Computers & Industrial Engineering 155 (2021) 107173

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A fast robot path planning algorithm based on bidirectional associative learning

Meng Zhao [a], Hui Lu [a,*], Siyi Yang [a], Yinan Guo [b,c], Fengjuan Guo [d]

[a] School of Electronic and Information Engineering, Beihang University, Beijing 100191, PR China
[b] China University of Mining and Technology, Xuzhou, Jiangsu 221116, PR China
[c] China University of Mining and Technology (Beijing), Beijing 100084, PR China
[d] Shaanxi Key Laboratory of Integrated and Intelligent Navigation, Shaanxi, Xi'an 710068, PR China

ARTICLE INFO

Keywords:
Path planning
Bidirectional associative learning
Experience guidance
Search scope
Planning efficiency

ABSTRACT

Fast path planning in unknown environment is important to reduce the loss of human and material resources. To reduce planning time while obtaining a short path, this paper proposes a Bidirectional Associative Learning Algorithm (BALA). In the proposed algorithm, an episode is defined as a bidirectional movement between the start point and the target point. The planning process in the BALA is divided into three stages: early stage, medium stage and end stage. In the early stage, the attraction of the target point is adopted to instruct the robot to select action. This strategy not only helps the robot avoid blind search, but also provides the search scope that may contain the global shortest path for the subsequent episodes. In the medium stage, we propose an action selection strategy based on the experience guidance, where the experience obtained in the obverse and reverse movements is used alternately to improve the learning efficiency of the robot. In the end stage, a strong connectivity relationship between nodes is defined. Planning by this relationship, the length of the final planned path will be the shortest based on the experience the robot obtains. The comparison results with Q-Learning and its improved algorithm reveal that the BALA demonstrates desirable and stable performance in planning efficiency in any environment, and it can well balance the planning time and path length. Additionally, the practicability of the proposed algorithm is validated on Turtlebot3 burger robot.

1. Introduction

Path planning is one of the key technologies of robot autonomous navigation system, which is the basis for the robot to complete complex tasks (Olcay, Schuhmann, & Lohmann, 2020; Tang, Sun, Yu, Chen, & Zheng, 2019). In unknown environment, it usually takes a lot of time to obtain environmental information and accumulate path experience, thus the efficiency of path planning often determines the work efficiency of the robot. More importantly, fast path planning is the key to reduce loss in the task such as environment exploration, search and rescue. Therefore, improving the planning efficiency of the robot in unknown environment is essential, which contributes to expand the application fields of the robot.

A large number of algorithms that can be used for robot path planning in unknown environment have been proposed so far (Han, 2019; Osanlou, Bursuc, Guettier, Cazenave, & Jacopin, 2019). In order to improve the autonomy and intelligence of the robot, most algorithms focus on improving the autonomous learning ability of the robot (Low, Ong, & Cheah, 2019; Wei & Ni, 2018). However, these algorithms have not been widely used in practice considering the following obstacles. First, the effectiveness of the algorithm usually relies on the accurate fusion of multi-sensor information. Second, there are many parameters in most algorithms, and these parameters have great influence on the planning efficiency. Third, to obtain the global shortest path as far as possible, the convergence rate of the algorithm is slow, which does not meet the need for fast path planning in some scenarios. Finally, the robot learns from the start point in each episode of some algorithms, which does not conform to the continuity of the robot position in practical application (Watkins & Dayan, 1992).

In our previous research (Zhao, Lu, Yang, & Guo, 2020), it has been demonstrated that the shortest distance information can be regarded as experience and the memory of it helps to improve the planning

* Corresponding author.
E-mail addresses: meng960416@163.com (M. Zhao), mluhui@buaa.edu.cn (H. Lu), siyiyang0215@foxmail.com (S. Yang), guoyinan@cumt.edu.cn (Y. Guo), gfj118@yeah.net (F. Guo).

https://doi.org/10.1016/j.cie.2021.107173
Received 14 October 2020; Received in revised form 20 January 2021; Accepted 29 January 2021
Available online 11 February 2021

efficiency. In this paper, the work has been extended to further reduce the planning time while maintaining the optimization ability. Considering the application requirements, the Bidirectional Associative Learning Algorithm (BALA) is proposed.

Considering that the position of robot should be continuous in the practical application, we redefine the episode as a back-and-forth movement between the start point and the target point. Therefore, in addition to the EM (Experience-Memory) table and IS (Instructions) table used in the previous work (Zhao, Lu, Yang, & Guo, 2020), we define the EMR (Experience-Memory-Reverse) table and ISR (Instructions-Reverse) table to record the reverse experience the robot obtains when it moves from the target point to the start point. The EMR table is used to record the shortest distance from each node to the target point and the ISR table is adopted to record the parent node of each node based on the shortest distance in the reverse movement. In addition, the length of the shortest path through each node is recorded in the EMT (Experience-Memory-Total) table. Based on the experience, the path planning based on the BALA can be divided into three stages. In the early stage of inexperience, the robot chooses the action by referring to the attraction of the target point. In the medium stage, the strategy of experience guidance is adopted by the robot to choose an action within the search scope provided in the early stage. Here, the experience recorded in the ISR and IS will be used respectively in the obverse and reverse movements. In the end stage, the path can be planned according to the strong connectivity relationship between nodes.

The contributions to fast path planning in unknown environment can be described in three aspects.

(1) A Bidirectional Associative Learning Algorithm for robot fast path planning in unknown environment is proposed. The definition of the episode in the proposed algorithm considers the continuity of robot position in the practical application. Therefore, this algorithm provides an opportunity to combine the theoretical research on path planning in unknown environment with practical application.
(2) We design a set of action selection strategies which the robot uses in the early stage of path planning. These strategies not only enable the robot to obtain more valuable environment information even when there are some concave obstacles in the environment, but also provide the search scope which may contain the global shortest path for the subsequent episodes.
(3) An action selection strategy based on experience guidance is proposed to improve the learning efficiency of the robot. Based on this strategy, the obverse experience and reverse experience are used alternately to select the action when the robot moves between the start point and the target point. The combination with the strategy of selecting actions randomly makes it possible to greatly improve the planning efficiency of the robot while having a small impact on the length of the final planned path.

In order to verify the effectiveness of the proposed algorithm, a series of experiments is performed. The experiment results in the episode times, planning time and path length are compared with the Q-Learning algorithm and the algorithm we proposed earlier. The comparison results show that the BALA has desirable performance in terms of planning efficiency and optimization ability. In addition, the practicability of the proposed algorithm is verified on Turtlebot3 burger robot.

The rest of this paper is organized as follows. Section 2 gives a summary of the research status for robot path planning algorithm with autonomous learning ability. The detailed description of the BALA is introduced in Section 3. Section 4 shows and discusses the experiment results. Section 5 summarizes this paper and points out our future research directions.

2. Related work

A wide variety of path planning algorithms that can be used in unknown environment have been proposed so far. The algorithms with autonomous learning ability can be roughly divided into three categories, which are respectively based on evolutionary algorithm, neural network and reinforcement learning.

Referring to the law of biological evolution, evolutionary algorithm (EA) can get the optimal solution of complex engineering technology step by step through the selection mechanism of survival of the fittest. Therefore, the self-learning ability of EA enables it to be used in many fields (Cheng, Ma, Lu, Lei, & Shi, 2021; Lu, Zhang, Fei, & Mao, 2015), including path planning. Due to the similarity of the optimization process, the genetic algorithm and ant colony algorithm are the two most commonly used algorithms in robot path planning (Zeng, Yang, & Xu, 2009; Guo, Mao, Ding, & Liu, 2019; Santos, Oório, Toledo, Otero, & Johnson, 2016; Shi & Cui, 2010). They can also be used to optimize the parameters of other planning algorithms (Hassanzadeh & Sadigh, 2009). In addition, particle swarm optimization algorithm, firefly algorithm and bee colony algorithm can also be adopted to instruct the robot to complete the path planning task safely in unknown environment (Dadgar, Jafari, & Hamzeh, 2016; Liang & Lee, 2015; Patle, Pandey, Jagadeesh, & Parhi, 2018).

The principle of evolutionary algorithm is simple, but the effectiveness of it is affected by the design of fitness function. However, it is difficult to evaluate the results of robot movement effectively in unknown environment. Therefore, when the robot is unable to obtain any prior information of the environment, the performance of evolutionary algorithm will be limited.

The neural network can enable the robot learn the mapping from state space to action space based on the data set about specific scenes, so that the robot can plan a feasible path in real time even in unknown environment. In the early days, scholars used the abilities of learning and competing provided by the neural network to enable the household robot to clean larger areas with low energy consumption (Tse, Lang, Leung, & Sze, 1998; Yang & Luo., 2004). With the deep research on neural network, various path planning algorithms based on the improved neural network emerge in endlessly (Edvardsen, 2019; Lebedev, Steil, & Ritter, 2005; Luo et al., 2019; Mu et al., 2019; Qu, Yang, Willms, & Yi, 2009). In order to enhance the path optimization ability, some scholars combined the neural network with classic path planning algorithms (Li, Meng, Li, & Chen, 2008; Qian, Liu, Tian, & Bao, 2020; Wang, Chi, Li, Wang, & Meng, 2020). In addition, since the autonomous localization is the basis of robot path planning, the application of neural network to the simultaneous localization and mapping (SLAM) is also one of the research hotspots (Li, Song, & Hou, 2015).

Apparently, the use of neural network can greatly improve the real-time of path planning and make the robot have the ability of online planning. However, due to the poor stability of the reasoning ability for the neural network, the effectiveness of the algorithm is affected by the completeness of the previous training data, so the application fields of this type of algorithm are limited.

Reinforcement learning algorithm can enable the robot to obtain the ability of intelligent decision-making through trial and error even there is no prior knowledge. As one of the most classic reinforcement learning algorithms, Q-learning has been used up to now (Watkins & Dayan, 1992). To improve the practicability of the algorithm in path planning, many improved algorithms were proposed. These improvements mainly include three aspects: the type of experience recorded in the Q table (Konar, Goswami Chakraborty, Singh, Jain, & Nagar, 2013; Low et al., 2019), the design of reward function (Li, Fu, Wang, & Hu, 2019; Yan & Xiang, 2018), and the strategy of action selection (Li et al., 2019; Li, Xu, & Zuo, 2015). However, although these algorithms are effective in improving the convergence rate, they still cannot solve the problem well when the state space is continuous. Therefore, some scholars proposed to introduce neural network to describe the value of Q-function and

applied the algorithm to path planning (Sharma, Andersen, Granmo, & Goodwin, 2020; Zhou, Liu, Xu, & Guo, 2018). Apart from that, some famous reinforcement algorithms such as State-CactionCrewardCstateCaction (Sarsa) and Asynchronous Advantage Actor-Critic (A3C) can also be used for path planning in unknown environment (Li, Chen, & Le, 2018; Luo, Tang, Fu, & Eberhard, 2018).

Because the reinforcement learning can give the robot the ability to learn online, the research on this type of algorithm is in an active stage. However, the state space and action space of the actual problems are mostly continuous, and the high or low sampling efficiency have a great impact on the experiment results. Therefore, it is a challenging problem to improve the convergence rate and optimization ability of the algorithm. In addition, the difficulty in designing reward function is also one of the important reasons for limiting its application fields.

In summary, numerous algorithms are effective for robot path planning in unknown environment and each has its own merits (Table 1). According to the above discussion, the reinforcement learning algorithm is more effective for robot path planning in the completely unknown environment. However, to apply it in practical application, there are still many problems to be solved. First, the planning efficiency needs to be improved to enhance the flexibility of the robot in the planning process. Second, the stability of the algorithm needs to be improved, where the algorithm is better to be less influenced by human factors. Third, the optimization ability of the algorithm needs to be enhanced to reduce the energy consumption of the robot in the movement. Considering these requirements, this paper proposes the Bidirectional Associative Learning Algorithm.

3. The bidirectional associative learning algorithm

3.1. The framework of algorithm

In the BALA, the path planning ability of the robot is enhanced by using the experience recorded in five tables: EM, IS, EMR, ISR, EMT. The definition of each table is given in Table 2. As the framework of the BALA shown in Fig. 1, the experience recorded in the EM and IS will be updated in the obverse movement and used in the reverse movement of the robot. When the robot moves from the target point to the start point, the experience recorded in the EMR and ISR will be updated, and this reverse experience is adopted to instruct the robot to move towards the target point. At the end of each episode, the nodes with strong connectivity and the length of the shortest path which passes through these nodes will be recorded in the EMT. When the EMT converges, the path can be planned based on the experience in the IS and ISR.

Based on these experience tables, the planning process of the robot in unknown environment can be divided into three stages. In the early stage, the path length of the robot is usually the longest in the whole planning process since there is no experience. After one episode, the robot will enter the medium stage of path planning, where it can choose the action under the guidance of the experience tables. In the end stage, the robot plans the shortest path based on the experience it obtains. Therefore, in the whole process of path planning, the path length of the robot in an episode has a decreasing trend and eventually converges, which can be described as Fig. 2.

To provide a clear description, the basic notations used in the BALA are listed in Table 3.

Table 1
The features of the algorithms used for path planning in unknown environment.

Algorithm	Features
Evolutionary algorithm	**Strengths**: The principle of the algorithm is simple. **Weaknesses**: It is difficult to design an effective fitness function.
Neural network	**Strengths**: The real-time performance of path planning is high. **Weaknesses**: The stability of the algorithm is limited.
Reinforcement learning	**Strengths**: No prior information about the environment is required. **Weaknesses**: Slow convergence speed and unstable optimization ability.

Table 2
The definitions of the tables used in the BALA.

Table	The experience recorded in the table
EM (Experience-Memory)	The node and the shortest distance from this node to the start point.
IS (Instructions)	The node and its parent node. (These relationships are obtained in the obverse movement.)
EMR (Experience-Memory-Reverse)	The node and the shortest distance from this node to the target point.
ISR (Instructions-Reverse)	The node and its parent node. (These relationships are obtained in the reverse movement.)
EMT (Experience-Memory-Total)	The node and the length of the shortest path through this node.

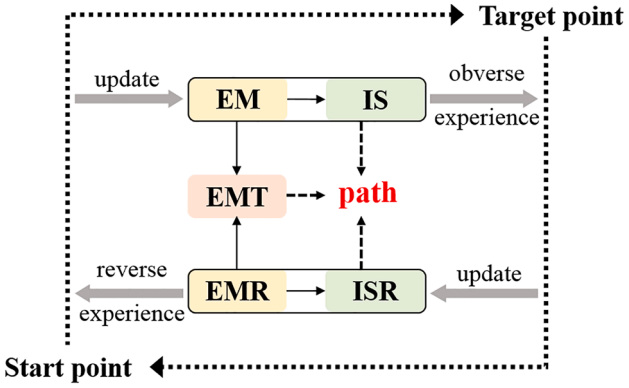

Fig. 1. The framework of the BALA.

3.2. The early stage of path planning

The experience the robot obtains in the early stage of path planning is the basis of subsequent episodes, which determines the planning efficiency to a certain extent. However, since there is no experience about the environment, it usually takes a long time for the robot to find a feasible path in the first episode. To solve this problem and make the robot accumulate more experience in this stage, we improve the action selection strategy.

Since the position of T is the only knowledge that the robot can obtain in the early stage of path planning in unknown environment, in addition to selecting the action randomly to get more experience, we consider the attraction of T in the selection of action. Based on this strategy, the node closest to T and S will be chosen respectively by the robot in the obverse and reverse movements. Therefore, the process of action selection can be described as (1), (2).

$$d(s_t^{a_i}, T) = \sqrt{(x_{s_t^{a_i}} - x_T)^2 + (y_{s_t^{a_i}} - y_T)^2} \quad (1)$$

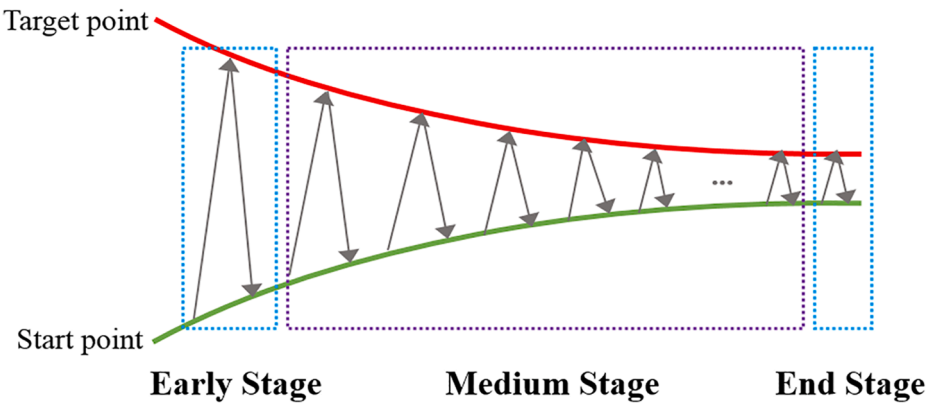

Fig. 2. The process of path planning based on the BALA.

Table 3
Basic notations.

Notation	Definition
A_t	The set of the feasible actions for the robot in the tth step. $A_t = \{a_1, \ldots, a_n\}$
A_t^{u}	The set of the feasible actions for the robot which have not been selected before. $A_t^{\mathrm{u}} \subseteq A_t$
S	The start point.
T	The target point.
F	The set of the nodes that the robot has reached over 100 times.
s_t	The position of the robot in the tth step.
a_t	The action of the robot in the tth step.
$s_t^{a_i}$	The node that the robot can reach after choosing action a_i in the tth step.
(x_h, y_h)	The coordinates of node h.
p_h^{IS}	The parent node of the node h, which is recorded in the IS.
c_h^{IS}	The child node of the node h, which is recorded in the IS.
p_h^{ISR}	The parent node of the node h, which is recorded in the ISR.
c_h^{ISR}	The child node of the node h, which is recorded in the ISR.
l_h^{EM}	The shortest distance from the node h to the S, which is recorded in the EM.
l_h^{EMR}	The shortest distance from the node h to the T, which is recorded in the EMR.
l_h^{EMT}	The length of the shortest path through the node h, which is recorded in the EMT.

$$a_t = \underset{a_i \in A_t}{\operatorname{argmin}} d(s_t^{a_i}, T) \tag{2}$$

When the robot moves towards S, T in the above formulas will be replaced by S.

Considering that this strategy may cause the robot to trap in local optimum near concave obstacles, we design a strategy of tracing the source based on the experience recorded in the IS and ISR. As shown in Fig. 3(a), if the robot repeatedly reaches one node, the parent node of this node in the moving direction will be queried. The process of tracing the source will continue until a node that has not been visited many times is found. Under the guidance of the nodes found in this process, the robot can be taken out of the area that may enable it to search meaninglessly. Therefore, this strategy can ensure the effectiveness of the robot action selection.

The pseudo code of the action selection strategy for the robot in the early stage of path planning is shown in Algorithm 1.

Algorithm 1. Candidate target point filtration based on the selective retention

while $s_t \neq T$ **do**
  if $s_t \in F$ **then**
    Tracing the source and get new s_t
  end if
  if $\in \leqslant Q$ **then**
    Select the action randomly in A_t to get a_t
  else
    for a_t^i in A_t **do**
      $d(s_t^{a_i}, T) = \sqrt{(x_{s_t^{a_i}} - x_T)^2 + (y_{s_t^{a_i}} - y_T)^2}$
    end for

(*continued on next page*)

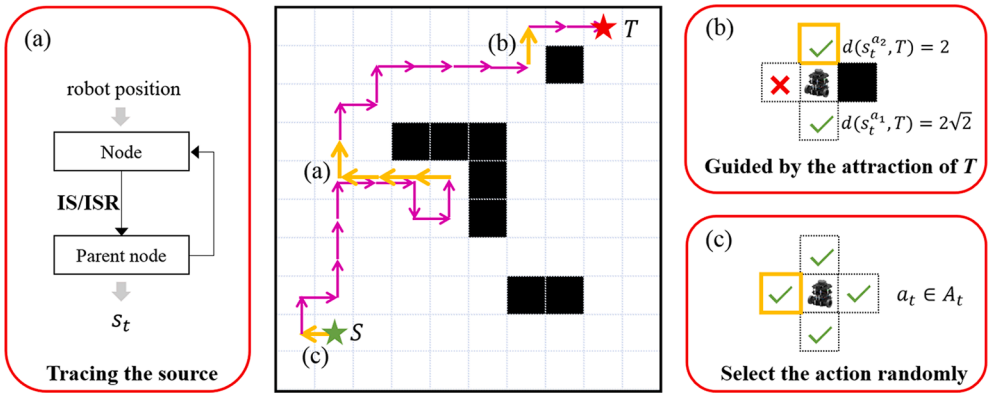

Fig. 3. The strategy of action selection for the robot in the early stage of path planning. (a) If a robot reaches a node frequently, it will be brought to a node where there are more choices by using the strategy of tracing the source. (b) The robot will select the action which makes it closer to T at times. (c) Sometimes, the action will be selected randomly by the robot.

(*continued*)

$a_t = \underset{a_i \in A_t}{\text{argmin}} d(s_t^{a_i}, T)$
end if
Update s_t, EM, IS
end while
while $s_t \neq S$ **do**
if $s_t \in F$ **then**
Tracing the source and get new s_t
end if
if $\in \leqslant Q$ **then**
Select the action randomly in A_t to get a_t
else
for a_t^i in A_t **do**
$d(s_t^{a_i}, S) = \sqrt{(x_{s_t^{a_i}} - x_S)^2 + (y_{s_t^{a_i}} - y_S)^2)}$
end for
$a_t = \underset{a_i \in A_t}{\text{argmin}} d(s_t^{a_i}, S)$
end if
Update s_t, EMR, ISR
end while

In order to search for the global shortest path as much as possible, the robot will have the possibility of selecting actions randomly in most algorithm. Therefore, the convergence rate of the experience table usually slows down as the size of environment increases, which affects the planning efficiency. To solve this problem, we propose to lock the search scope for subsequent episodes after the robot completes the first episode.

As shown in Fig. 4, since the robot movements are bidirectional, an area smaller than the size of the environment can be enclosed by robot historical trajectories in the first episode. Moving in this area enables the robot to reduce the searches for the areas that are not meaningful for planning the final path, thus the planning efficiency is improved. In addition, since the action selection strategy in the first episode is based on the attraction of T, the global shortest path is usually included in this enclosed area. Therefore, locking the search scope after the early stage will not affect the length of the planned path greatly in most cases.

3.3. The medium stage of path planning

Since the second episode, the robot has had some experience of the environment. In order to avoid searching blindly, we design an action selection strategy based on the experience guidance, where the planning efficiency can be improved by usingthe experience obtained in the obverse and reverse movements.

According to the definitions of the IS and ISR, if the robot is in the node h when it moves towards T, then the shortest path through the node h must pass through the node p_h^{IS}. Hence if the robot chooses to move to the p_h^{IS} when it reaches the node h on the reverse movement, the time it spends in reaching S will be reduced. The cross-use of experience is also effective in the obverse movement. Therefore, according to whether the parent node of s_t is recorded, the action selection strategy the robot uses in the medium stage of path planning can be described as two cases. Here, we take the obverse movement as example.

(1) If $p_{s_t}^{\text{ISR}}$ can be queried in the ISR:

$$a_t \leftrightarrow p_{s_t}^{\text{ISR}} \tag{3}$$

(2) If $p_{s_t}^{\text{ISR}}$ can not be queried in the ISR:

$$a_t = \underset{a_i \in A_t}{\text{argmin}}(l_{s_t^{a_i}}^{\text{EMR}}) \tag{4}$$

Here, if $s_t^{a_i}$ is not in the EMR, $l_{s_t^{a_i}}^{\text{EMR}}$ is equal to $-inf$.

When the robot moves towards S, $p_{s_t}^{\text{IS}}$ will be queried in the IS, and the $l_{s_t^{a_i}}^{\text{EMR}}$ is replaced by $l_{s_t^{a_i}}^{\text{EM}}$.

Apart from that, to make the length of the final planned path as short as possible, the robot will have a certain probability to select randomly in the feasible action set, and the actions that have not been selected will be preferred. Therefore, the strategy of action selection in the medium stage can be described as Fig. 5.

The pseudo code of the action selection strategy for the robot in the medium stage of path planning is shown in Algorithm 2.

Algorithm 2. The action selection strategy used in the medium stage

while $s_t \neq T$ **do**
if $\in \leqslant P$ **then**
if $A_t^{\text{u}} = \varnothing$ **then**
Select the action randomly in A_t to get a_t
else
Select the action randomly in A_t^{u} to get a_t
end if
else
if s_t in the ISR **then**
$a_t \leftrightarrow p_{s_t}^{\text{ISR}}$
else
$a_t = \underset{a_i \in A_t}{\text{argmin}} l_{s_t^{a_i}}^{\text{EMR}}$
end if

(*continued on next page*)

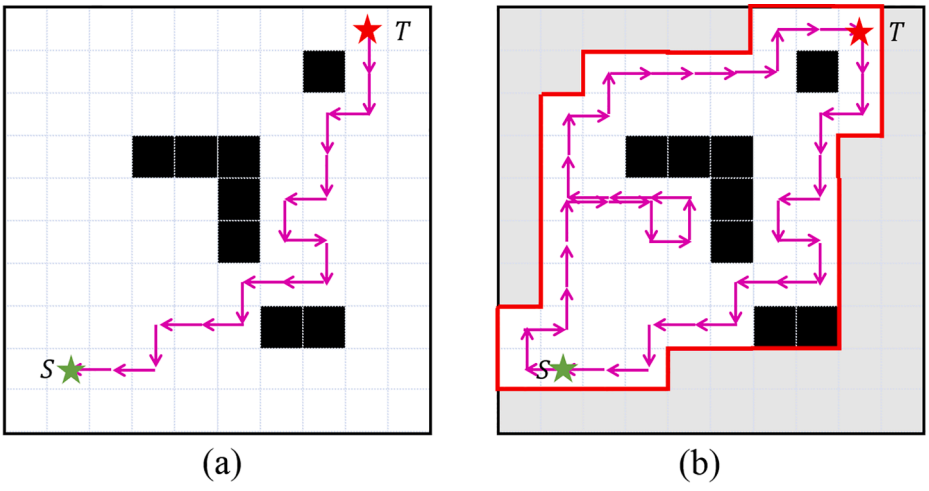

Fig. 4. The lock of the search scope after the early stage (the first episode). (a) The path of the robot moving from T to S (reverse movement). (b) The search scope of the robot in the subsequent episodes.

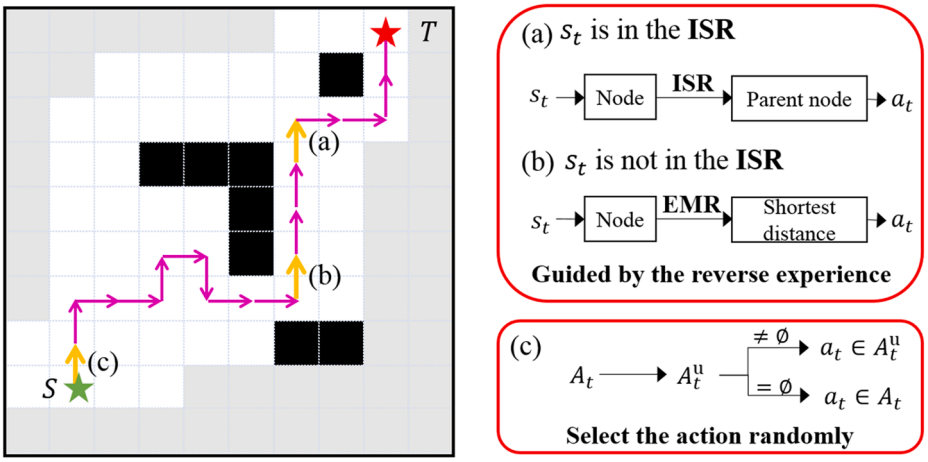

Fig. 5. The strategy of action selection when the robot moves from S to T in the medium stage of path planning. (a) If s_t is in the ISR, the robot will move to $p_{s_t}^{\mathrm{ISR}}$. (b) If s_t is not in the ISR, the robot will choose the node that can make it reach T faster. (c) Sometimes, the robot will select the action randomly.

(*continued*)

end if
Update s_t, EM, IS
end while
while $s_t \neq S$ **do**
if $\epsilon \leqslant P$ **then**
if $A_t^u = \varnothing$ **then**
Select the action randomly in A_t to get a_t
else
Select the action randomly in A_t^u to get a_t
end if
else
if s_t in the IS **then**
$a_t \leftrightarrow p_{s_t}^{\mathrm{IS}}$
else
$a_t = \underset{a_t \in A_t}{\operatorname{argmin}} l_{s_t}^{\mathrm{EM}}$
end if
end if
Update s_t, EMR, ISR
end while

3.4. The end stage of path planning

In the BALA, the relationships between nodes have been recorded in the IS and ISR, thus the robot can plan the final path by querying the experience tables. However, since the parent-child relationship between nodes recorded in the IS and ISR is based on the shortest distance to S and T respectively, this relationship is not necessarily mutual in two directions. Therefore, in order to make full use of the experience and make the planned path as short as possible, we restrict the nodes recorded in the EMT. Only the nodes that have the strong connectivity with their parent nodes will be recorded in the EMT.

The strong connectivity is defined as the mutual parent-child relationship between two nodes in the obverse and reverse movements. For example, if node g and node h have strong connectivity relationship, then node g must be the parent node of node h in the IS and the parent node of node g is node h in the ISR. This strong connectivity shows that the robot at node h can get to S faster by passing through node g. Moreover, in the obverse movement, the shortest path from node g to T must pass through node h. Therefore, the nodes on the final planned path must have the characteristics of strong connectivity.

If the nodes corresponding to the shortest path length recorded in the EMT include S and T, the robot will have two ways to plan the path. One way is that the robot plans from T, and queries the parent node of the current node in the IS one by one, all the way to S. In another way, the robot uses the experience recorded in the ISR to plan the path from S.

Sometimes the distance from multiple nodes to a node is the same, but each node can only have one parent node, thus some of the nodes on the shortest path may be not recorded in the EMT. To ensure that the path planning can continue in this case, we propose to build the bridge path.

Fig. 6 shows an example that the robot plans the path from S. It can be seen that the robot can reach node 43 under the guidance of the ISR. However, node 43 is not recorded in the EMT because the parent-child relationship between it and node 45 is not mutual. To test whether node 43 is on the shortest path, the distance from this node to S and T will be computed. The sum is equal to the shortest path length recorded in the EMT, hence node 43 is on the shortest path and the planning process can continue after building a bridge path. This node is not recorded in the EMT because node 43 has only one parent node based on the definition of the IS. However, if the sum is not equal to the shortest path length, the planning of the path will be tried again from T according to the experience recorded in the IS.

The pseudo code of planning the path in the end stage is shown in Algorithm 3. To describe clearly, we only show the case of planning from S, and planning from T can be completed by querying the IS. In addition, if both the obverse and reverse planning fail, the robot will start a new episode to accumulate experience.

Algorithm 3. The action selection strategy used in the medium stage

$s_t = S$
while $s_t \neq T$ **do**
if s_t is in the EMT **then**
$a_t \leftrightarrow p_{s_t}^{\mathrm{ISR}}$
else
if $l_{s_t}^{\mathrm{EM}} + l_{s_t}^{\mathrm{EMR}} = l_{c_{s_t}^{\mathrm{ISR}}}^{\mathrm{EMT}}$ **then**
$a_t \leftrightarrow p_{s_t}^{\mathrm{ISR}}$
else
break
end if
end if
Update s_t
end while

4. Experiments and analysis

The performance of the proposed algorithm is evaluated in three types of experiments in this section. First, the planning process was simulated on the road network map. Then, we did the simulation experiments on the grid map and changed the experimental conditions from many aspects to test the stability of the BALA. In the above simulation experiments, the proposed algorithm is compared with the

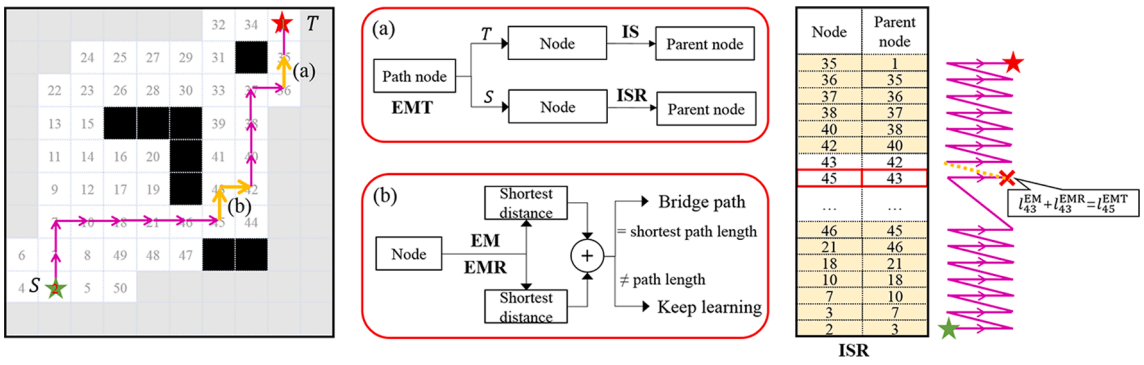

Fig. 6. An example of the end stage of path planning. (a) The nodes that have strong connectivity with their parent nodes will be recorded in the EMT. (b) If the sum of the shortest distances from a node to *S* and *T* is equal to the shortest path length recorded in the EMT, this node will be used to build bridge path although it is not in the EMT.

Classical Q-Learning (CQL) algorithm (Watkins & Dayan, 1992) and the Experience-Memory Q-Learning (EMQL) algorithm (Zhao et al., 2020). The experiment results are given from four aspects: episode times, planning time, path length and the number of times to get the global shortest path in 50 experiments (NT50). The planning efficiency is reflected by the episode times and planning time required to plan the final path. Moreover, due to the randomness of the robot action selection in each step, all the experiment results shown in this paper are the average values of 50 experiments. Finally, the BALA was used for the path planning task of Turtlebot3 burger robot in the robot simulation platform and real-world environment to further verify the practicability of the proposed algorithm.

4.1. Path planning based on the road network map

The road network map used in this experiment is derived from Beijing road network GIS data set (Song, Gao, & Shan, 2017), which is shown in Fig. 7. In this map, the robot can reach all the nodes that are connected with its current position.

The experiment results in episode times, planning time, path length and NT50 are shown in Table 4. It can be seen from the comparison results that based on the BALA, the robot needs fewer episodes, shorter planning time to plan a short path. In addition, the proposed algorithm can plan the global shortest path 38 times in 50 experiments, while the shortest path cannot be obtained based on the other two algorithms. Therefore, in this topological environment, improving the convergence rate does not weaken the optimization ability of the BALA.

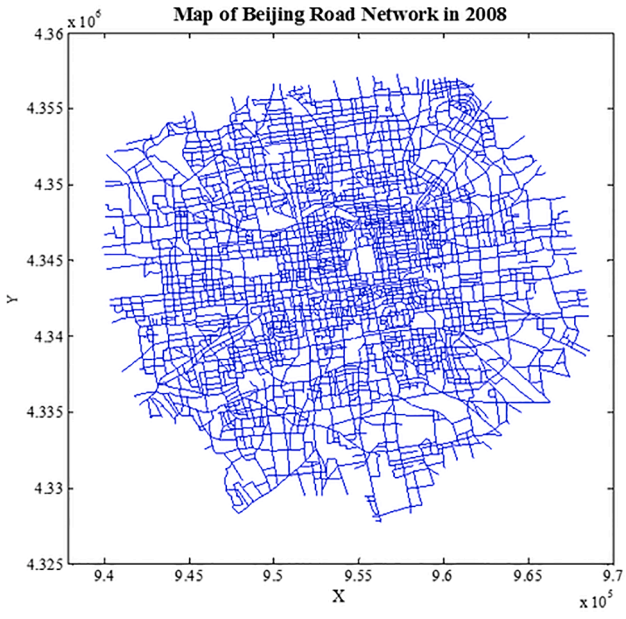

Fig. 7. The map of Beijing road network in 2008, where not all the intersecting points in this map are connected since this is a 2D map.

Table 4
The experiment result on the road network map.

	Episode times	Planning time/s	Path length/m	NT50
CQL	18422.82	706.23	36981.10	0
EMQL	4408.94	516.49	33363.94	0
BALA	**3887.98**	**324.58**	**32250.65**	**38**

4.2. Path planning based on the grid map

To further verify that the performance of the BALA is stable and desirable in planning efficiency, we did three sets of experiments on the grid map. First, we tested the effectiveness of the BALA on the grid maps with different characteristics. Second, to illustrate that the lock of search scope is essential to reduce the planning time, the path planning experiments were carried out on the grid maps of different sizes. Finally, we tested the effect of parameter changes on the performance of the algorithm.

4.2.1. Path planning based on different types of maps

In this part, we adopt the Real-World Benchmarks as the grid maps (Sturtevant, 2012). These maps are shown in Fig. 8, the abundant features of which can also test the stability of the algorithm to a certain extent.

As the experiment results shown in Fig. 9, Fig. 10 and Fig. 11, the superiority of the BALA in planning efficiency does not change with the change of map types. In addition, it can be seen from Fig. 9 that compared with the CQL and EMQL, planning by the BALA can save about 98% and 75% episode times respectively, which can save a lot of energy of the robot in practical application. To evaluate the planning efficiency of the robot more effectively, the planning time and path length are marked according to the amount of change, the comparison results with the CQL and EMQL are shown in Fig. 12 and Fig. 13 respectively. Apparently, there is only little loss on the path length when the planning time is reduced greatly. This conclusion can also be drawn from the data shown in Table 5 and Table 6, where the comparison results are presented as percentages.

In addition, since we pursue high efficiency in the planning process, the number of times to get the global shortest path is less than that of the CQL and EMQL in some maps (Table 7). Nevertheless, the BALA has the possibility of planning the global shortest path in all maps, while the CQL and EMQL cannot plan the global shortest path in the London gird

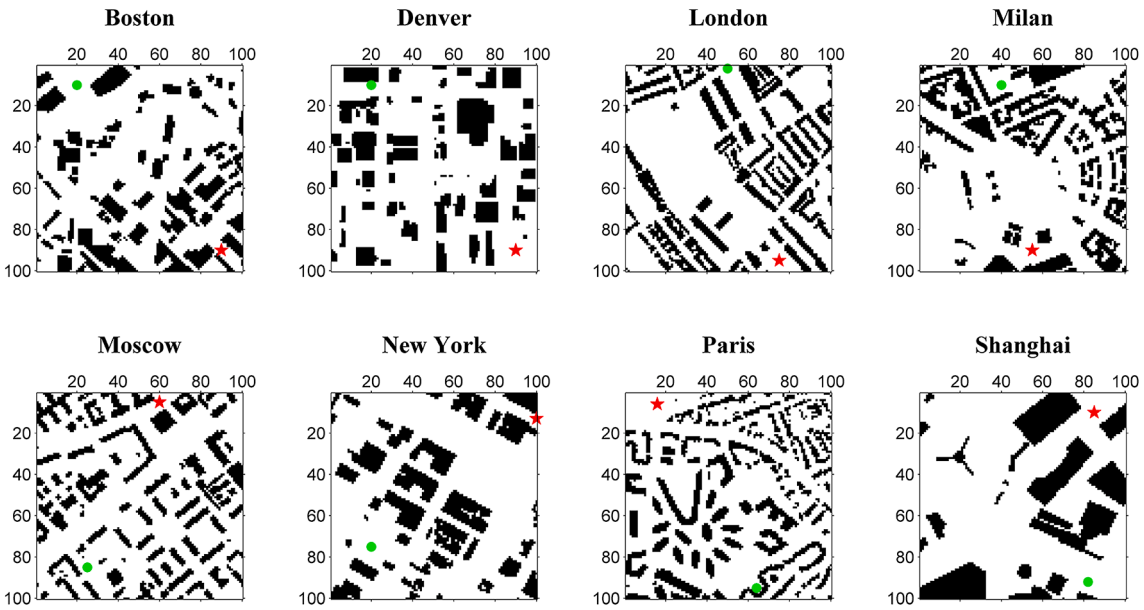

Fig. 8. The grid maps of different cities, where the green circle represents *S*, and the red pentagram represents *T*.

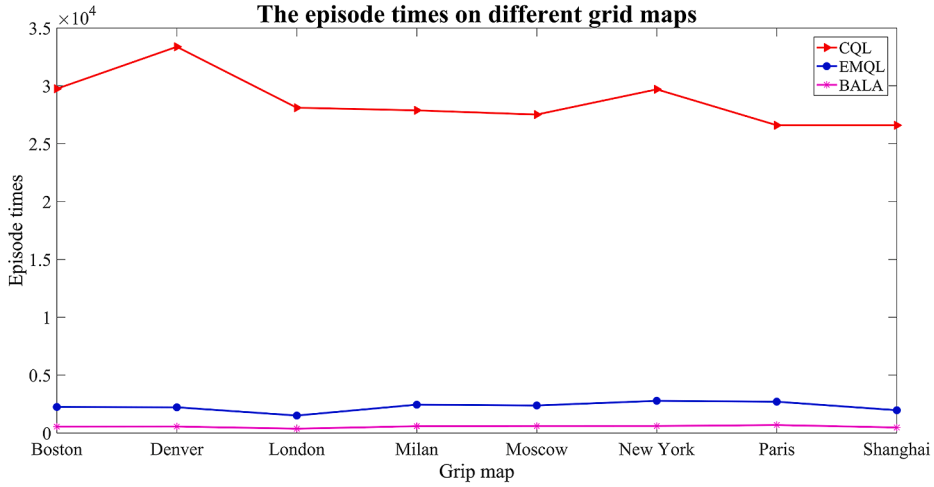

Fig. 9. The episode times in each algorithm on different grid maps.

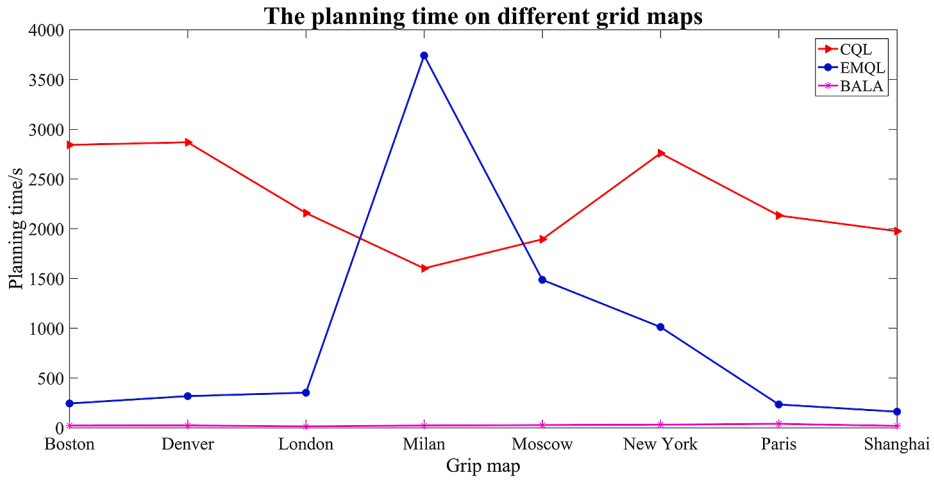

Fig. 10. The planning time in each algorithm on different grid maps.

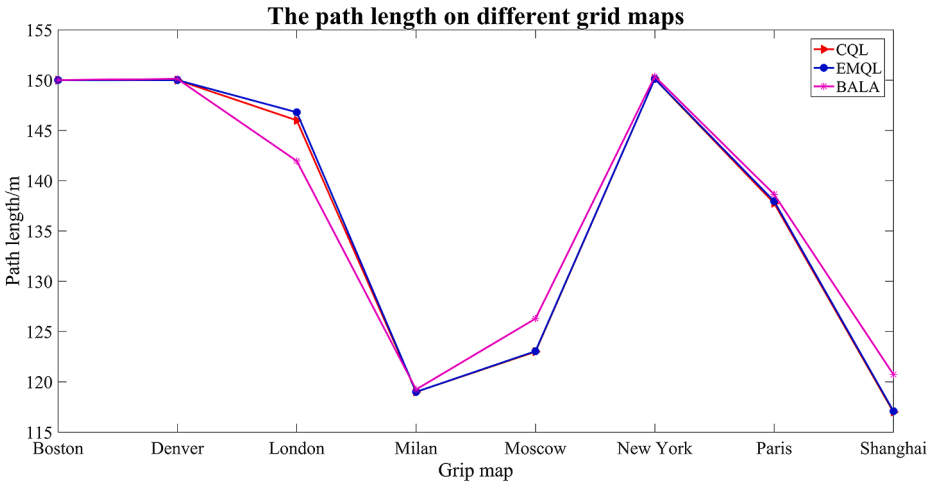

Fig. 11. The path length in each algorithm on different grid maps.

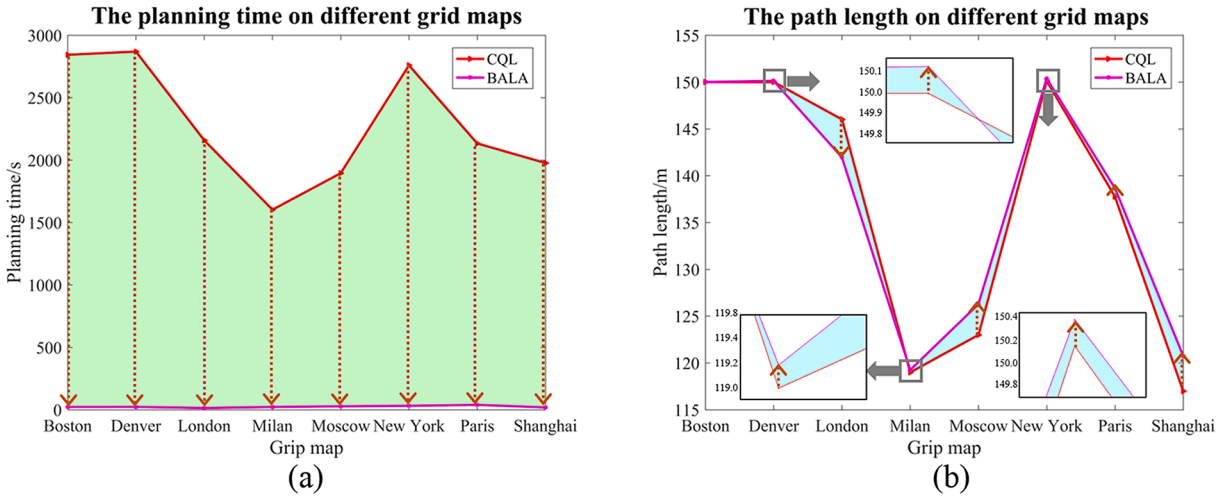

Fig. 12. The comparison results with the CQL, the performance improvement of the BALA over CQL is marked in brown arrows. (a) Planning time. (b) Path length.

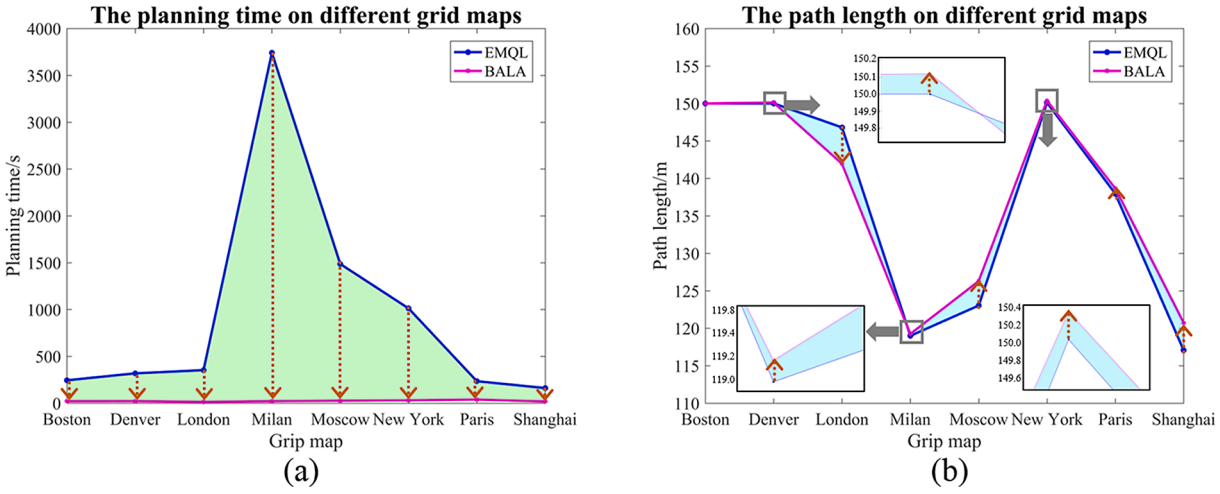

Fig. 13. The comparison results with the EMQL, the performance improvement of the BALA over EMQL is marked in brown arrows. (a) Planning time. (b) Path length.

Table 5
The comparison results with the CQL in planning time and path length.

	Planning time/s		Comparison	Path length/m		Comparison
	CQL	BALA		CQL	BALA	
Boston	2842.74	22.59	↓ **99.21**%	150.00	150.00	0.00%
Denver	2868.40	23.41	↓ **99.18**%	150.00	150.12	↑ **0.08**%
London	2158.10	14.78	↓ **99.32**%	146.00	141.96	↓ **2.77**%
Milan	1602.06	22.91	↓ **98.57**%	119.00	119.24	↑ **0.20**%
Moscow	1894.64	27.97	↓ **98.52**%	123.00	126.28	↑ **2.67**%
New York	2758.49	31.99	↓ **98.84**%	150.12	150.36	↑ **0.16**%
Paris	2132.91	39.95	↓ **98.13**%	137.76	138.64	↑ **0.64**%
Shanghai	1975.95	19.21	↓ **99.03**%	117.00	120.72	↑ **3.18**%

Table 6
The comparison results with the EMQL in planning time and path length.

	Planning time/s		Comparison	Path length/m		Comparison
	EMQL	BALA		EMQL	BALA	
Boston	244.52	22.59	↓ **90.76**%	150.00	150.00	0.00%
Denver	318.59	23.41	↓ **92.65**%	150.00	150.12	↑ **0.08**%
London	353.10	14.78	↓ **95.81**%	146.80	141.96	↓ **3.30**%
Milan	3740.72	22.91	↓ **99.39**%	119.00	119.24	↑ **0.20**%
Moscow	1486.57	27.97	↓ **98.12**%	123.04	126.28	↑ **2.63**%
New York	1012.35	31.99	↓ **96.84**%	150.12	150.36	↑ **0.16**%
Paris	234.64	39.95	↓ **82.97**%	137.96	138.64	↑ **0.49**%
Shanghai	161.71	19.21	↓ **88.12**%	117.08	120.72	↑ **3.11**%

Table 7
The number of times to get the global shortest path in 50 experiments.

	CQL	EMQL	BALA
Boston	50	50	50
Denver	50	50	47
London	**0**	**0**	12
Milan	50	50	48
Moscow	50	49	15
New York	49	47	43
Paris	31	26	27
Shanghai	50	48	16

map. Overall, the performance of the BALA is more stable.

4.2.2. Path planning based on different sizes of maps

Since the convergence rate of the experience table is usually positively related to the size of environment, the larger the size of environment is, the more time the robot usually needs to complete the path planning task. However, unlike other algorithms, the bidirectional movement enables the robot to obtain the search scope after the early stage of path planning in the BALA. Therefore, to verify that the lock of search scope can enable the robot to reduce the time spent on the meaningless searches, we did experiments on the different sizes of maps. As shown in Fig. 14, we scaled the size of Shanghai grid map to 20*20 (0.2), 40*40 (0.4), 60*60 (0.6), 80*80 (0.8) and 100*100 (1.0). Here, to ensure the fairness of the experiment, the positions of S and T were also changed accordingly.

We calculated the mean value and the variance of planning time in 50 experiments on different size of Shanghai grid map. As shown in Fig. 15(a), as the size of environment increases, the planning time used in the CQL and EMQL increases greatly, while this change has little effect on the planning time in the BALA. Moreover, as the size of environment increases, the gaps in planning time between the BALA and the other two algorithms increase gradually. In addition, it can be seen from Fig. 15(b) that the fluctuation degree of the planning time used in the CQL and EMQL in each experiment also increases with the environment area. Therefore, the performance of the BALA in planning efficiency is more stable in different sizes of environments than that of the CQL and EMQL.

4.2.3. Path planning with different parameters

In most optimization algorithms, the setting of parameters will affect the performance of the algorithm. Compared with the CQL and EMQL, there is no need to design reward function in the BALA, so the proposed algorithm has better stability in theory. However, like the learning rate (α) and discount rate (γ) used in reinforcement learning, there are two parameters (P, Q) used in the action selection strategy of the BALA. To test the influence degree of these two parameters on the performance of the BALA, we did a comparative experiment on Shanghai grid map based on five groups of parameters (Table 8).

It can be seen from the experiment results shown in Fig. 16 and Fig. 17, the changes of parameters will seriously affect the performance of the CQL in the episode times and planning time. Because of the large scale of coordinates, it seems that the performance of the EMQL and BALA is stable under different parameters. To get a clear conclusion, we calculated the variance of each evaluation indicator for different algorithms under 5 groups of parameters. As the statistical data shown in Table 9, the BALA has the least fluctuation in terms of episode times and planning time under different parameters. The performance improvements in these two aspects make the BALA less stable than the CQL and EMQL in path length (Fig. 18), but it can be seen from Table 9 that this performance penalty is minor.

4.3. Running on Turtlebot3 burger robot

The performance of the algorithm in real environment is affected by many factors. Therefore, to test the effectiveness of the proposed algorithm, the BALA is used in the path planning task of Turtlebot3 burger robot (Fig. 19) in unknown environment.

4.3.1. Moving in the ideal environment

Gazebo is a 3D robot simulation platform, which supports the simulation of experimental environment, robot structure, sensor data

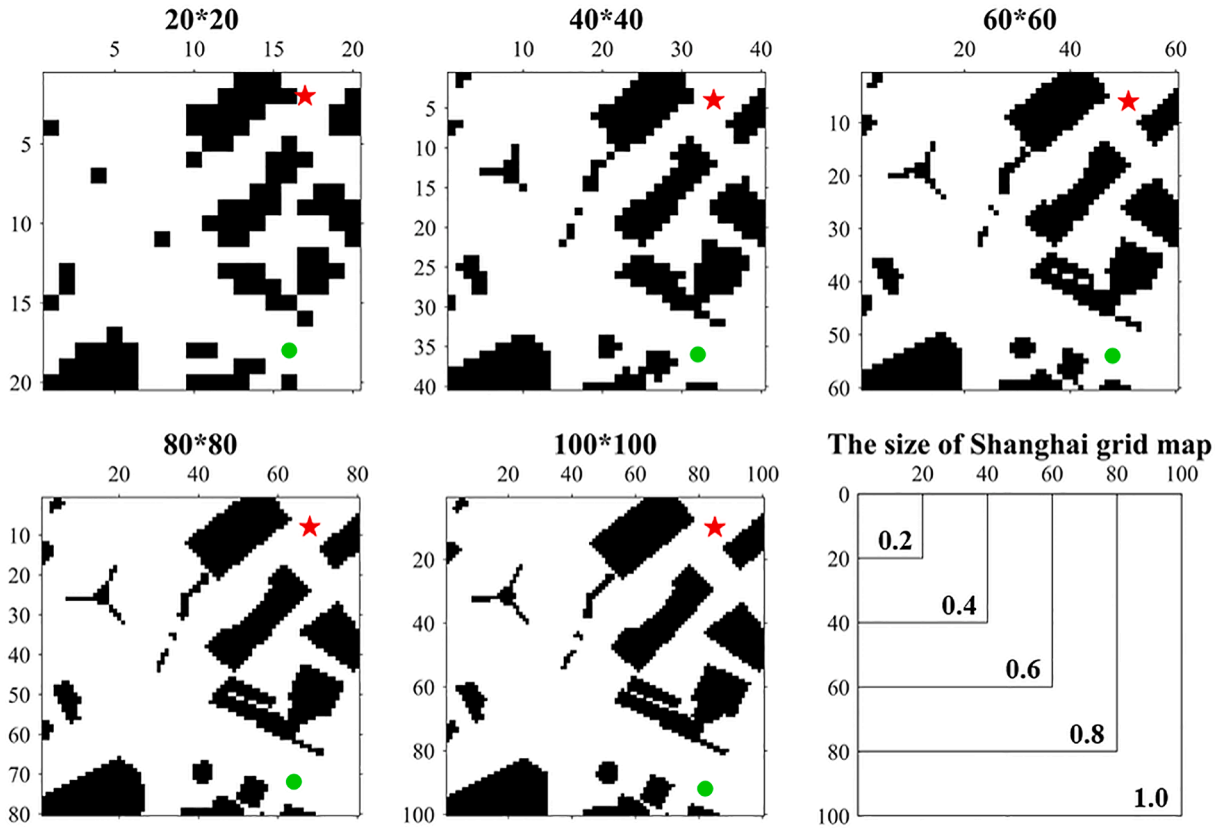

Fig. 14. Different size of Shanghai grid map.

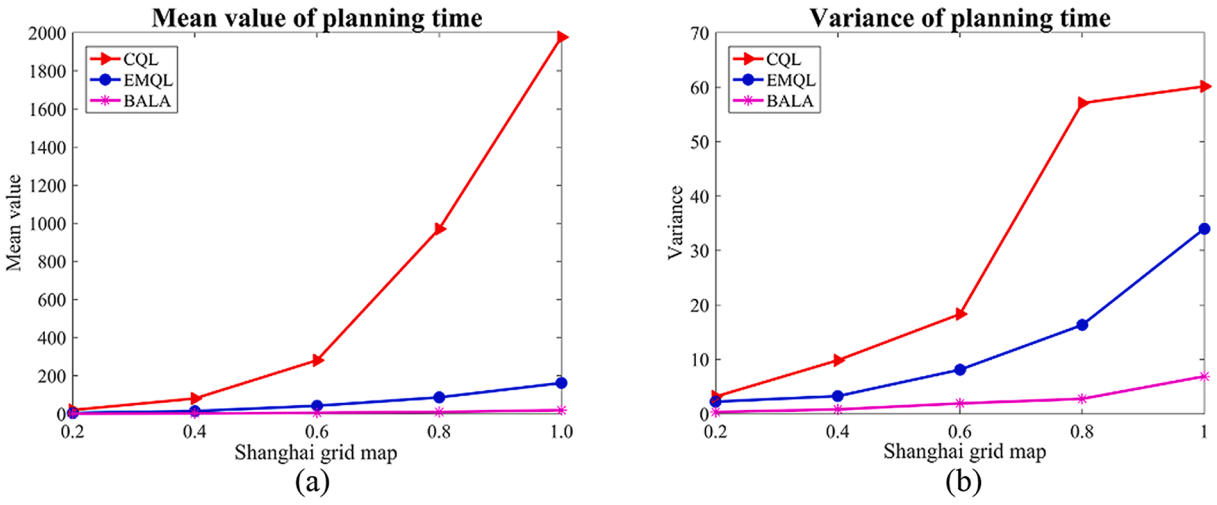

Fig. 15. (a) The mean value of planning time on different sizes of Shanghai grid map. (b) The variance of planning time on different sizes of Shanghai grid map.

and communication network. Experiments on this platform can eliminate the influence of ground frication and sensor noise on the performance of the planning algorithm. Therefore, before running the robot in real environment, we carried out an experiment in an ideal environment of 3 m*3 m in Gazebo.

As shown in Fig. 20, Turtlebot3 burger robot is required to plan a short path from (0,0) to (2.4,2.4) in unknown environment. Some of the snapshots in the planning process of the robot are shown in Fig. 21. It can be seen from Fig. 21(a) that an effective search scope for later episodes is provided in the early stage of path planning. In addition, although the path length of the robot is long in the medium stage of path planning (Fig. 21(b)), a global shortest path can be planned in the end stage based on the BALA (Fig. 21(c)).

4.3.2. *Moving in the real environment*

The experimental environment and the path planning task in the real environment are exactly the same as that in Gazebo. However, we found that there is a deviation between the actual trajectory of the robot and the ideal trajectory provided by the BALA in the real-world experiment, which will cause the robot to collide with obstacles. As shown in Fig. 22, when the robot moves along the path (green line, ideal trajectory) planned by the proposed algorithm, its actual trajectory (purple line) does not coincide with the ideal trajectory. According to the measurement result, the position of the robot has deviated from the ideal

Table 8
The groups of parameters.

	1	2	3	4	5
CQL	$\gamma = 0.95$ $\alpha = 0.30$	$\gamma = 0.90$ $\alpha = 0.10$	$\gamma = 0.80$ $\alpha = 0.20$	$\gamma = 0.70$ $\alpha = 0.30$	$\gamma = 0.85$ $\alpha = 0.25$
EMQL	$\gamma = 0.95$ $\alpha = 0.30$	$\gamma = 0.90$ $\alpha = 0.10$	$\gamma = 0.80$ $\alpha = 0.20$	$\gamma = 0.70$ $\alpha = 0.30$	$\gamma = 0.85$ $\alpha = 0.25$
BALA ($\tau =$ −0.00001*episode)	$P = 0.50$ $Q =$ 0.30-τ	$P = 0.70$ $Q =$ 0.20-τ	$P = 0.60$ $Q =$ 0.40-τ	$P = 0.80$ $Q =$ 0.40-τ	$P = 0.70$ $Q =$ 0.30-τ

trajectory by 22 cm after moving about 12 m. Nevertheless, it is found by checking the position information provided by the odometer and IMU that the robot does not realize that it has deviated from the ideal trajectory in the moving process. Compared with the ideal environment in Gazebo, the ground is not even and the noise of the sensor is greater. If these factors make the positioning error of the robot accumulate gradually, then the robot will inevitably collide with obstacles when it moves along the planned path for a long time. Therefore, we did an experiment to analyze the effect of robot positioning error on the path planning in the experimental environment.

As shown in Fig. 23, the robot should pass through Point 1, Point 2, Point 3 and Point 4 in turn and periodically. The position information

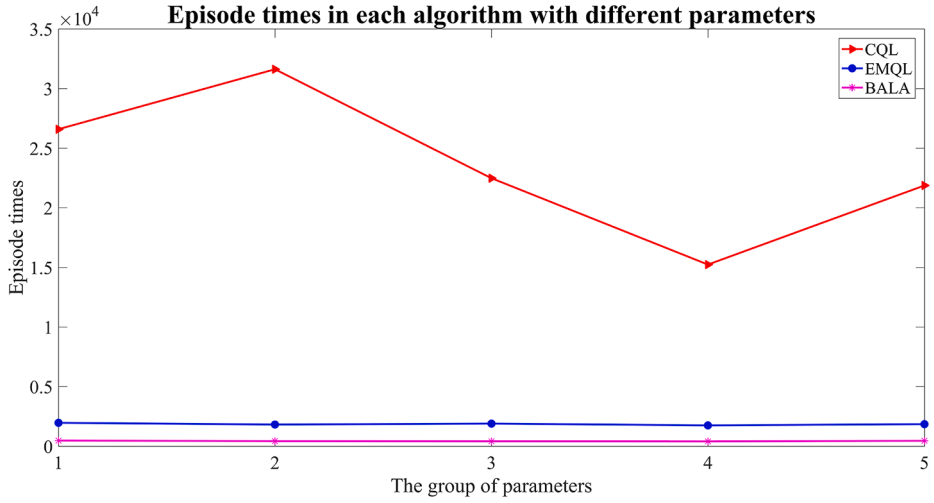

Fig. 16. The episode times in each algorithm with different parameters.

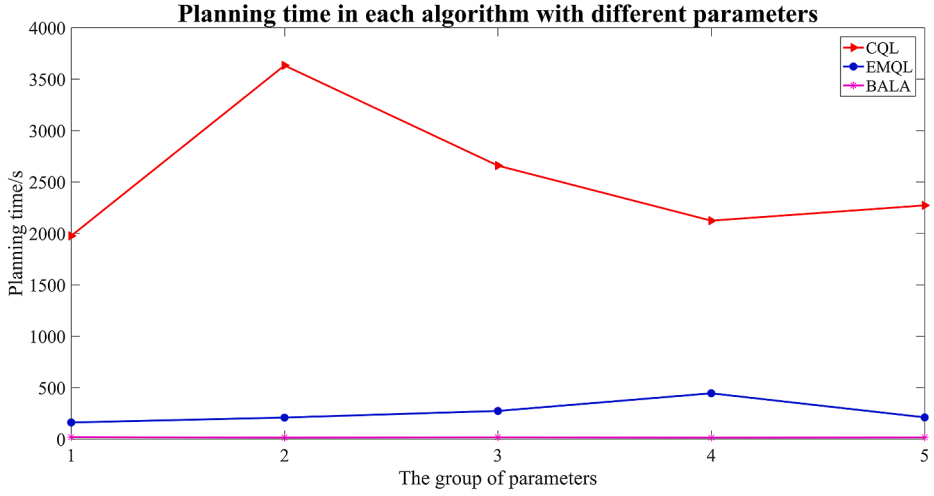

Fig. 17. The planning time in each algorithm with different parameters.

Table 9
The experiment results in each algorithm with different parameters.

	Episode times			Planning time/s			Path length/m		
	CQL	EMQL	BALA	CQL	EMQL	BALA	CQL	EMQL	BALA
1	26592.40	1964.60	473.90	1975.95	161.71	19.21	117.00	117.08	120.72
2	31618.80	1819.56	428.94	3632.88	209.72	15.27	117.08	117.04	119.52
3	22481.98	1897.70	416.57	2658.89	273.99	17.83	117.00	117.00	120.12
4	15237.04	1749.72	407.20	2123.86	445.54	14.71	117.00	117.00	118.80
5	21869.22	1852.16	455.42	2272.28	211.66	17.05	117.00	117.04	120.48
Variance	5430.14	72.34	**24.78**	595.19	99.14	**1.65**	0.032	**0.029**	0.694

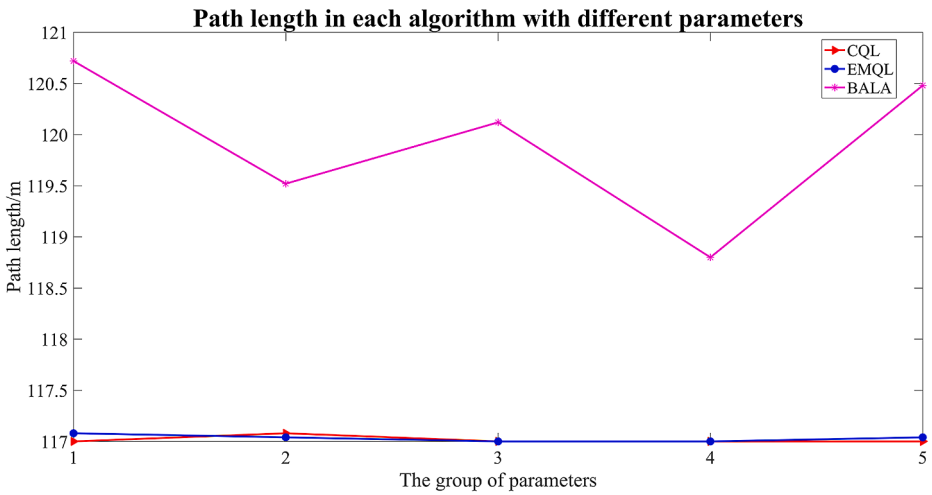

Fig. 18. The path length in each algorithm with different parameters.

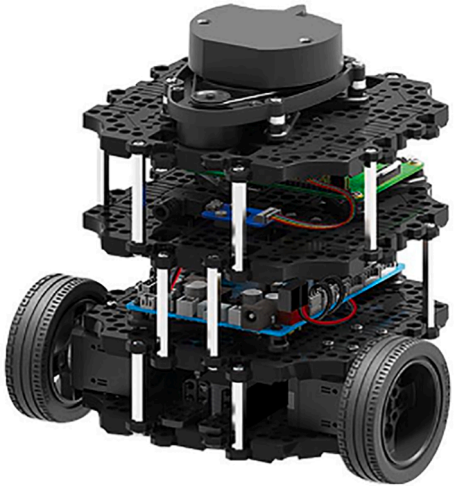

Fig. 19. Turtlebot3 burger robot.

the robot obtains from odometer and IMU in these points was compared with actual measured value, and the difference is shown in Fig. 24. Apparently, with the increase of robot passing times, the positioning error of robot at each point increases. Therefore, it can be deduced that the positioning error of robot will increase with the increase of moving distance.

Through the statistical analysis of many experiments, it can be concluded that in the environment shown in Fig. 23, there will be a positioning error of 1 cm~2 cm for each meter of the robot movement, and the error is cumulate. Therefore, when the robot needs to moves for a long distance to accumulate experience in unknown environment, the accumulated positioning error will make the robot gradually deviate from the ideal trajectory and eventually collide with obstacles.

4.4. Summary

The above experiment results show that the action selection strategy in BALA is more effective than the action selection strategy based on the Q table in the CQL and EMQL. Since there is no obstacle to provide penalty signal in the road network map, the performance of the BALA is the best in both path optimization ability and planning efficiency. In addition, although the BALA cannot plan the global shortest path every time in some environments, it has the ability to find the global optimum

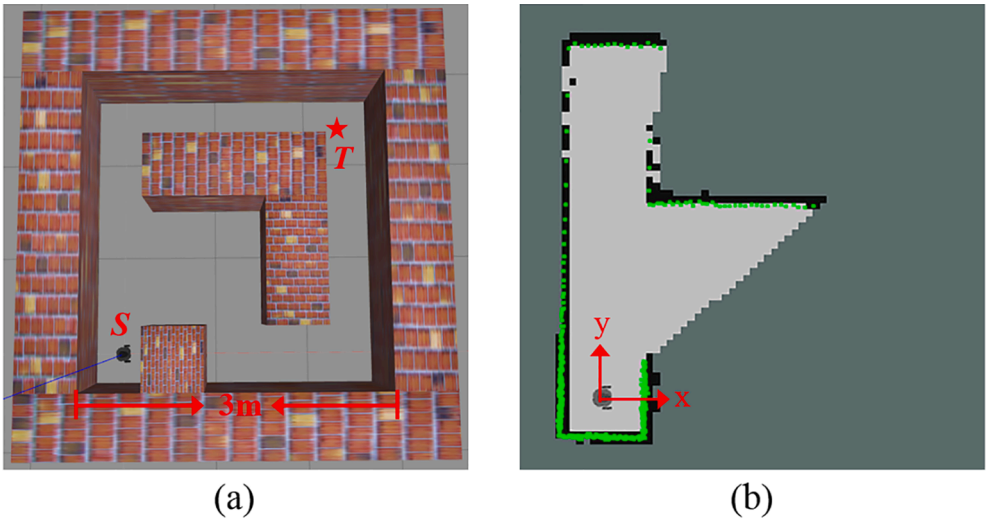

Fig. 20. (a) The environment in which the robot performs path planning task in Gazebo. (b) The environment information the robot obtains from laser radar at the start point.

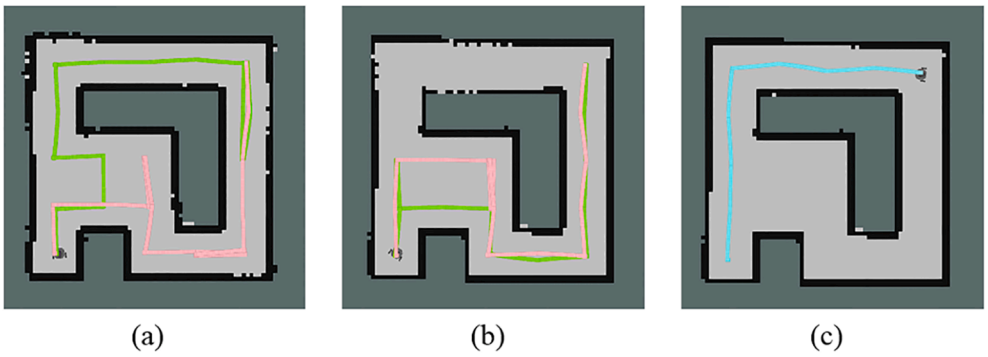

Fig. 21. The trajectories of the robot in the planning process. The trajectories of the robot in the obverse and reverse movements are marked pink and green respectively, and the blue trajectory is the final planned path. (a) The early stage. (b) The medium stage. (c) The end stage.

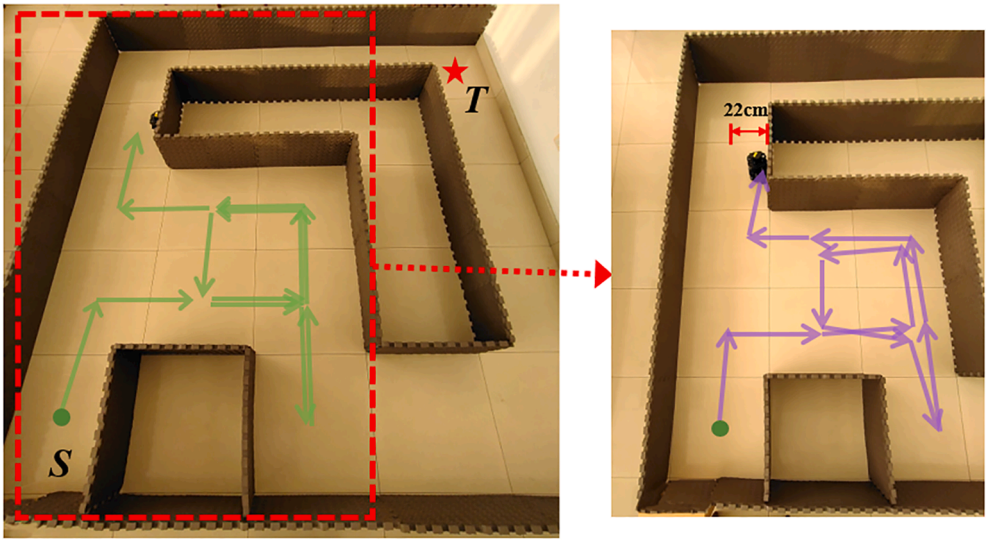

Fig. 22. Turtlebot3 burger robot performs path planning task in real environment,where the area of a brick is 60 cm*60 cm (In order to better present the experiment result, not all the paths are shown in the figure.). The green line presents the path planned by the BALA for the robot, where the robot needs to move 60 cm in one direction (front, back, left, right) in each step. The purple line is the actual trajectory of the robot.

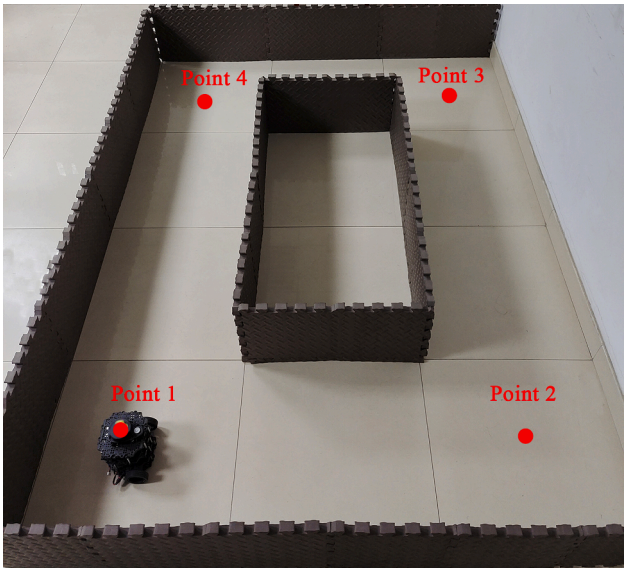

Fig. 23. The environment for testing the positioning accuracy of the robot.

in any environment. Therefore, the proposed algorithm has the ability to balance planning efficiency and path length, and it has better stability than the other two algorithms. Although due to the insufficient positioning accuracy caused by the external factors, the robot cannot move along the planned path in the real environment, the successful planning in Gazebo shows that the BALA is effective in practical application.

5. Conclusion

For the requirement of fast path planning in the environment exploration, search and rescue and other tasks, this paper focuses on improving the planning efficiency of the robot in unknown environment while ensuring the short length of the path. Considering the continuity of the robot position in practical application, we define the episode as a bidirectional movement between the start point and the target point, and propose a path planning algorithm based on the bidirectional associative learning. This method consists of three stages. In the early stage without any experience, the attraction of the target point is used as the guidance for the robot to select action. Moreover, the historical trajectories in this stage will provide the search scope for the robot to reduce the unnecessary searches in subsequent episodes. In the medium stage, we propose an action selection strategy based on the experience guidance. In this strategy, the experience obtained in the obverse and reverse movements is used alternately by the robot to speed up the

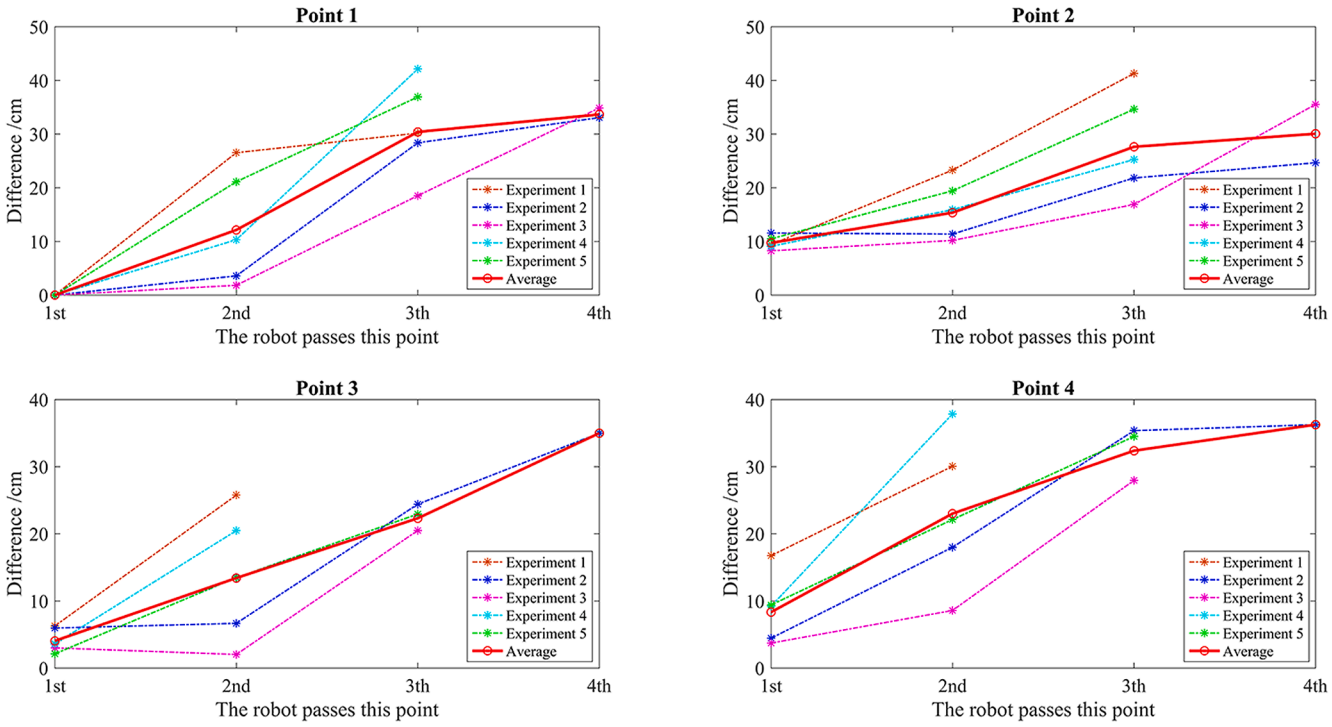

Fig. 24. The positioning error of the robot at four special positions in five experiments. The red line in each graph is the average value of five experiments.

learning process. In the end stage, a shorter path can be planned according to the experience recorded in five experience tables.

Compared with the classical Q-Learning algorithm and the algorithm which only uses the shortest distance memorized in the obverse movement, the BALA has fewer parameters to set and performs more stably. The experiment results show that the BALA can greatly reduce the planning time while having little or no influence on the path length. The well balance of planning time and path length makes the proposed algorithm more suitable for practical application. In addition, the bidirectional movement also provides the possibility to expand the application fields of the robot.

Our future work will focus on improving the practicality of the BALA from the following two aspects. First, we will further explore the experience that the robot can use to improve the planning efficiency in unknown environment. Second, it is also one of our future research directions to improve the positioning ability of the robot by adopting information fusion method.

CRediT authorship contribution statement

Meng Zhao: Methodology, Software, Writing - original draft. **Hui Lu:** Data curation, Writing - review & editing. **Siyi Yang:** Visualization, Investigation. **Yinan Guo:** Writing - review & editing. **Fengjuan Guo:** Supervision.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant No. 61827901, Shaanxi Key Laboratory of Integrated and Intelligent Navigation under Grant No. SKLIIN-20190201 and the National Natural Science Foundation of China under Grant No. 61973305.

References

Cheng, Shi, Ma, Lianbo, Lu, Hui, Lei, Xiujuan, & Shi, Yuhui (2021). Evolutionary computation for solving search-based data analytics problems. *Artificial Intelligence Review*, 1321–1348.

Dadgar, M., Jafari, S., & Hamzeh, A. (2016). A pso-based multi-robot cooperation method for target searching in unknown environments. *Neurocomputing, 177*, 62–74. https://doi.org/10.1016/j.neucom.2015.11.007

Edvardsen, V. (2019). Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network. *Natural Computing, 18*, 13–27. https://doi.org/10.1007/s11047-016-9575-0

Guo, H., Mao, Z., Ding, W., & Liu, P. (2019). Optimal search path planning for unmanned surface vehicle based on an improved genetic algorithm. *Computers & Electrical Engineering, 79*, 106467. https://doi.org/10.1016/j.compeleceng.2019.106467

Han, J. (2019). A surrounding point set approach for path planning in unknown environments. *Computers & Industrial Engineering, 133*(Jul.), 121–130. https://doi.org/10.1016/j.cie.2019.05.013

Hassanzadeh, I., & Sadigh, S. M. (2009). Path planning for a mobile robot using fuzzy logic controller tuned by ga. In *International Symposium on Mechatronics & Its Applications* (pp. 1–5). https://doi.org/10.1109/ISMA.2009.5164798

Konar, A., Goswami Chakraborty, I., Singh, S. J., Jain, L. C., & Nagar, A. K. (2013). A deterministic improved q-learning for path planning of a mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43*(5), 1141–1153. https://doi.org/10.1109/TSMCA.2012.2227719

Lebedev, D. V., Steil, J. J., & Ritter, H. J. (2005). The dynamic wave expansion neural network model for robot motion planning in time-varying environments. *Neural Networks, 18*(3), 267–285. https://doi.org/10.1016/j.neunet.2005.01.004

Liang, J.-H., & Lee, C.-H. (2015). Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm. *Advances in Engineering Software, 79*, 47–56. https://doi.org/10.1016/j.advengsoft.2014.09.006

Li, W., Chen, D., & Le, J. (2018). Robot patrol path planning based on combined deep reinforcement learning. In *2018 IEEE International Conference on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)* (pp. 659–666). https://doi.org/10.1109/BDCloud.2018.00101

Li, R., Fu, L., Wang, L., & Hu, X. (2019). Improved q-learning based route planning method for uavs in unknown environment. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)* (pp. 118–123).

Li, T., Li, Q., Li, W., Xia, J., Tang, W., & Wang, W. (2019). A path planning algorithm for space manipulator based on q-learning. In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* (pp. 1566–1571). https://doi.org/10.1109/ITAIC.2019.8785427

Li, Y., Meng, Q. H., Li, S., & Chen, W. (2008). A quadtree based neural network approach to real-time path planning. In *IEEE International Conference on Robotics & Biomimetics*. https://doi.org/10.1109/ROBIO.2007.4522360

Li, Q.-L., Song, Y., & Hou, Z.-G. (2015). Neural network based fastslam for autonomous robots in unknown environments. *Neurocomputing, 165*, 99–110. https://doi.org/10.1016/j.neucom.2014.06.095

Li, S., Xu, X., & Zuo, L. (2015). Dynamic path planning of a mobile robot with improved q-learning algorithm. In *2015 IEEE International Conference on Information and Automation* (pp. 409–414). https://doi.org/10.1109/ICInfA.2015.7279322

Low, E. S., Ong, P., & Cheah, K. C. (2019). Solving the optimal path planning of a mobile robot using improved q-learning. *Robotics and Autonomous Systems, 115*, 143–161. https://doi.org/10.1016/j.robot.2019.02.013

Lu, Hui, Zhang, Mengmeng, Fei, Zongming, & Mao, Kefei (2015). Multi-Objective Energy Consumption Scheduling in Smart Grid Based on Tchebycheff Decomposition. *IEEE Transactions on Smart Grid, 6*(6), 2869–2883.

Luo, H., Gao, Y., Wu, Y., Liao, C., Yang, X., & Cheng, K. (2019). Real-time dense monocular slam with online adapted depth prediction network. *IEEE Transactions on Multimedia, 21*(2), 470–483. https://doi.org/10.1109/TMM.2018.2859034

Luo, W., Tang, Q., Fu, C., & Eberhard, P. (2018). *Deep-sarsa based multi-uav path planning and obstacle avoidance in a dynamic environment*, 102–111. https://doi.org/10.1007/978-3-319-93818-9_10

Mu, X., He, B., Zhang, X., Song, Y., Shen, Y., & Feng, C. (2019). End-to-end navigation for autonomous underwater vehicle with hybrid recurrent neural networks. *Ocean Engineering, 194*, 106602. https://doi.org/10.1016/j.oceaneng.2019.106602

Olcay, E., Schuhmann, F., & Lohmann, B. (2020). Collective navigation of a multi-robot system in an unknown environment. *Robotics and Autonomous Systems, 132*, 103604. https://doi.org/10.1016/j.robot.2020.103604

Osanlou, K., Bursuc, A., Guettier, C., Cazenave, T., & Jacopin, E. (2019). Optimal solving of constrained path-planning problems with graph convolutional networks and optimized tree search. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3519–3525. https://doi.org/10.1109/IROS40897.2019.8968113

Patle, B., Pandey, A., Jagadeesh, A., & Parhi, D. (2018). Path planning in uncertain environment by using firefly algorithm. *Defence Technology, 14*(6), 691–701. https://doi.org/10.1016/j.dt.2018.06.004

Qian, K., Liu, Y., Tian, L., & Bao, J. (2020). Robot path planning optimization method based on heuristic multi-directional rapidly-exploring tree. *Computers & Electrical Engineering, 85*, 106688. https://doi.org/10.1016/j.compeleceng.2020.106688

Qu, H., Yang, S. X., Willms, A. R., & Yi, Z. (2009). Real-time robot path planning based on a modified pulse-coupled neural network model. *IEEE Transactions on Neural Networks, 20*(11), 1724–1739. https://doi.org/10.1109/TNN.2009.2029858

Santos, V. de C., Oório, F. S., Toledo, C. F. M., Otero, F. E. B., & Johnson, C. G. (2016). Exploratory path planning using the max-min ant system algorithm. In *Evolutionary Computation*. https://doi.org/10.1109/CEC.2016.7744327

Sharma, J., Andersen, P., Granmo, O., & Goodwin, M. (2020). Deep q-learning with q-matrix transfer learning for novel fire evacuation environment. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1–19. https://doi.org/10.1109/TSMC.2020.2967936

Shi, P., & Cui, Y. (2010). Dynamic path planning for mobile robot based on genetic algorithm in unknown environment. In *Chinese Control & Decision Conference*. https://doi.org/10.1109/CCDC.2010.5498349

Song, J., Gao, L., & Shan, X. (2017). Historical street network gis datasets of beijing within 5th ring-road. *Chinese Scientific Data, 2*, 1–6. https://doi.org/10.11922/csdata.580.2016.0114

Sturtevant, N. (2012). Benchmarks for grid-based pathfinding. *Transactions on Computational Intelligence and AI in Games, 4*(2), 144–148. https://doi.org/10.1109/TCIAIG.2012.2197681

Tang, C., Sun, R., Yu, S., Chen, L., & Zheng, J. (2019). Autonomous indoor mobile robot exploration based on wavefront algorithm. *Intelligent Robotics and Applications*, 338–348. https://doi.org/10.1007/978-3-030-27541-9_28

Tse, P. W., Lang, S., Leung, K. C., & Sze, H. C. (1998). Design of a navigation system for a household mobile robot using neural networks. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence* (Vol. 3, pp. 2151–2156). https://doi.org/10.1109/IJCNN.1998.687193

Wang, J., Chi, W., Li, C., Wang, C., & Meng, Q. H. (2020). Neural rrt*: Learning-based optimal path planning. *IEEE Transactions on Automation Science and Engineering*, 1–11. https://doi.org/10.1109/TASE.2020.2976560

Watkins, C. J. C. H., & Dayan, P. (1992). Technical note: Q-learning. *Machine Learning, 8*, 279–292. https://doi.org/10.1023/A:1022676722315

Wei, C., & Ni, F. (2018). Tabu temporal difference learning for robot path planning in uncertain environments. *Towards Autonomous Robotic Systems*, 123–134. https://doi.org/10.1007/978-3-319-96728-8_11

Yang, S. X., & Luo, C. (2004). A neural network approach to complete coverage path planning. *IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 34*(1), 718. https://doi.org/10.1109/TSMCB.2003.811769

Yan, C., & Xiang, X. (2018). A path planning algorithm for uav based on improved q-learning. In *2018 2nd International Conference on Robotics and Automation Sciences (ICRAS)* (pp. 46–50). https://doi.org/10.1109/ICRAS.2018.8443226

Zeng, B., Yang, Y., & Xu, Y. (2009). Mobile robot navigation in unknown dynamic environment based on ant colony algorithm. In *Global Congress on Intelligent Systems*. https://doi.org/10.1109/GCIS.2009.274

Zhao, M., Lu, H., Yang, S., & Guo, F. (2020). The experience-memory q-learning algorithm for robot path planning in unknown environment. *IEEE Access, 8*, 47824–47844. https://doi.org/10.1109/ACCESS.2020.2978077

Zhou, S., Liu, X., Xu, Y., & Guo, J. (2018). A deep q-network (dqn) based path planning method for mobile robots. In *2018 IEEE International Conference on Information and Automation (ICIA)* (pp. 366–371). https://doi.org/10.1109/ICInfA.2018.8812452