master's degree in electronic information. Her research interests include path planning for mobile robots and intelligent optimization algorithms.



MENGGE SHI received the degree in electrical automation from the Qingdao University of Science and Technology, in 2020, where she is currently pursuing the master's degree in electronic information. Her research interests include autonomous navigation and intelligent control.