Adv. Manuf. (2022) 10:114–130
https://doi.org/10.1007/s40436-021-00366-x

# A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm

Zhen Zhang[1] · Rui He[1] · Kuo Yang[1]

Received: 5 March 2021 / Revised: 28 May 2021 / Accepted: 29 May 2021 / Published online: 12 August 2021

**Abstract** In this paper, a bioinspired path planning approach for mobile robots is proposed. The approach is based on the sparrow search algorithm, which is an intelligent optimization algorithm inspired by the group wisdom, foraging, and anti-predation behaviors of sparrows. To obtain high-quality paths and fast convergence, an improved sparrow search algorithm is proposed with three new strategies. First, a linear path strategy is proposed, which can transform the polyline in the corner of the path into a smooth line, to enable the robot to reach the goal faster. Then, a new neighborhood search strategy is used to improve the fitness value of the global optimal individual, and a new position update function is used to speed up the convergence. Finally, a new multi-index comprehensive evaluation method is designed to evaluate these algorithms. Experimental results show that the proposed algorithm has a shorter path and faster convergence than other state-of-the-art studies.

**Keywords** Path planning · Linear path strategy · Sparrow search algorithm · Multi-index comprehensive evaluation algorithm

## 1 Introduction

Mobile robots are widely used in aerospace, entertainment, agriculture, military, mining, and rescue operations [1]. The path planning method plays a key role in the application of mobile robots. Its main purpose is to find a safe and collision-free optimal or suboptimal path from the starting position to the target position in a workspace with obstacles, according to certain performance indicators (walking path, planning time, path smoothness).

There are many classical path planning methods, such as the cell decomposition method [2–4], artificial potential field (APF) algorithm [5], Astar algorithm (A*) [6, 7], probabilistic road map (PRM) algorithm [8], and rapidly exploring random tree (RRT) algorithm [9, 10]. Montiel et al. [11] proposed a new bacterial potential field algorithm based on APF, which had good performance in determining the optimal path time. Jose and Pratihar [12] used Astar for path planning and a genetic algorithm (GA) for task allocation, and completed task allocation and collision-free path planning tasks for three robots at 90 points in a factory. Yan et al. [13] proposed an improved probabilistic road map algorithm that performed well for path planning in narrow passages in the map. Classical algorithms have the advantages of simple principles, but they have certain problems. It is worth noting that some classical methods are deterministic algorithms that are easily trapped in local optimal solutions in complex situations. In addition, the algorithm may not reach the end point in a complex environment. The path generated may be too rough, which would cause collisions between robots and obstacles. The algorithm may converge slowly in a complex environment.

Unlike traditional methods, the biological mechanisms of animals have been optimized through millions of years

✉ Zhen Zhang
zhangzhen_ta@shu.edu.cn

[1] School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, People’s Republic of China

**Table 1** Advantages and disadvantages of each algorithm

| Algorithm | Insufficiency | Advantages | Improvement direction |
|---|---|---|---|
| APF [5] | Long-running time<br>Easy to fall into local optimum<br>There is a problem that the target point cannot be reached | Simple structure<br>The planned path is smooth | Improve the calculation formula of gravitation and repulsion |
| A* [6, 7] | A large amount of calculation<br>Many turning points | The principle is simple<br>Robust | Delete redundant points and inflection points |
| PRM [8] | Low efficiency and long running time<br>The planned path is longer | Strong adaptability to the environment | Increase the effective sampling point density |
| RRT [9, 10] | Long-running time<br>The path is not smooth<br>Low search efficiency in complex environments | Few parameters<br>Strong adaptability to the environment | Make node expansion directional |
| Ant colony optimization (ACO) [21] | Long-running time<br>Easy to fall into local optimum<br>Long path | Fewer adjustment parameters<br>Robust | Dynamically adjust its own parameters<br>Combine with other algorithms |
| GA [22] | Slow convergence<br>Easy to fall into local optimum<br>The coding method of chromosome has a great influence on the optimization ability | Easy to combine with other algorithms | Combine with other algorithms<br>Improve or increase its own operation operator |
| GWO [14] | Easy to fall into local optimum<br>The convergence speed becomes slower in the later stage. | Faster convergence<br>Strong global search ability | Balance the relationship between exploration and utilization |
| ALO [15] | Slow convergence<br>Easy to fall into local optimum<br>High-dimensional solution accuracy is low | Fewer adjustment parameters<br>Robust | Increase the diversity of ant populations |
| WOA [16] | Easy to fall into local optimum<br>Reduced population diversity in the later stage of the algorithm | Fewer adjustment parameters<br>Robust | Increase population diversity<br>Balance the relationship between exploration and utilization |
| SSA [17, 23, 24] | The algorithm is easy to fall into the local optimum. | Short running time<br>Strong global optimization ability | Improve the quality of the initial solution<br>Optimize algorithm parameters |
| Improved sparrow search algorithm (ISSA) | Less population diversity | Short running time<br>Stronger global search ability<br>Low population requirements | Increase population diversity<br>Increase the survival of the fittest mechanism to improve population quality |

of natural evolution, and hence their behaviors can perform many intelligent tasks accurately and robustly. Therefore, we have a good reason to learn from nature to improve path-planning technologies. In recent years, researchers have proposed bioinspired algorithms, such as the gray wolf optimization (GWO) algorithm [14], ant lion optimization (ALO) algorithm [15], whale optimization algorithm (WOA) [16], and sparrow search algorithm (SSA) [17]. The GWO originated from the predation behavior and social hierarchy of gray wolf populations. Gray wolves follow a strict social hierarchy. Xu et al. [18] proposed an

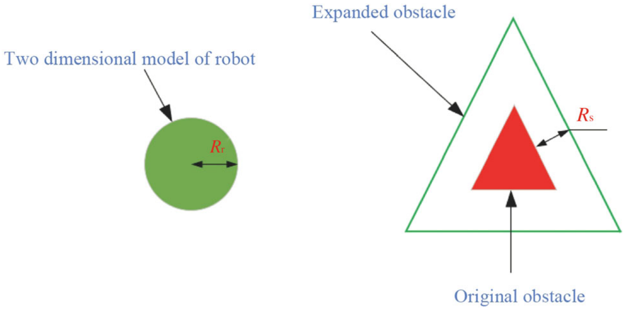

**Fig. 1** Mobile robot size and corresponding obstacle extended size

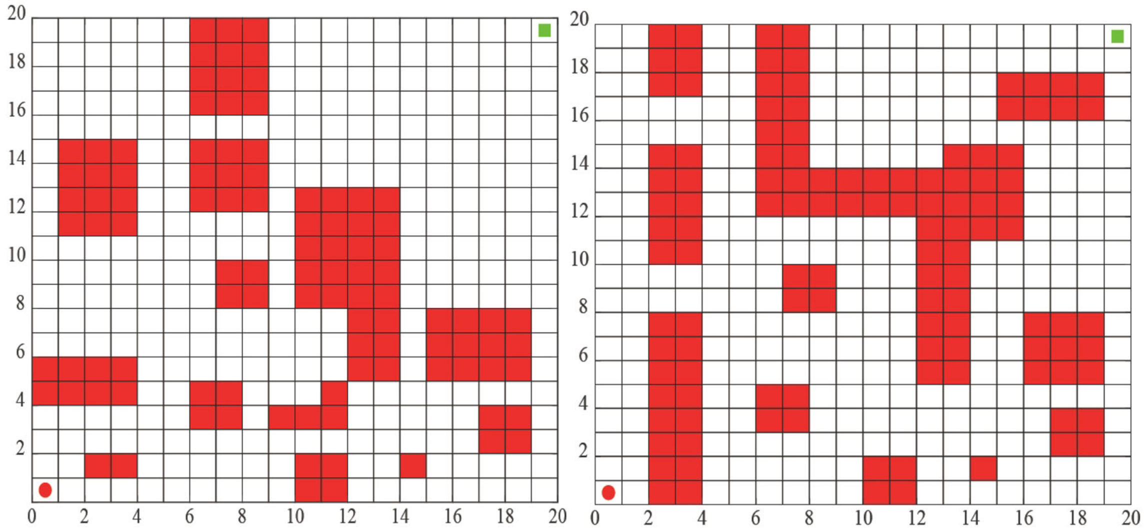

**Fig. 2** Grid map Environment 1 and Environment 2

improved GWO that used a 2-opt dynamic elite mechanism. Through the optimal route selection of tourist attractions, the effectiveness of the proposed method is verified, and the deficiencies of the basic gray GWO are solved. The ALO simulates the hunting mechanism of antlion hunting ants to achieve global optimization. Tian et al. [19] proposed an improved ALO, which integrated a chaotic mutation mechanism and improved the iterative search process, and applied the algorithm to the parameter identification of a hydraulic turbine governing system. The WOA simulates the social behavior of humpback whales and introduces bubble-net hunting strategies. Yildiz [20] proposed a hybrid optimization algorithm based on the Nelder–Mead local search algorithm and the WOA, and applied the proposed algorithm to solve production optimization problems. The SSA is inspired by the foraging behavior and anti-predation behavior of sparrows. Xue and Shen [17] proposed a basic SSA and verified the superiority of this algorithm compared with other algorithms through test results on 19 benchmark functions, and finally applied the algorithm to two engineering examples. To understand the advantages and disadvantages of each algorithm more clearly, this article summarizes the advantages and disadvantages of some methods and the direction of improvement, as listed in Table 1 [5–10, 14–17, 21–24]. In summary, this article proposes an algorithm for obtaining high-quality paths with rapid convergence.

To obtain high-quality paths and fast convergence, a new bioinspired path planning method is proposed in this work, the main aspect of which is an ISSA. First, to solve the problem of many corners in the path planned by the traditional algorithm, a linear path strategy (LPS) is proposed, which uses the method of turning the polyline in the

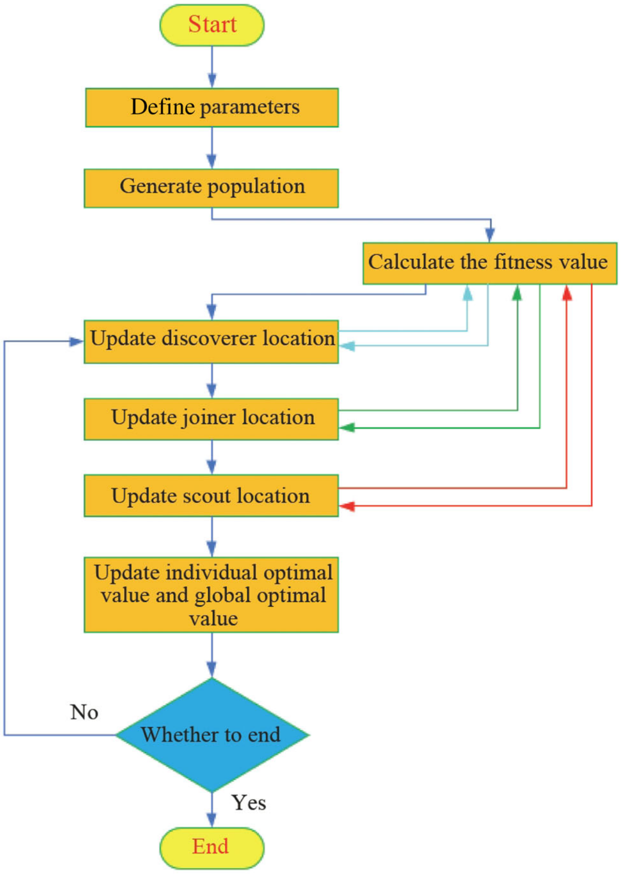
Start
Define parameters
Generate population
Calculate the fitness value
Update discoverer location
Update joiner location
Update scout location
Update individual optimal value and global optimal value
No
Whether to end
Yes
End

**Fig. 3** SSA execution flow chart

corner section of the path into a straight line to smooth the generated path. Then, a new position update function of the SSA is improved to speed up the convergence of the search algorithm. The neighborhood search strategy is used to improve the fitness value of the global optimal individual. Finally, a multi-index comprehensive evaluation method is presented to evaluate the performance of our ISSA and other search algorithms.

The remainder of this paper is organized as follows. Section 2 describes the modeling of a mobile robot working environment. Section 3 proposes the ISSA and a multi-index comprehensive evaluation algorithm. Section 4 shows the results of the experiments with the ISSA and other algorithms. Section 5 uses the proposed path evaluation algorithm to comprehensively evaluate the experimental path of each algorithm. Section 6 concludes the paper with a summary of the main results of this study and a discussion of future works.

## 2 Environment modeling

In this study, the grid method was used to model the map environment. On a two-dimensional map, a mobile robot is regarded as a circle with radius $R_r$, as shown in Fig. 1. To simplify the problem, the mobile robot is regarded as a mass point, and the obstacle is expanded by $R_s$. $R_s$ is obtained by

$$R_s = R_r + \sigma, \tag{1}$$

where $\sigma$ represents the safety distance, which is artificially chosen to prevent the mobile robot from contacting obstacles.

As shown in Fig. 2, two mobile robot working environments, Environments 1 and 2, were established, with a map size of 20×20. Compared with Environment 1, Environment 2 has more obstacles and more complex terrain, which is a challenge for mobile robot path planning methods. The starting point of the planned path is grid (0, 0), represented by a red circle, and the end point is grid (20, 20), represented by a green square.

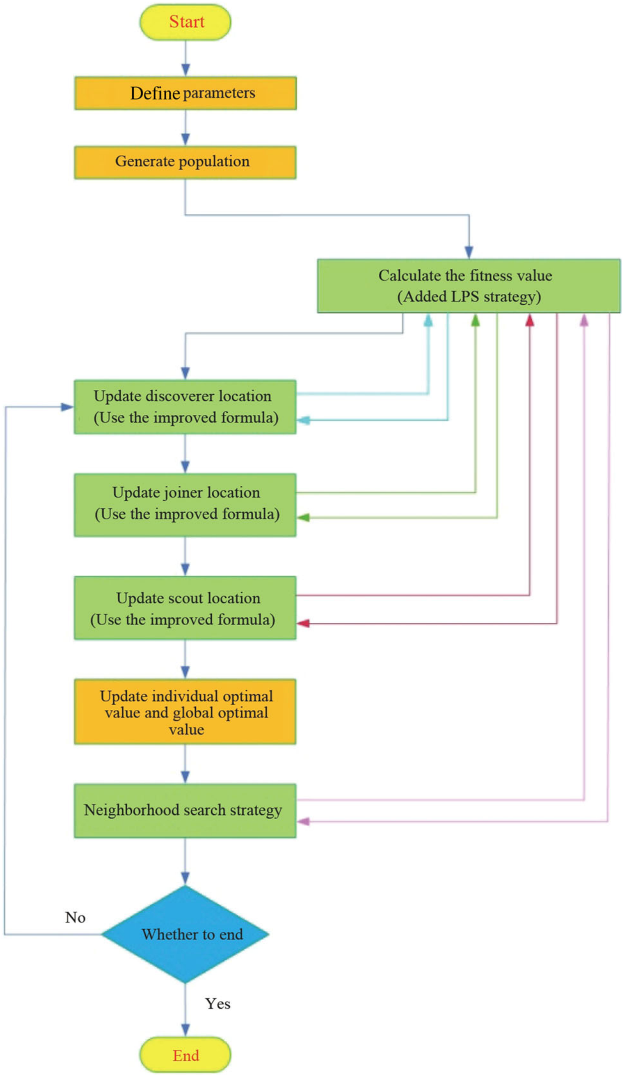
Start
Define parameters
Generate population
Calculate the fitness value
(Added LPS strategy)
Update discoverer location
(Use the improved formula)
Update joiner location
(Use the improved formula)
Update scout location
(Use the improved formula)
Update individual optimal
value and global optimal
value
Neighborhood search strategy
No
Whether to end
Yes
End

**Fig. 4** ISSA execution flow chart

**Table 2** Similarities and differences between SSA and ISSA

| Algorithm | Similar | Different |
|---|---|---|
| SSA and ISSA | The algorithm principle is the same | The ISSA increases the LPS strategy, thereby shortening the path length |
| | The basic framework of the algorithm is similar | The ISSA uses an improved location update formula to update the location of discoverers, joiners, and scouts |
| | | The ISSA increases the neighborhood search strategy, thereby reducing the fitness value of the global optimal individual. |

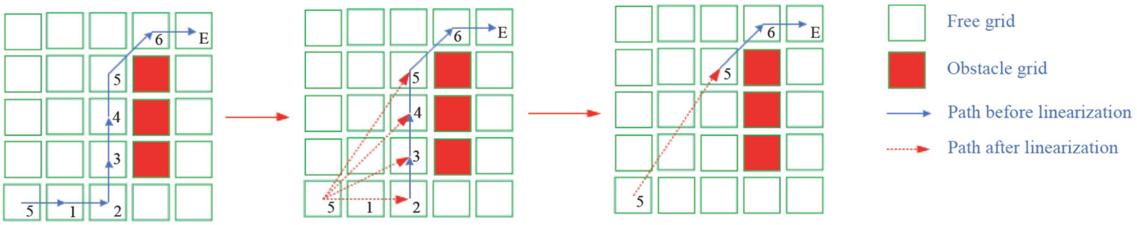

**Fig. 5** LPS execution process

**Table 3** Pseudo code of LPS

```
Algorithm 1 Linear path strategy algorithm
Input: Feasible route.
    Length_route=length(route);
    $i$=1;
Process:
    while $i$ not equal to Length_toute-2
        Extract the coordinates of point $i$ and point $i$+2;
        if There is no obstacle grid point between the coordinate points $x$ and $y$
            route($i$+1)=[ ];
            $i$=$i$-1;
        end
        $i$=$i$+1;
        Length_route=length(route);
    end
Output: The path of linearization
```

## 3 Method

### 3.1 Basic SSA

The SSA simulates the foraging process of sparrow flocks [17]. The foraging process of sparrow flocks is a discoverer-joiner model, with a superimposed reconnaissance and early warning mechanism. There are three distinct sparrow populations: discoverers, joiners, and scouts. Among the sparrows, individuals who can easily find food serve as discoverers, and other individuals serve as joiners. The discoverer has a high fitness value, a wide search range, and guides the population to search and forage. To obtain better fitness, the joiner follows the discoverer to forage. At the same time, a certain proportion of individuals in the population are selected as scouts to observe surrounding companions and dangerous predators, thereby improving their predation and risk prevention abilities.

In the model, the location update formula of the discoverer is given by

$$X_i^{t+1} = \begin{cases} X_i^t \exp\left(-\dfrac{i}{a\, t_{\max}}\right), & \text{if } R < S, \\ X_{i,j}^t + Q\boldsymbol{L}, & \text{if } R \geqslant S, \end{cases} \tag{2}$$

where $t$ represents the current iteration number, and $X_i$ represents the position information of the $i$th sparrow. $a$ is a random number in the range of $(0,1]$. $R(R \in [0,1])$ and $S(S \in [0.5,1])$ represent warning and safety parameters, respectively, where $R$ is a random number and $S$ is a given constant. When $R < S$, no danger is found in the population; the search environment is safe; and the discoverer can conduct a wide range of searches. When $R \geqslant S$, the scouts find danger, adjust the search strategy, and quickly move closer to the safe area. $Q$ is a random number that follows the normal distribution. $\boldsymbol{L}$ represents an all-one matrix of dimensions $1 \times d$.

The location update formula of the joiner is given by

$$X_i^{t+1} = \begin{cases} Q\exp\left(\dfrac{X_{\mathrm{r}} - X_i^t}{i^2}\right), & \text{if } i > \dfrac{n}{2}, \\ X_{\mathrm{b}}^{t+1} + \left|X_i^t - X_{\mathrm{b}}^{t+1}\right| \boldsymbol{A}^+ \boldsymbol{L}, & \text{otherwise}, \end{cases} \tag{3}$$

where $X_{\mathrm{b}}$ is the current best position of the discoverer and $X_{\mathrm{r}}$ represents the current worst position in the world. $\boldsymbol{A}$ represents a matrix of dimensions $1 \times d$, where each element has a value of 1 or $-1$ and $\boldsymbol{A}^+ = \boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\right)^{-1}$.

The location update formula of the scout is shown as

$$X_i^{t+1} = \begin{cases} X_{\mathrm{B}}^t + \beta\left|X_i^t - X_{\mathrm{B}}^t\right|, & \text{if } f_i \neq f_{\mathrm{B}}, \\ X_i^t + K\dfrac{\left|X_i^t - X_{\mathrm{r}}^t\right|}{(f_i - f_{\mathrm{R}}) + \varepsilon}, & \text{if } f_i = f_{\mathrm{B}}, \end{cases} \tag{4}$$

where $X_{\mathrm{B}}$ is the current global optimal position. $\beta$ is a step-length control parameter, a random number drawn on a normal distribution with a mean value of 0 and a variance of 1. $K$ is a random number in the range of $[-1,1]$. $f_i$ represents the current individual fitness value of the sparrow. $f_{\mathrm{B}}$ and $f_{\mathrm{R}}$ represent the current global optimal and worst fitness values, respectively. $\varepsilon$ is a very small constant.

When the iteration ends, the optimization result is output. The overall SSA execution flowchart is shown in Fig. 3.

### 3.2 ISSA

Based on the SSA, we propose a new algorithm, the ISSA. There are three improvements in the ISSA: LPS strategies, neighborhood search strategy, and an improved location

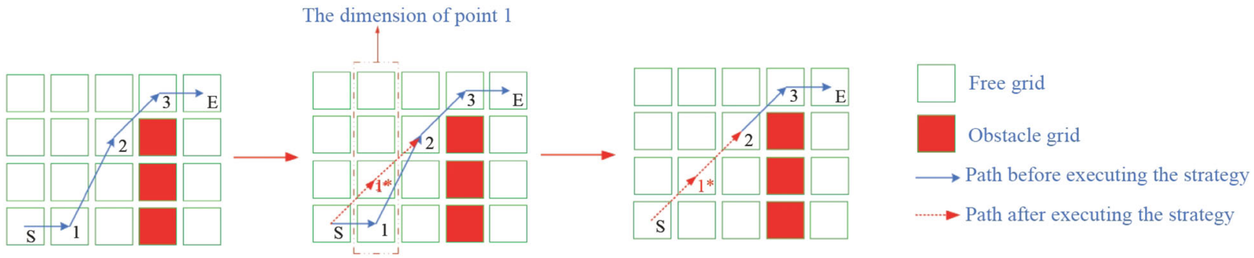
The dimension of point 1
S
1
2
3
E
1*
Free grid
Obstacle grid
Path before executing the strategy
Path after executing the strategy

**Fig. 6** Neighborhood search strategy execution process

**Table 4** Pseudo code of neighborhood search strategy

```
Algorithm 2 Neighborhood search strategy
Input: Optimal fitness individual $X$, Map width $W$.
  Length = The length of path $X$;
  $f$ = Calculate the fitness value of $X$;
Process:
    for $i = 1$ : Length
        $x_{new} = X$;
        $x_{new}(i)$ = Random generate from $W$;
        $f_{new}$ = Calculate the fitness value of $x_{new}$;
        if $f_{new} < f$
            Replace $X$ with path $x_{new}$;
        end
    end
Output: Optimal individual and optimal fitness value after neighborhood search
```

update formula. The LPS strategy was used to calculate the fitness function to linearize the path. The population is updated using the improved position-update formula. The neighborhood search strategy is used after the location update formula to reduce the fitness value of the global optimal individual. Table 2 shows the similarities and differences between the SSA and ISSA. Figure 4 shows the execution flowchart of the ISSA.

### 3.2.1 *LPS*

In Ref. [25], a sequential LPS is proposed. However, the strategy checks whether the path and obstacles intersect, and uses the method of whether each obstacle index is the same as the path index. With many obstacles, the strategy cannot be easily implemented. This study proposes a high-efficiency LPS used within the SSA to improve the path quality and reduce the path acquisition time.

The LPS is divided into two stages: obstacle detection and path connection. Due to the fact that the LPS is used inside the algorithm, a judgment method with a large number of calculations cannot be used. It can be explained that iterations will increase the execution time of the LPS. The LPS is outlined as follows.

Step 1 Start from the starting point, extract three points in order.
Step 2 Find the coordinate range between the first and third points.
Step 3 Determine whether there are obstacles in this range.
Step 4 If there is no obstacle, remove the second point from the path; if there is an obstacle, terminate the cycle and continue to linearize the path.

The execution process of the strategy is illustrated in Fig. 5. Table 3 lists the pseudo-code of the LPS. In the schematic diagram, we separated the grids in the grid map by a certain distance for the sake of clarity. The following diagrams follow this rule. Note that there are no gaps among the grids in the actual grid map.

### 3.2.2 *Neighborhood search strategy*

In the SSA, the global optimal individual in the population plays a role in leading the direction of the population exploration, such that the global optimal individual has a great influence on the entire group. Thus, a neighborhood search is performed on individuals with optimal fitness to reduce their fitness value. The neighborhood search strategy can be divided into the following three steps.

Step 1 Get the dimensionality of the path.
Step 2 Randomly search each dimension of the path and calculate the fitness value of the path after searching.

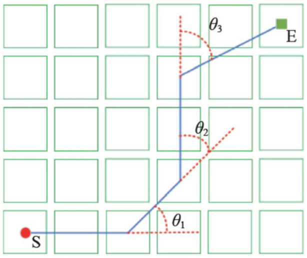

**Fig. 7** Schematic diagram of path corner

Step 3 Judgment. If the path fitness value is lower after the search, replace the previous path with the searched path. If the path fitness is higher after the search, the search result is discarded and the original path is retained.

The implementation process of the neighborhood search strategy is shown in Fig. 6. This strategy searches for the dimension of 1 point and finds a 1* point. By calculating the fitness value, the 1* point constitutes a path with a lower fitness value, thereby replacing 1 point in the previous path with a 1* point. Table 4 presents pseudo code of the neighborhood search strategy process.

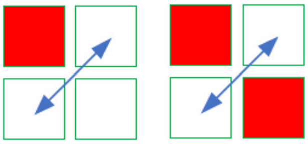

**Fig. 8** Schematic diagram of risk degree calculation

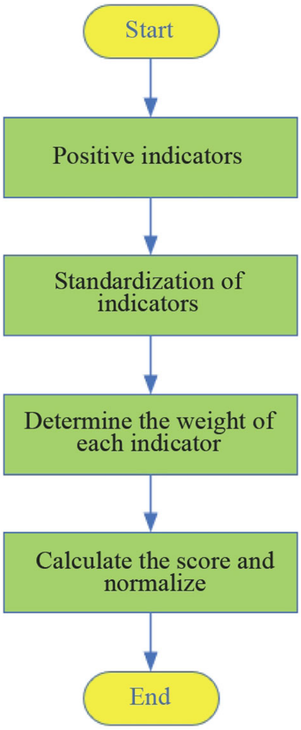
Start
Positive indicators
Standardization of indicators
Determine the weight of each indicator
Calculate the score and normalize
End

**Fig. 9** Evaluation algorithm execution flow chart

### 3.2.3 Improve location update formula

In the position update formula of the basic SSA, the iteration process is $X_i^t \rightarrow X_i^{t+1}$; that is, the position of the sparrow at the next moment depends on the position of the sparrow at the previous moment. But this presents a problem. If the fitness value of the individual $X_i$ at time $t$ is not the optimal fitness of the individual, that is, the position of the individual at time $t$ is not the best at time $1-t$, then the updated position of the individual at time $t$ is poor, compared with the position updated by the individual with the best fitness at time $1-t$. Therefore, the basic SSA is improved in this work to address this problem, by updating the individual with the optimal fitness of $1-t$. The position update formula of the SSA is modified to Eqs. (5)–(7).

The improved position update formula for the discoverer is given by

$$X_i^{t+1} = \begin{cases} X_{\mathrm{e}}^t \exp\left(-\dfrac{i}{at_{\max}}\right), & \text{if } R < S, \\ X_{\mathrm{e}}^t + QL, & \text{if } R \geqslant S, \end{cases} \tag{5}$$

where $X_{\mathrm{e}}^t$ represents the position of individual $i$ with the optimal fitness value at time $1-t$.

The improved position update formula of the joiner is given by

$$X_i^{t+1} = \begin{cases} Q\exp\left(\dfrac{X_{\mathrm{r}} - X_{\mathrm{e}}^t}{i^2}\right), & \text{if } i > \dfrac{n}{2}, \\ X_{\mathrm{b}}^{t+1} + \left|X_{\mathrm{e}}^t - X_{\mathrm{b}}^{t+1}\right| \boldsymbol{A}^{+}\boldsymbol{L}, & \text{otherwise}. \end{cases} \tag{6}$$

The improved position update formula for the scout is given by

$$X_i^{t+1} = \begin{cases} X_{\mathrm{B}}^t + \beta\left|X_{\mathrm{e}}^t - X_{\mathrm{B}}^t\right|, & \text{if } f_i \neq f_{\mathrm{B}}, \\ X_{\mathrm{e}}^t + K\dfrac{\left|X_{\mathrm{e}}^t - X_{\mathrm{r}}^t\right|}{(f_i - f_{\mathrm{R}}) + \varepsilon}, & \text{if } f_i = f_{\mathrm{B}}. \end{cases} \tag{7}$$

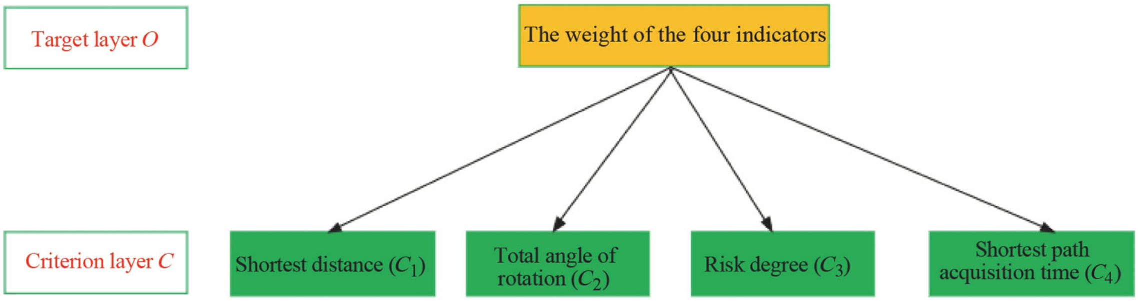
Target layer O
The weight of the four indicators
Criterion layer C
Shortest distance (C1)
Total angle of rotation (C2)
Risk degree (C3)
Shortest path acquisition time (C4)

**Fig.10** Hierarchy diagram for determining weights

## 3.3 Comprehensive path evaluation method

### 3.3.1 Path evaluation index

Adopting a mobile robot to plan and execute the path with the best time as the goal, it is proposed to use multiple indicators to comprehensively evaluate the path. We select four indicators as the basis for path evaluation: the shortest distance value, the total rotation angle value, the risk degree, and the shortest path acquisition time.

The shortest distance value is the length of the optimal path obtained by the algorithm. The total rotation angle value is the sum of the angles of the path planned by the algorithm. An angle diagram is shown in Fig. 7.

The risk degree is the number of intersections between the planned path and obstacles in the grid map. As shown in Fig. 8, if the intersection of the mobile robot's trajectory and the obstacle is 1, the risk degree is 1, and if the intersection of the mobile robot's trajectory and the obstacle is 2, the risk degree is 2.

The shortest path acquisition time is the program execution time to generate the shortest path. Some studies [26, 27] used the execution time of the algorithm program as the path acquisition time, which is unrealistic. In general, the number of iterations required to obtain the shortest path is less than the number of iterations set by the algorithm; therefore, some portion of the number of iterations does not affect the generation of the shortest path. To describe the shortest path acquisition time more accurately, we introduce the concept of the number of shortest path iterations ($N$), which refers to the number of iterations for the algorithm to obtain the shortest path. Therefore, the shortest path acquisition time ($T$) is calculated using

$$T = \frac{N}{t_{\max}} \times T_{\mathrm{c}}, \tag{8}$$

where $T_{\mathrm{c}}$ indicates the execution time of the algorithm and $t_{\max}$ represents the maximum number of iterations set by the algorithm.

Table 5 Judgment matrix

| $O$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| $C_1$ | 1 | 3 | 5 | 2 |
| $C_2$ | 1/3 | 1 | 3 | 1/2 |
| $C_3$ | 1/5 | 1/3 | 1 | 1/3 |
| $C_4$ | 1/2 | 2 | 3 | 1 |

### 3.3.2 Comprehensive path evaluation algorithm

Because the improved algorithm achieves good performance on one index while it may be worse on other indexes, it is necessary to weight each index of the path to comprehensively evaluate the path quality. However, many studies ignore the comprehensive evaluation and conduct a single-index evaluation [28–33]. In this paper, by improving the technique for order preference by similarity to an ideal solution, a comprehensive path evaluation algorithm applied is proposed. The comprehensive path evaluation process is divided into four parts: index normalization, index standardization, determination of index weight, and finally calculation of scores and normalization. The execution process of the algorithm is illustrated in Fig. 9.

(i) Positive indicators

To increase the path score, the better the quality, the more positive is the index. Because the four indicators are all extremely small, they can be transformed and shown as

$$x_{i,j}^{p} = \max_j - x_{i,j}, \tag{9}$$

where $x_{i,j}$ represents the original data; the row (a total of $n$ rows) in the original data represents the algorithm, that is, the evaluation object; the column (a total of $m$ columns) represents the evaluation index; $x_{i,j}^{p}$ represents the normalized data; and $\max_j$ represents the maximum value of index $j$.

(ii) Standardization of indicators

Owing to the different dimensions of the four indicators in this study, the influence of the dimensions was eliminated by

Table 6 Weighted result

| | Arithmetic mean method | Geometric mean method | Eigenvalue method | Mean |
|---|---|---|---|---|
| Shortest distance | 0.476 0 | 0.479 2 | 0.476 8 | 0.477 4 |
| Total angle of rotation | 0.175 6 | 0.172 2 | 0.174 0 | 0.173 9 |
| Risk degree | 0.080 3 | 0.079 1 | 0.079 5 | 0.079 6 |
| Shortest path acquisition time | 0.268 1 | 0.269 5 | 0.269 6 | 0.269 1 |

**Table 7** Parameter settings of each algorithm

| Algorithm | Parameters | Value |
|---|---|---|
| ACO | Number of ants: m | 50 |
| | The maximum number of iterations: NC_max | 200 |
| | Pheromone factor: Alpha | 2 |
| | Heuristic factor: Beta | 6 |
| | Pheromone Evaporation Factor: Rho | 0.1 |
| ACO+GA | Number of ants: m | 50 |
| | The maximum number of iterations: NC_max | 200 |
| | Maximum evolutionary algebra: max_generation | 200 |
| | Probability of crossing: p_crossover | 0.2 |
| | Probability of mutation: p_mutation | 0.05 |
| SSA | Number of population: NP | 50 |
| | The maximum number of iterations: maxgen | 200 |
| | Proportion of discoverers: rPercent | 0.3 |
| | Proportion of scouts: sPercent | 0.2 |
| | Safety value: ST | 0.8 |
| ISSA | Same as SSA parameters setting | |

$$X_{i,j} = \frac{x_{i,j}^{p}}{\sqrt{\sum_{k=1}^{n}\left(x_{k,j}^{p}\right)^{2}}}, \qquad (10)$$

where $X_{i,j}$ represents the standardized data.

(iii) Determination of index weight

The analytic hierarchy process is used to determine the weights between the path evaluation indicators. The weight determination process is shown as follows.

Step 1 Establish a hierarchical structure diagram

The weight determination problem is decomposed into two layers: the first layer is the target layer $O$, which is the weight of the four indicators; the second layer is the criterion layer, including the shortest distance $C_1$, the total turning angle $C_2$, the risk degree $C_3$, and the shortest path acquisition time $C_4$. A hierarchical structure diagram is constructed, as shown in Fig. 10.

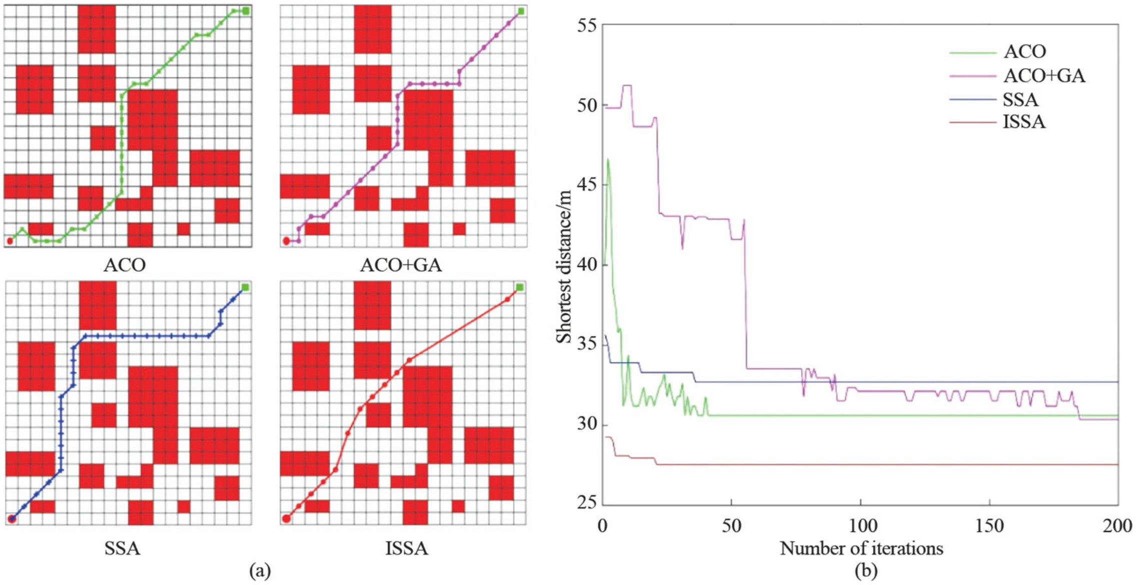

**Fig. 11** Experiment with ACO, ACO+GA, SSA, ISSA in Environment 1 **a** the shortest path diagram, **b** the shortest path convergence

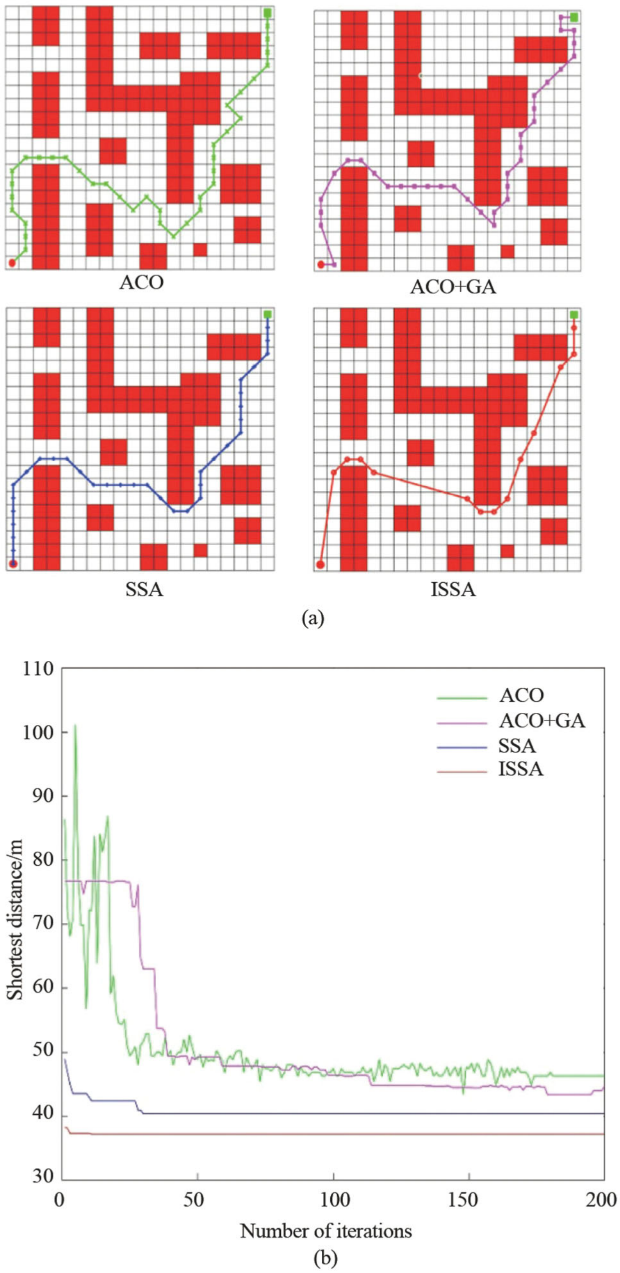

**Fig. 12** Experiment with ACO, ACO+GA, SSA, ISSA in Environment 2 **a** the shortest path diagram, **b** the shortest path convergence

Step 2 Determining judgment matrix

The mobile robot scene in this study is as follows: the mobile robot can turn four wheels with differential speed and medium load. To determine the judgment matrix, the judgment matrix is assigned a value according to the degree of influence of each index on the path quality, as shown in Table 5. The order of importance of the indicators in this study is as follows: the shortest path distance is the most important, followed by the shortest path acquisition time, the total rotation angle of the path, and the path risk degree. After the obstacles are enlarged, although the path of the mobile robot and the obstacles on the map have intersections, there is still a safe distance $\sigma$ between the obstacles and the mobile robot, and thus we believe that the importance of the path risk degree ranks last.

Step 3 Consistency test

The above judgment matrix cannot be used directly, and a consistency test is required. Set $R_{\mathrm{C}}$ as the consistency test parameter, the test formula is given by

$$R_{\mathrm{C}} = \frac{I_{\mathrm{C}}}{I_{\mathrm{R}}}, \tag{11}$$

where $I_{\mathrm{R}}$ is the random consistency index, which can be obtained from a lookup table, and $I_{\mathrm{C}}$ is the consistency index, which is calculated by

$$I_{\mathrm{C}} = \frac{\lambda_{\max} - d}{d - 1}, \tag{12}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the judgment matrix and $d$ is the dimension of the judgment matrix.

It is generally considered that if $R_{\mathrm{C}} < 0.1$, the consistency of the judgment matrix is acceptable. $R_{\mathrm{C}} = 0.022\,2(<0.1)$ is calculated to judge that the matrix passes the consistency test.

Step 4 Get the index weight

Index weights were calculated using the arithmetic mean, geometric mean, and eigenvalue methods. To eliminate the error caused by the calculation method, the calculation results of the three methods were averaged by taking their arithmetic mean. The index weights are presented in Table 6.

The formula for calculating the weight using the arithmetic mean method is shown as

$$w_i = \frac{1}{n} \sum_{j=1}^{n} \frac{a_{i,j}}{\sum_{k=1}^{n} a_{k,j}}, \tag{13}$$

where $w_i$ represents the weight of the $i$th indicator; $n$ represents the number of indicators; and $a_{i,j}$ represents the importance of the $i$th indicator relative to the $j$th indicator in the judgment matrix.

The calculation formula for the geometric mean method for calculating the weight is shown as

$$w_i = \frac{\left(\prod_{j=1}^{n} a_{i,j}\right)^{\frac{1}{n}}}{\sum_{k=1}^{n} \left(\prod_{j=1}^{n} a_{k,j}\right)^{\frac{1}{n}}}. \tag{14}$$

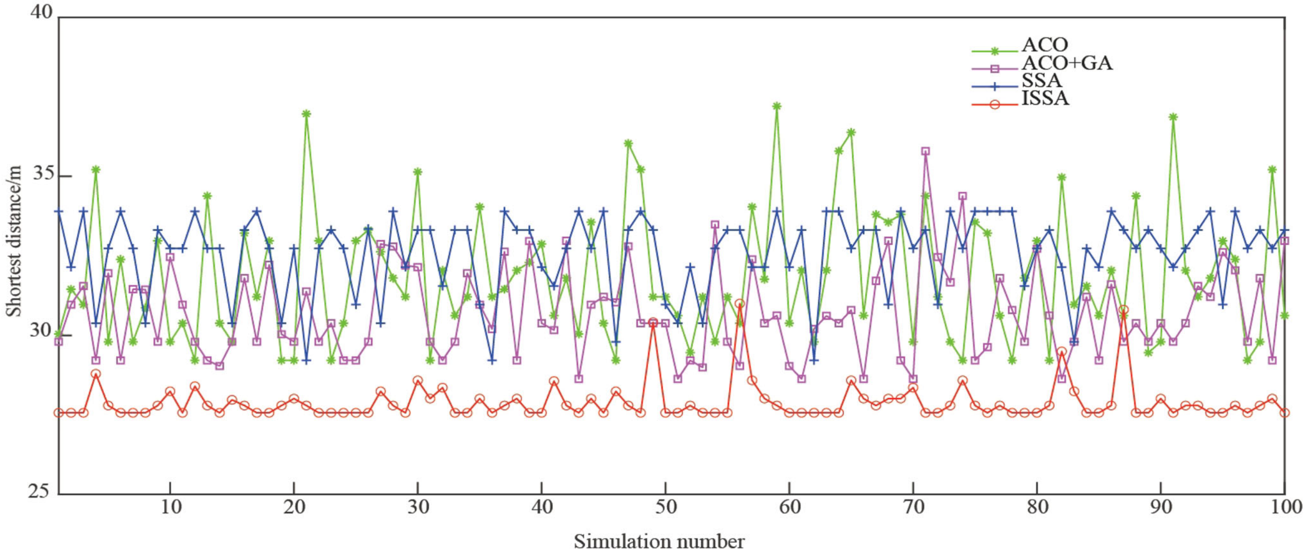

**Fig. 13** Comparison of the shortest path value of Environment 1

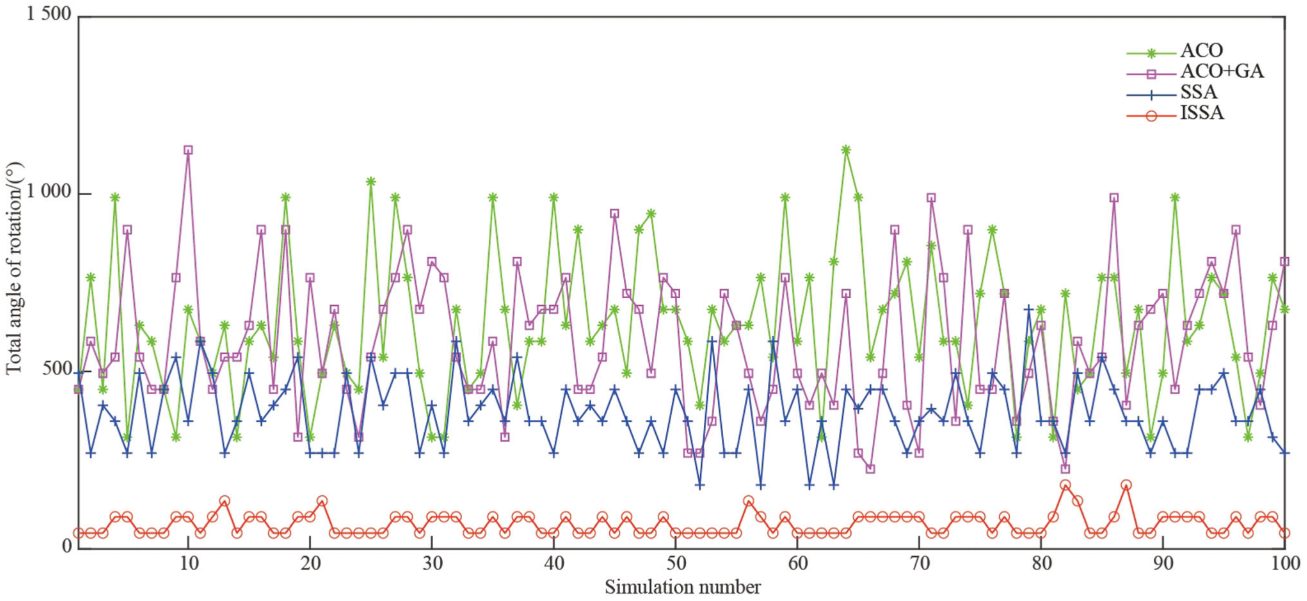

**Fig. 14** Comparison of total rotation angle value of Environment 1

(iv) Score calculation and normalization

The evaluation algorithm proposed in this paper integrates classic evaluation methods and is innovatively applied to the path evaluation of mobile robots.

The formula for calculating the maximum value of each evaluation index is given by

$$X_i^{\max} = \max\left(X_{1,i}, X_{2,i}, \cdots, X_{n,i}\right), \tag{15}$$

where $i = 1, 2, \cdots, m$.

The formula for calculating the minimum value of each evaluation index is given by

$$X_i^{\min} = \min\left(X_{1,i}, X_{2,i}, \cdots, X_{n,i}\right). \tag{16}$$

The calculation formula for the distance between the *i*th evaluation object and the maximum value is shown as

$$D_i^{\max} = \sqrt{\sum_{j=1}^{m} \omega_j \left(X_j^{\max} - X_{i,j}\right)^2}. \tag{17}$$

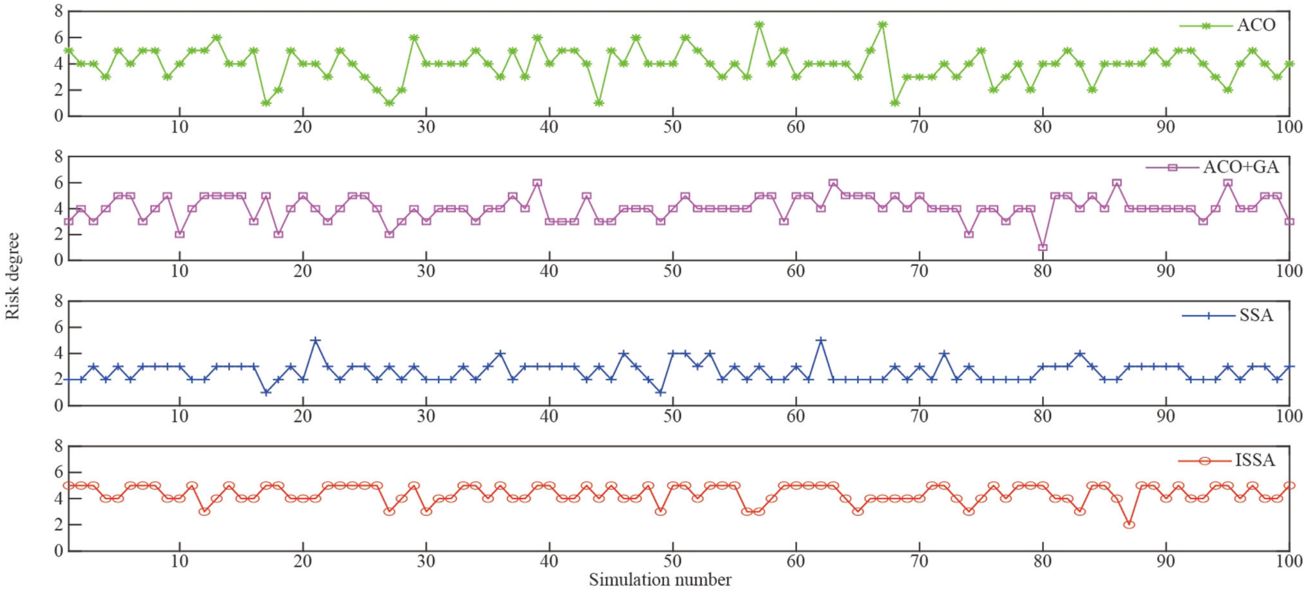

**Fig. 15** Comparison of the risk degree value of Environment 1

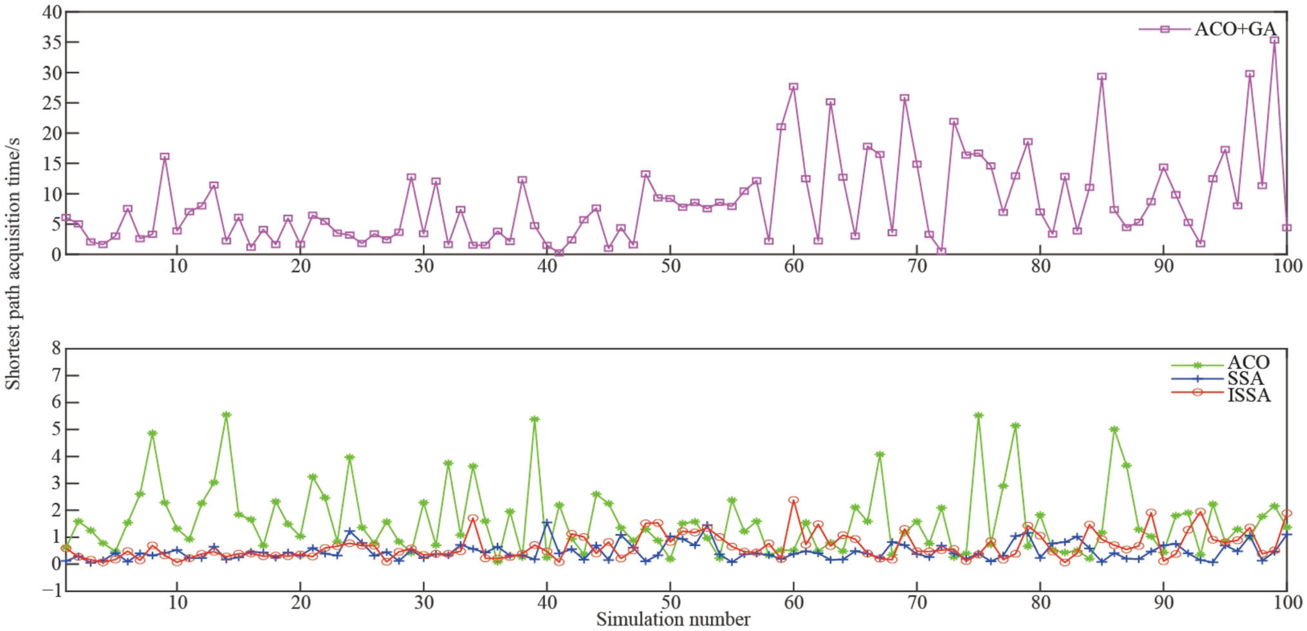

**Fig. 16** Comparison of the shortest path acquisition time of Environment 1

The calculation formula for the distance between the *i*th evaluation object and the minimum value is shown as

$$D_i^{\min} = \sqrt{\sum_{j=1}^{m} \omega_j \left(X_j^{\min} - X_{i,j}\right)^2}. \tag{18}$$

Using Eqs. (15)–(18), we can calculate the score of the *i*-th (*i*=1,2,···,*n*) evaluation object without normalization, shown as

$$S_i = \frac{D_i^{\min}}{D_i^{\min} + D_i^{\max}}. \tag{19}$$

It can be seen from Eq. (19) that $0 \leqslant S_i \leqslant 1$, and the larger $S_i$, the closer it is to the maximum value. The score $S_i$ was normalized using

$$\overline{S_i} = \frac{S_i}{\sum_{i=1}^{n} S_i}. \tag{20}$$

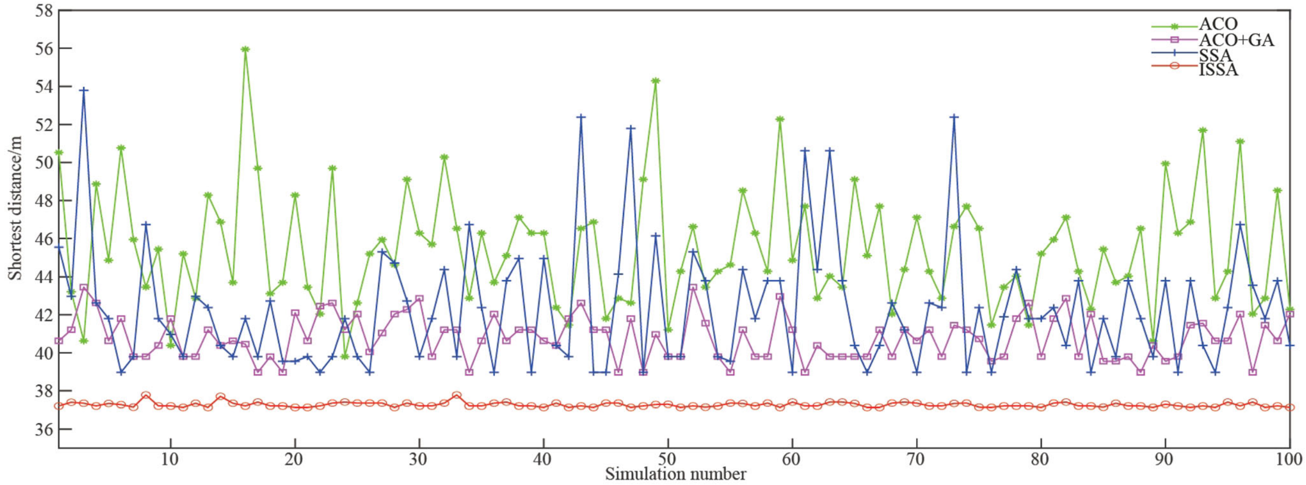

**Fig. 17** Comparison of the shortest path value of Environment 2

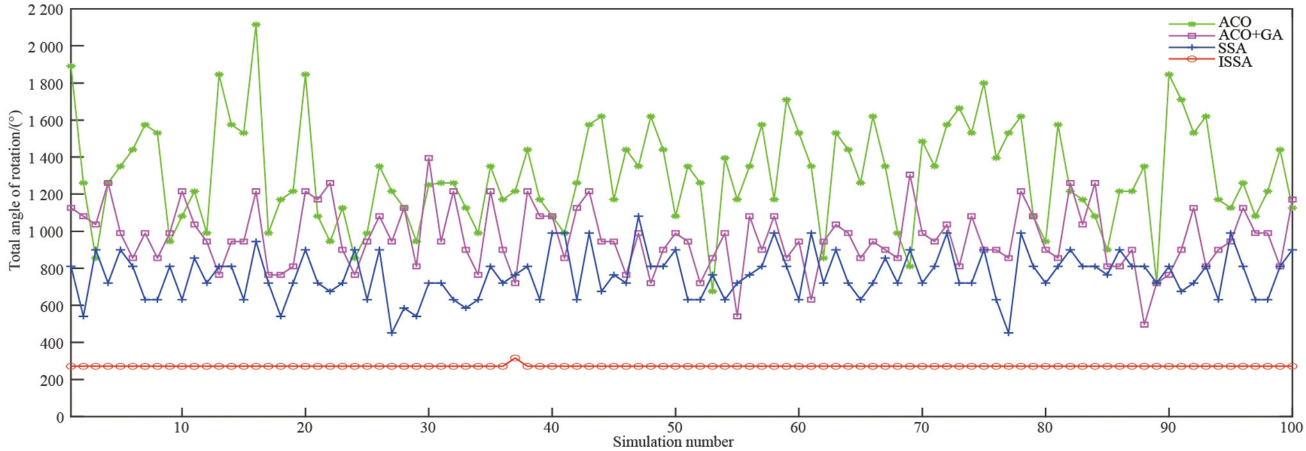

**Fig. 18** Comparison of total rotation angle value of Environment 2

## 4 Experiment

To demonstrate the superiority of the proposed algorithm, ISSA, compared with other popular algorithms, experiments were carried out under the same environment and parameters. We compare the ISSA with the basic SSA, ACO, and ACO fused with a GA to evaluate the performance of the algorithm.

In this study, ISSA, SSA, ACO, and ACO+GA were all simulated using the same software and hardware platform, and the programming environment was MATLAB R2019b (MathWorks Co., USA). To establish a fair comparison of the algorithms, the basic parameters of each algorithm were set to the same value, as shown in Table 7.

To verify the effectiveness of the proposed ISSA, two maps with different difficulty levels, Environment 1 and Environment 2, were used. The experimental results for Environments 1 and 2 are shown in Figs. 11 and 12, respectively. The results of the path planning of the four algorithms in this article in Environments 1 and 2 can also be observed through a video, which is included in the supplementary information.

The code for the experiment and discussion in this article is published on the GitHub platform. The code can be accessed via the following link: https://github.com/herryCccc/Mobile-robot-path-planning.

Figure 11 shows the shortest path diagram and the shortest path convergence process planned by ACO, ACO + GA, SSA, and ISSA in Environment 1. Obviously, ISSA has better results than the other algorithms in terms of path length, path smoothness, and the number of shortest path convergences. This shows that ISSA performs well in

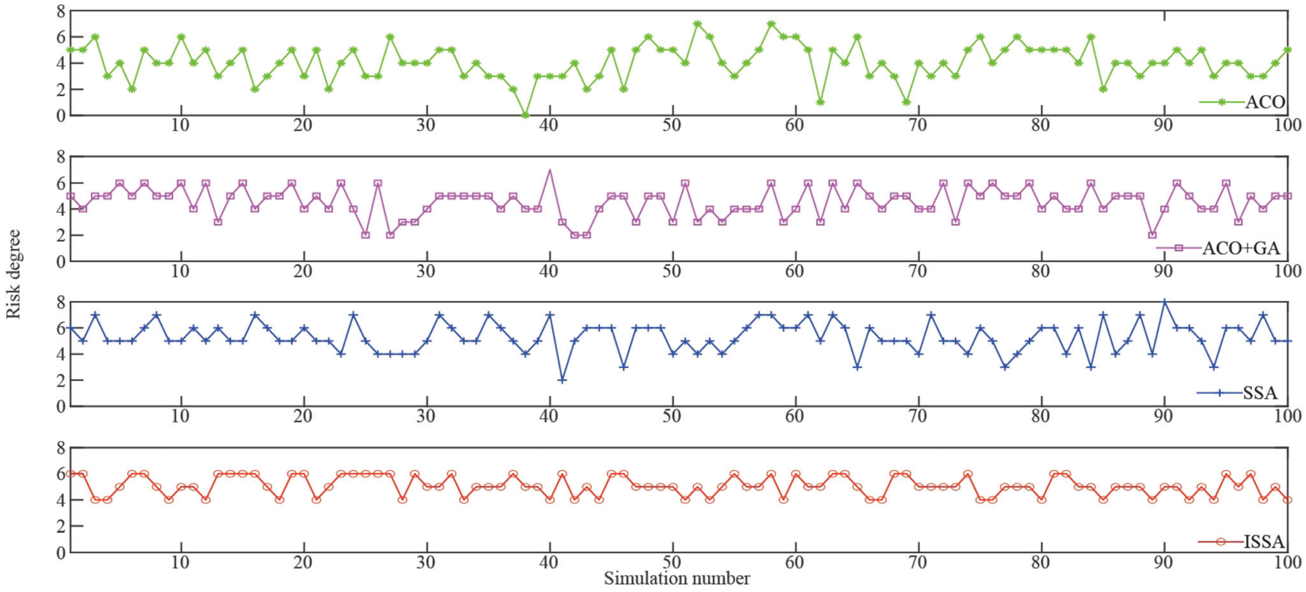

**Fig. 19** Comparison of the risk degree value of Environment 2

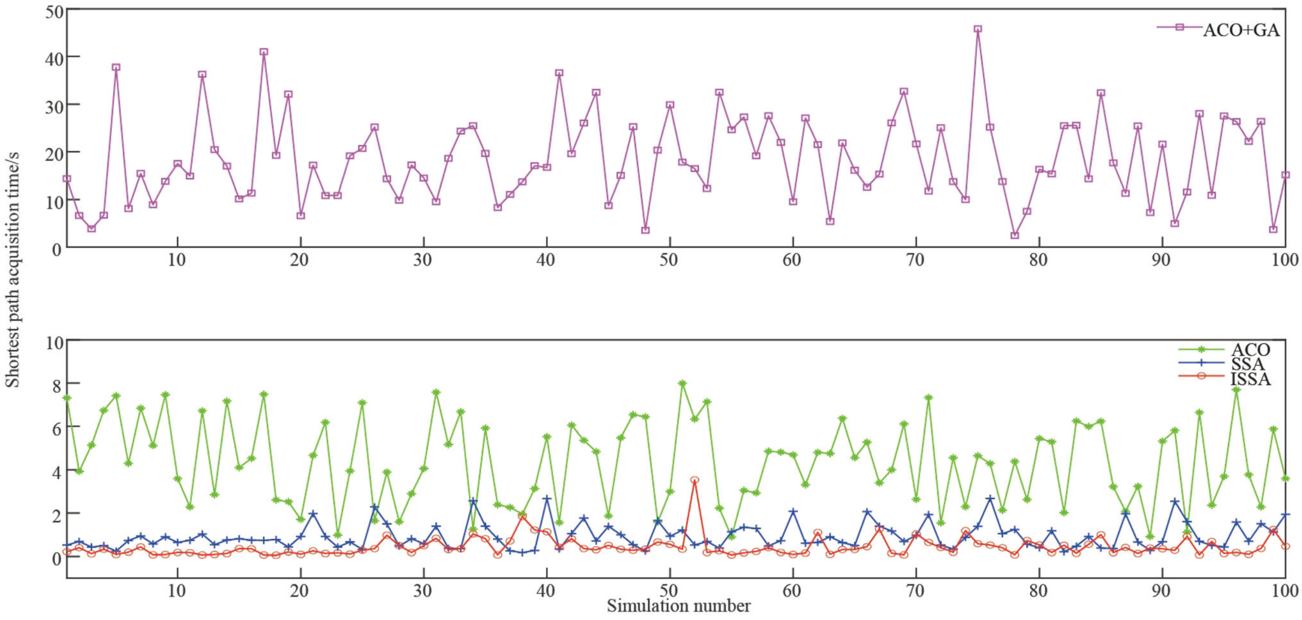

**Fig. 20** Comparison of the shortest path acquisition time of Environment 2

Environment 1. Figure 12 shows the shortest path diagram and the convergence process of the shortest path planned by ACO, ACO + GA, SSA, and ISSA in Environment 2. For more complex map conditions, ISSA still performs better than ACO, ACO+GA, and SSA.

## 5 Discussion

To evaluate the robustness of ISSA, we repeated the four algorithms of ACO, ACO+GA, SSA, and ISSA 100 times on Environments 1 and 2. The four indicators of the shortest distance, total rotation angle, risk degree, and the shortest path acquisition time of the four algorithms were collected.

**Table 8** Maximum, minimum, and average values of four indexes of ACO, ACO+GA, SSA, and ISSA in Environments 1 and 2

| | | ACO | ACO+GA | SSA | ISSA |
|---|---|---|---|---|---|
| *Environment 1* | | | | | |
| Shortest distance | Maximum | 37.213 2 | 35.7990 | 33.899 5 | 31.003 7 |
| | Minimum | 29.213 2 | 28.627 4 | 29.213 2 | 27.560 2 |
| | Average | 31.817 26 | 30.706 5 | 32.601 0 | 27.893 4 |
| Total angle of rotation | Maximum | 1 125 | 1 125 | 675 | 180 |
| | Minimum | 315 | 225 | 180 | 45 |
| | Average | 627.75 | 588.60 | 385.45 | 70.20 |
| Risk degree | Maximum | 7 | 6 | 5 | 5 |
| | Minimum | 1 | 1 | 1 | 2 |
| | Average | 3.96 | 4.07 | 2.63 | 4.38 |
| Shortest path acquisition time | Maximum | 5.542 6 | 35.331 3 | 1.549 1 | 2.383 9 |
| | Minimum | 0.084 1 | 0.230 9 | 0.045 1 | 0.061 5 |
| | Average | 1.617 8 | 8.521 7 | 0.463 5 | 0.653 0 |
| *Environment 2* | | | | | |
| Shortest distance | Maximum | 55.941 1 | 43.455 8 | 53.799 0 | 37.782 7 |
| | Minimum | 39.799 0 | 38.970 6 | 38.970 6 | 37.132 7 |
| | Average | 45.453 3 | 40.783 1 | 42.313 667 | 37.264 7 |
| Total angle of rotation | Maximum | 2 115 | 1 395 | 1 080 | 315 |
| | Minimum | 675 | 495 | 450 | 270 |
| | Average | 1 300.85 | 964.35 | 762.75 | 270.45 |
| Risk degree | Maximum | 7 | 7 | 8 | 6 |
| | Minimum | 0 | 2 | 2 | 4 |
| | Average | 4.08 | 4.55 | 5.32 | 5.08 |
| Shortest path acquisition time | Maximum | 7.996 4 | 45.799 3 | 2.678 9 | 3.537 7 |
| | Minimum | 0.893 3 | 2.422 8 | 0.170 1 | 0.051 4 |
| | Average | 4.342 0 | 18.387 6 | 0.933 5 | 0.436 4 |

**Table 9** Average of the results of 100 times of ACO, ACO + GA, SSA, ISSA in Environments 1 and 2

| | | ACO | ACO+GA | SSA | ISSA |
|---|---|---|---|---|---|
| Environment 1 | Average shortest distance | 31.817 2 | 30.706 5 | 32.601 0 | 27.893 4 |
| | Average total angle of rotation | 627.75 | 588.60 | 385.45 | 70.20 |
| | Average risk degree | 3.96 | 4.07 | 2.63 | 4.38 |
| | Average shortest path acquisition time | 1.617 8 | 8.521 7 | 0.463 5 | 0.653 0 |
| Environment 2 | Average shortest distance | 45.453 3 | 40.783 1 | 42.313 7 | 37.264 7 |
| | Average total angle of rotation | 1300.85 | 964.35 | 762.75 | 270.45 |
| | Average risk degree | 4.08 | 4.55 | 5.32 | 5.08 |
| | Average shortest path acquisition time | 4.342 0 | 18.387 6 | 0.933 5 | 0.436 4 |

The repetitive experimental results of the four indicators of the four algorithms in Environment 1 are shown in Figs. 13–16. It can be seen from Figs. 13 and 14 that the shortest distance and the total rotation angle of the path planned by ISSA are significantly better than those of the other algorithms. However, from Figs. 15 and 16, the path risk and shortest path acquisition time are not optimal for ISSA. We surmise that this is because a shorter path will cause the path to get closer to the obstacle. As the LPS and neighborhood search strategies are added to the basic SSA, the shortest path acquisition time is slightly increased compared with SSA. As ISSA shows different performances with respect to different indicators, it also reflects the importance of comprehensive path evaluation.

The repetitive experimental results of the four indicators of the four algorithms of Environment 2 are shown in Figs. 17–20. The results show that ISSA surpasses the other algorithms regarding three indicators: the shortest distance, total rotation angle, and shortest path acquisition time. It is worth noting that ISSA surpasses SSA in the shortest path

**Table 10** Comprehensive score of the path planned by ACO, ACO+GA, SSA, and ISSA

| | Environment 1 | Environment 2 |
|---|---|---|
| ACO | 0.105 8 | 0.104 9 |
| ACO+GA | 0.092 7 | 0.190 1 |
| SSA | 0.205 5 | 0.233 8 |
| ISSA | 0.596 0 | 0.471 1 |

acquisition time. This is because, for more difficult maps, the performance of ACO, ACO+GA, and SSA to obtain the optimal path decreases, whereas the performance of ISSA remains unchanged, which also indicates that ISSA is robust to environmental changes. However, it can be seen from Fig. 19 that the risk degree of the path planned by ISSA is still high. This is because the process of shortening the path also increases the risk.

Because ISSA shows different performances (better or worse) with respect to different indicators, it is necessary to comprehensively evaluate the path planned by ISSA. Through the test results of ACO, ACO+GA, SSA, and ISSA for the mobile robot in Environments 1 and 2, we count the maximum, minimum, and average values of each indicator, and plot the data in Tables 8 and 9. In Environment 1, ISSA has the best performance for the shortest distance and total rotation angle. However, SSA has better scores in the shortest path acquisition time and risk degree. In Environment 2, ISSA also has the best performance for the shortest distance and total rotation angle. Additionally, it surpasses SSA in the minimum and average values of the shortest path acquisition time. However, it is inferior to the other algorithms in terms of path risk, as explained earlier.

The proposed evaluation algorithm is used to process the data and obtain comprehensive scores of the paths planned by ACO, ACO+GA, SSA, and ISSA for Environments 1 and 2, as shown in Table 10. It can be concluded from Table 10 that ISSA has the highest comprehensive score in Environments 1 and 2. In Environment 1, the ISSA score is increased by 190% relative to the SSA score, whereas in environment 2, the ISSA score is improved with respect to the SSA score by 102%. Therefore, it can be concluded that the comprehensive performance of the proposed algorithm surpasses that of the other algorithms. Moreover, the proposed algorithm surpasses the other algorithms in terms of the shortest path and convergence speed.

## 6 Conclusions

In mobile robot path planning, traditional algorithms easily fall into local optima and exhibit slow convergence. ISSA, an improved algorithm based on the LPS, neighborhood search strategy, and improved location update formula, is proposed in this study to address the aforementioned challenges. This algorithm has the characteristics of fast convergence and strong optimization ability. Experiments were conducted in two different environments to verify the performance of the algorithm. Because of the different performance of the path planned by each algorithm with respect to each index, this work proposes a comprehensive evaluation algorithm to evaluate the quality of the path planned by each algorithm. The experimental results show that the proposed algorithm represents significant progress compared with the current algorithm. In the future work, it will be very meaningful to apply ISSA to actual robot path planning. In addition, we will also use ISSA for dynamic obstacles and multirobot path planning.

**Acknowledgements** This research was jointly supported by the National Key R&D Program of China (Grant No. 2018YFB1309200) and the Opening Project of Shanghai Robot Industry R&D and Transformation Functional Platform.

## References

1. Patle BK, Ganesh BL, Anish P et al (2019) A review: on path planning strategies for navigation of mobile robot. Def Technol 15:582–606
2. Gonzalez R, Kloetzer M, Mahulea C (2017) Comparative study of trajectories resulted from cell decomposition path planning approaches. In: 2017 21st international conference on system theory, control and computing, Sinaia, pp 49–54
3. Zhang Z, Yang X (2019) Bio-inspired motion planning for reaching movement of a manipulator based on intrinsic tau jerk guidance. Adv Manuf 7:315–325
4. Yang K, Tang Y, Zhang Z (2021) Parameter identification and state-of-charge estimation for lithium-ion batteries using separated time scales and extended Kalman filter. Energies 14(4):1054. https://doi.org/10.3390/en14041054
5. Lee K, Choi D, Kim D (2021) Incorporation of potential fields and motion primitives for the collision avoidance of unmanned aircraft. Appl Sci Basel 11(7):3103. https://doi.org/10.3390/app11073103
6. Guruji AK, Agarwal H, Parsediya DK (2016) Time-efficient A* algorithm for robot path planning. In: The 3rd international conference on innovations in automation and mechatronics engineering, Elsevier, Vallabh Vidhyanagar, pp 144–149
7. Chen C, Cai J, Wang Z et al (2020) An improved A* algorithm for searching the minimum dose path in nuclear facilities. Prog Nucl Energy 126:103394. https://doi.org/10.1016/j.pnucene.2020.103394
8. Chen G, Luo N, Liu D et al (2021) Path planning for manipulators based on an improved probabilistic roadmap method. Robot Comput Integr Manuf 72:102196. https://doi.org/10.1016/j.rcim.2021.102196
9. Sun Y, Zhang C, Sun P et al (2020) Safe and smooth motion planning for mecanum wheeled robot using improved RRT and cubic spline. Arab J Sci Eng 45:3075–3090
10. Wu X, Xu L, Zhen R et al (2019) Biased sampling potentially guided intelligent bidirectional RRT algorithm for UAV path planning in 3D environment. Math Probl Eng 2019:5157403. https://doi.org/10.1155/2019/5157403

11. Montiel O, Orozco-Rosas U, Sepúlveda R (2015) Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles. Expert Syst Appl 42:5177–5191
12. Jose K, Pratihar DK (2016) Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. Robot Auton Syst 80:34–42
13. Yan F, Liu YS, Xiao JZ (2013) Path planning in complex 3D environments using a probabilistic roadmap method. Int J Autom Comput 10:525–533
14. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
15. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
16. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
17. Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng 8:22–34
18. Xu R, Cao M, Huang M et al (2018) Research on the quasi-TSP problem based on the improved grey wolf optimization algorithm: a case study of tourism. Geogr Geo Inf Sci 34:14–21
19. Tian T, Liu C, Guo Q et al (2018) An improved ant lion optimization algorithm and its application in hydraulic turbine governing system parameter identification. Energies 11:95. https://doi.org/10.3390/en11010095
20. Yildiz AR (2019) A novel hybrid whale-Nelder-Mead algorithm for optimization of design and manufacturing problems. Int J Adv Manuf Technol 105:5091–5104
21. Wang X, Shi H, Zhang C (2016) Path planning for intelligent parking system based on improved ant colony optimization. IEEE Access 8:65267–65273
22. Niu H, Ji Z, Savvaris A et al (2020) Energy efficient path planning for nnmanned surface vehicle in spatially-temporally variant environment. Ocean Eng 196:106766. https://doi.org/10.1016/j.oceaneng.2019.106766
23. Zhang C, Ding S (2021) A stochastic configuration network based on chaotic sparrow search algorithm. Knowl Based Syst 220:106924. https://doi.org/10.1016/j.knosys.2021.106924
24. Liu G, Shu C, Liang Z et al (2021) A modified sparrow search algorithm with application in 3D route planning for UAV. Sensors 21:1224. https://doi.org/10.3390/s21041224
25. Raouf F, Mohammed B, Tamer R et al (2020) Enhancing path quality of real-time path planning algorithms for mobile robots: a sequential linear paths approach. IEEE Access 8:167090–167104
26. Ajeil FH, Ibraheem KI, Sahib MA et al (2018) Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm. Appl Soft Comput 89:106076. https://doi.org/10.1016/j.asoc.2020.106076
27. Li X, Huang Y, Zhou Y et al (2018) Robot path planning using improved artificial bee colony algorithm. In: 2018 IEEE 3rd advanced information technology, electronic and automation control conference, Chongqing, China, pp 603–607
28. Zhang D, You X, Liu S et al (2020) Dynamic multi-role adaptive collaborative ant colony optimization for robot path planning. IEEE Access 8:129958–129974
29. Zinage V, Ghosh S (2020) Directional sampling-based generalized shape expansion for accelerated motion planning in 2-D obstacle-cluttered environments. IEEE Contr Syst Lett 5:1067–1072
30. Huang Y, Li Z, Jiang Y et al (2019) Cooperative path planning for multiple mobile robots via HAFSA and an expansion logic strategy. Appl Sci Basel 9:672. https://doi.org/10.3390/app9040672
31. Alaa T, Mohamed E, Aboul EH et al (2019) Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm. Cluster Comput 22:4745–4766
32. Hassani I, Maalej I, Rekik C (2018) Robot path planning with avoiding obstacles in known environment using free segments and turning points algorithm. Math Probl Eng 2018:2163278. https://doi.org/10.1155/2018/2163278
33. Wang Z, Xiang X (2018) Improved A star algorithm for path planning of marine robot. In: 2018 37th Chinese control conference. IEEE, Wuhan, China, pp 5410–5414

**Zhen Zhang** received his Ph.D. degree in Mechanical Engineering from Shanghai University, China, in 2006; his master degree in Machinery Manufacturing and Automation from Shandong University of Science and Technology, China, in 2002; and his Bachelor of Science degree from Sichuan University, China, in 1998. He is currently an Associate Professor in the Department of Precise Mechanical Engineering at Shanghai University.

**Rui He** is studying for a master degree in Mechatronic Engineering at Shanghai University, China; he received his bachelor degree in Mechatronic Engineering from Xihua University, China, in 2020.

**Kuo Yang** is studying for a Ph.D. degree in Mechanical Engineering at Shanghai University, China; he received a Bachelor degree in Automotive Engineering from Shanghai Dianji University, China in 2018.