



An improved sparrow search based intelligent navigational algorithm for local path planning of mobile robot

Guangjian Zhang¹ · Enhao Zhang¹

Received: 12 November 2021 / Accepted: 6 June 2022 / Published online: 24 June 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

In this paper, an improved sparrow search algorithm (SSA) for local path planning problem of mobile robot in an unknown environment is presented. The problems of premature convergence and decline of population diversity of basic SSA are solved by the inspiration of fitness-distance balance (FDB) selection and Harris Hawks Algorithm. A hybrid fitness function is formulated considering both path length and path safety, which enables the mobile robot to move to the target location safely. The effectiveness and superiority of the proposed improved SSA (ISSA) is verified in CEC 2017 suite for comparison experiments with multiple intelligent optimization algorithms. Local path planning simulation experiments are implemented using the proposed algorithm in the unknown environment and compared with other algorithms, and the results show that our algorithm is effective and robust in solving local path planning problem of mobile robots.

Keywords Local path planning · Sparrow search algorithm · Swarm intelligence algorithm · Mobile robot

1 Introduction

Mobile robot is an intelligent autonomous system with wide applications in industrial, medical, military, and aerospace fields. Path planning is one of the significant and challenging tasks in mobile robotics. The path planning problem can be divided into global path planning and local path planning according to the level of knowledge of map information. Global path planning means that the robot has sufficient information about the environment before planning the path, and when the robot is not aware of the obstacle information in the environment is called local path planning. Therefore, the correct path planning method is necessary to ensure that the robot completes its task successfully.

Path planning refers to the navigation of a mobile robot to a target location while satisfying certain constraints (e.g., path length, path smoothness, path safety, etc.). There are many related studies on path planning methods for mobile robots (Sood and Panchal 2020; Patle et al 2019; Zhang et al

2018), however, classical methods require clear information about the environment and are computationally overloaded. Researchers have therefore turned to algorithms of computational intelligence applied to path planning problems. Xu et al (2020) proposed an artificial bee colony algorithm introducing a co-evolutionary framework for path planning of mobile robots, which improves the convergence speed and avoids dimensionality dependence. Zhao et al (2020) proposed a co-optimization of a multi-objective Cauchy mutation cat colony optimization and an artificial potential field method to solve the path planning problem of an intelligent patrol car navigation system. Song et al (2021) solved the problem of local optimum and premature convergence by an improved PSO algorithm that combines continuous high-degree Bezier curves to plan the smooth path of the robot.

However, due to the NFL principle (Wolpert and Macready 1997), each intelligent optimization algorithm has its advantages and disadvantages, and no particular algorithm is superior for all types of problems. Therefore, it is necessary to develop new intelligent optimization algorithms to be applied to path planning problems. Further, the study of local path planning problems in uncertain environments is very limited. The global path planning approach cannot be used when the robot does not have information about the obstacles in its area. Relying solely on sensors to provide

✉ Enhao Zhang
52192325122@2019.cqu.edu.cn

¹ School of Artificial Intelligence, Chongqing University of Technology, Pufu Street, Chongqing 401135, Chongqing, China

information about nearby obstacles to accomplish the task of safely reaching the target is challenging. Mohanty (2020) proposed intelligent cuckoo search algorithm for navigation in an unknown environment and successfully ported to a real robot. Pattnaik et al (2021) proposed a hybrid particle swarm and chemical reaction optimization algorithm and used it to plan a smooth path for a robot in an unknown environment with circular and/or polygonal obstacles.

The sparrow search algorithm (SSA) is a swarm intelligence algorithm developed by Xue and Shen (2020) and has been successfully applied in several fields of research (Ouyang et al 2021; Tang et al 2021; Liu et al 2021; Tuerxun et al 2021). In this paper, a local path planning algorithm is developed for a mobile robot using an improved SSA method. In this paper, SSA is improved with FDB (Kahraman et al 2020) selection method and HHO (Heidari et al 2019) algorithm, which enhances the diversity of SSA populations and balances the exploration and exploitation of the algorithm. In the local path planning problem, the environment is uncertain for the robot, i.e., the robot has no information about the presence of obstacles in the environment and the shape, size, and location of the obstacles. Under such conditions, the robot relies only on the perception of obstacles over a limited distance to calculate the distance to the nearest obstacle, and thus the algorithm is used to avoid the obstacle and navigate safely to the target area. In order to solve the local path planning problem, a fitness function is established considering the two objectives of obstacle avoidance and access to the target region as the objective function of the proposed algorithm. The performance of the proposed algorithm is tested under the Congress on Evolutionary Computation 2017 (CEC2017) function set (Wu et al 2017), and the results are compared with those of PSO and SSA to demonstrate the effectiveness of the algorithm. To explain the results of this study in more detail, the Wilcoxon signed-rank test was also included to verify the superiority of this algorithm. The results show that the proposed ISSA algorithm has better performance and stability on most of the benchmark functions. In addition, the proposed ISSA algorithm is applied to complete the local path planning task in a simulated complex and uncertain scenario, and the results obtained are compared with those of other algorithms to demonstrate the applicability and effectiveness of the algorithm for the local path planning problem.

The remainder of this paper is framed as follows: in Sect. 2, the theory of SSA and the concept of ISSA algorithm are

discussed. Section 3 presents the formulation of the path planning problem and the implementation of ISSA. The simulation results and discussion are explained in Sect. 4. The last section concludes the whole paper.

2 Sparrow search algorithm

SSA is a new swarm intelligence algorithm proposed in 2019, which has the advantages of few adjustable parameters, high solution efficiency and strong local search capability.

In the basic SSA algorithm, the sparrow population is divided into two roles, i.e., producers and scroungers. Producers have higher energy reserves, i.e. they have better fitness values, which allow them to search over a wider range. On the other hand, scroungers occupy poorer positions, so they follow the producers who can find better food sources and search around them to find better positions. When the scrounger finds a better food source, that is, occupies a better position, it becomes the new producer. The duties performed by individual sparrows can vary, but the ratio of producers to scroungers is fixed throughout the population. In addition, when an individual in the sparrow population senses danger, it alerts the rest of the population. Individuals at the edge of the population move quickly toward safety, and individuals in the center of the population move toward other individuals. The standard SSA procedure is described as follows:

1. Initialization Suppose a population of size n is searched in a D -dimensional space. The initialization formula of the population can be expressed as:

$$X_{i,j} = X_j^{min} + rand * (X_j^{max} - X_j^{min}) \quad (1)$$

Where $i = 1, 2, \dots, n$, $j = 1, 2, \dots, D$. $rand$ is a random number in the range of $(0, 1)$. X_j^{max} and X_j^{min} refer to the upper and lower bounds for j^{th} dimension in the space respectively.

2. Producers phase During the implementation of the algorithm, the proportion of discoverers is generally set to 10 – 20% of the population size. The formula for producer update position is as follows:

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^t \cdot \exp\left(\frac{-i}{\alpha \cdot T}\right) & \text{if } R_2 \leq ST \\ X_{ij}^t + Q \cdot L & \text{if } R_2 > ST \end{cases} \quad (2)$$

Where t refers to the current iteration, $\alpha \in (0, 1]$ is a random number. $R_2 \in [0, 1]$, $ST \in [0.5, 1]$ are the alarm value and the safety threshold respectively. Q is a random number obeys normal distribution. L is a vector in which all elements are 1. If $R_2 \leq ST$, the population receives no danger signals and producers are free to search over a wide area. When $R_2 > ST$, otherwise, the population senses the threat of predators, so they search more cautiously.

3. Scroungers phase The individuals in the population with poor fitness values are called scroungers. Their renewal can be described as follows:

$$X_{ij}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^t - X_{ij}^t}{i^2}\right) & \text{if } i > n/2 \\ X_p^t + |X_{ij}^t - X_p^t| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \quad (3)$$

where X_p is the best position for the producer currently. X_{worst} denotes the current global worst location. A represents a matrix of $1 * d$ for which each element inside is randomly assigned 1 or -1 , and $A^+ = A^T(AA^T)^{-1}$. $i > n/2$ means that the individual is in a very poor position and cannot find a food source. Conversely, when $i \leq n/2$, the individual followed the producer to forage.

4. Anti-predator phase It is generally assumed that 10-20% of the individuals in the population perceive the danger and update the location of the anti-predators using the following equation:

$$X_{ij}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{ij}^t - X_{best}^t| & f_i > f_g \\ X_{ij}^t + K \cdot \left(\frac{|X_{ij}^t - X_{worst}^t|}{(f_i - f_{worst}) + \epsilon}\right) & f_i = f_g \end{cases} \quad (4)$$

where X_{best} is the current global optimal location. β is the step control parameter that obeys a normal distribution. K is a random step control parameter in $[-1, 1]$. f_i represents the fitness value of sparrow i . f_g and f_w denote the current global best and worst fitness values, respectively. ϵ is a constant used to avoid the denominator being zero. If $f_i > f_g$, i.e., the individual's position is at the edge of the population, indicating that it is vulnerable to predation and needs to move to the center of the population. Otherwise, the individual moved closer to other individuals to stay safe.

Algorithm 1 shows the detailed process of standard SSA.

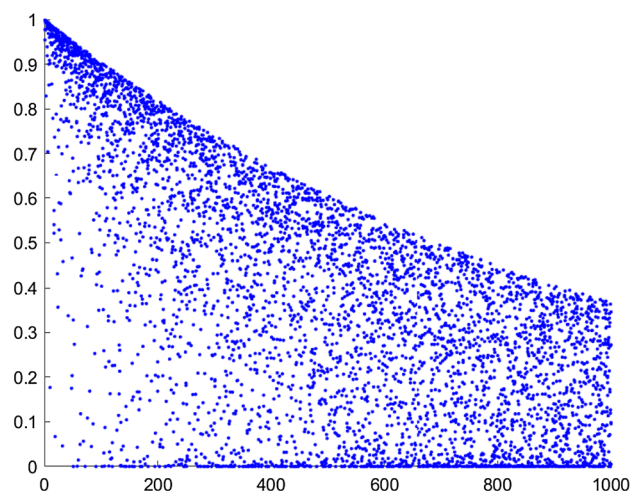


Fig. 1 $y = \exp\left(\frac{-i}{\alpha \cdot T}\right)$

Algorithm 1 Pseudo-code of standard SSA

Input:

- T : the maximum iterations
- pop : the size of population
- NP : the number of producers
- NA : the number of anti-predators

Output: X_{best}, f_{best}

```

Initialize a population;
while t < T do
    Sort the population by fitness values and
    find the current best and worst individual;
    for i = 1 : NP do
        update the producers' location with
        Equation (2);
    end for
    for i = (NP + 1) : n do
        update the scroungers' location with
        Equation (3);
    end for
    for j = 1 : n do
        update the anti-predators' location with
        Equation (4);
    end for
    t = t + 1;
end while
return Xbest, fbest
    
```

3 ISSA

Correspondingly, the basic SSA faces the problems of decreasing population diversity in the late iteration, inability to jump out of local optimum, and susceptibility to structural bias (Ong et al 2021; Caraffini et al 2019). This chapter explains the proposed ISSA algorithm in detail, which involves the producer update formulation for enhanced global search, the FDB selection method, and the Harris Hawk algorithm (HHO) based progressive search.

3.1 Enhanced global search for producer update operator

In the standard SSA, producers are getting smaller coordinate values for each dimension at $R_2 < ST$, as shown in Fig. 1.

This is clearly detrimental to the global exploration of the population. To solve this problem, the following formula is used in this paper to replace the update formula of the basic SSA:

$$X_{ij}^{t+1} = X_{v,j}^{t+1} - r * (X_{v,j}^t - X_{i,j}^t) \quad (5)$$

where $X_{v,j}^t$ is a randomly selected individual in the population; r is a random number within $[0, 1]$. The performance of this operator in two dimensions is shown in Fig. 2. As can be seen from the figure, the producer randomly selected an individual and explored randomly within a certain range.

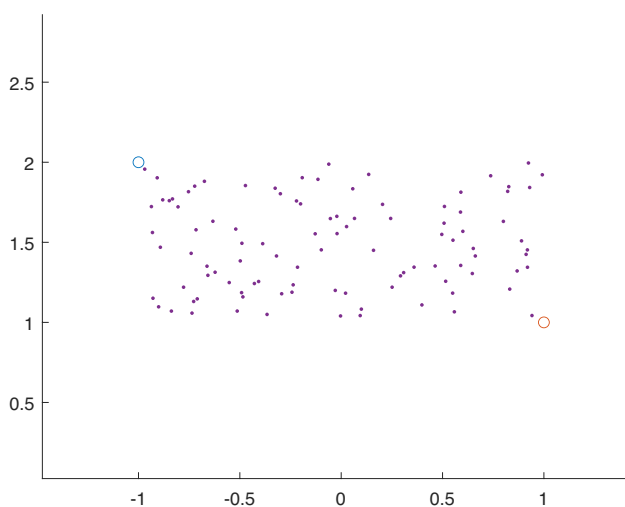


Fig. 2 Demonstration of the effect of improved operator. The two hollow circles represent $X_{i,j}^t$ and $X_{v,j}^t$, respectively, and the solid points indicate the new positions obtained using the proposed operator

The formula describes the random exploration behavior of the producers when no danger is received and enhances the global exploration capability of the algorithm.

3.2 Scrounger update method based on FDB selection method

The scroungers at the edge of the population are repositioned near the origin, which is factually incorrect and loses information from the previous iterations. On the other hand, the operation of jumping to the origin tends to make the algorithm structurally biased, i.e., the search process is very slow when the global optimal solution is not at the origin (Kononova et al 2015). In order to maintain the population diversity without losing the previous information, the Fitness-distance balance (FDB) selection method is introduced in this paper.

Selection methods play an important role in the convergence process of intelligent optimization algorithms. In the search process, the candidate solutions selected by the algorithm from the population directly affect the direction of individual convergence and the success rate of convergence. In order to solve the problem of premature convergence in the algorithm optimization process, Kahraman et al. developed a novel selection method based on fitness and distance balance named FDB selection method in 2020. The method can select the candidate solution with the maximum potential for improving the algorithmic optimization process and indicate the convergence direction for the algorithmic population. Since the introduction of this method, several studies have applied the FDB selection method to various algorithms to solve practical problems. Sefa Aras et al. applied the FDB selection method to fractal search algorithms for improving the diversity of populations and balancing the problem of algorithm development and exploration (Aras et al 2021). The literature (Duman et al 2021a) proposed a coyote optimization algorithm that more closely follows the laws of nature, using the FDB selection method and levy flight to determine the social tendencies of coyote populations, and improved the global exploration and local exploitation capabilities of the algorithm to solve the problem of premature convergence. The improved algorithm is applied to the optimal power flow problem and shows absolute advantages. To solve the optimal power flow problem, Ugur Guvenc et al. used the FDB method to balance the exploration and exploitation of the algorithm by guiding the choice of reference locations in the adaptive guided differential evolution algorithm (Guvenc et al 2021).

The FDB selection method can find the candidate solution that contributes most to the search process in a population in a stable and efficient manner. It is implemented as follows:

In the first step, the Euclidean distance of each individual from the optimal solution is calculated:

$$\forall i \in n, X_i \neq X_{best}, dist_{X_i} = \sqrt{(X_{i,1} - X_{best,1})^2 + \dots + (X_{i,D} - X_{best,D})^2} \tag{6}$$

In the second step, the FDB score is calculated for each individual:

$$\forall i \in n, S_i = \omega * normf_i + (1 - \omega) * normdist_{X_i} \tag{7}$$

where $\omega \in (0, 1)$ is the weighting factor. $normf_i$ and $normdist_{X_i}$ are the normalized fitness value and normalized distance value of X_i , respectively.

The selection probability of each individual is calculated based on the FDB score, and then a roulette wheel method is used to select the reference position(Aras et al 2021). After determining the reference location in the above manner, the new scrounger update can be described as follows:

$$X_i^{t+1} = X_v^t + k * r * (X_i^t - X_{fdb}^t) \tag{8}$$

where X_v^t is a randomly selected individual in the population; k is a predetermined scale factor; r is a [0, 1] uniform random number; and X_{fdb}^t is an individual selected by FDB method. The ability of the sparrow population to explore globally and jump out of the local optimum trap is improved after the roulette FDB-based update operator.

3.3 Harris Hawks based progressive search

The individuals with better fitness values among the scroungers in the basic SSA approach the optimal position in all dimensions, which leads to fast convergence of the population. However, the diversity of the population is thus reduced, leading the algorithm to fall into the local optimum trap easily.

Inspired by Hard besiege with progressive rapid dives of Harris Hawks optimization(HHO) algorithm, a progressive search is introduced to make scroungers move more reasonable and jump out of the local optimal solution without losing their local exploitation capabilities. The formula for the progressive search can be described by the following equation:

$$X_i^{t+1} = \begin{cases} X_{new}^1 & \text{if } f(X_{new}^1) < f(X_i^t) \\ X_{new}^2 & \text{if } f(X_{new}^2) < f(X_i^t) \end{cases} \tag{9}$$

where

$$X_{new}^1 = X_{best}^t - E * (J * X_{best}^t - X_i^t) \tag{10}$$

$$X_{new}^2 = X_{new}^1 + levy(D) \tag{11}$$

where $E = 2E_0(1 - t/T)$, $E_0 \in [-1, 1]$ is a random number, $J = 2(1 - rand)$, D means dimensions, $levy$ refers to levy flight, and can be described as follows:

$$levy(x) = 0.01 * \mu * \sigma / |v|^{1/\beta} \tag{12}$$

where

$$\sigma = \left(\frac{\Gamma(1 + \beta) * \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) * \beta * 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}} \tag{13}$$

where μ, v are random number in [0, 1], β is set to 1.5.

Levy flight is a random walk whose step lengths obey the levy distribution, and its movement is shown in Fig. 3. In the progressive search, two parameters, E and J , are used to incorporate the randomness of the search, while levy flight matches the movement of birds foraging, perfectly simulating the behavior of scroungers foraging near the producer.

3.4 ISSA

The sparrow search algorithm is a newer swarm intelligence optimization algorithm with better search performance, but still has disadvantages such as premature convergence, decreasing population diversity, falling into local traps, and structural bias. To improve these problems, this paper proposes a producer update approach with enhanced global capability, and a scrounger update formulation with FDB selection-based approach and progressive search. These approaches can increase the diversity of sparrow populations, balance exploration and exploitation capabilities, and further improve algorithm performance. The proposed ISSA algorithm is shown in Algorithm 2.

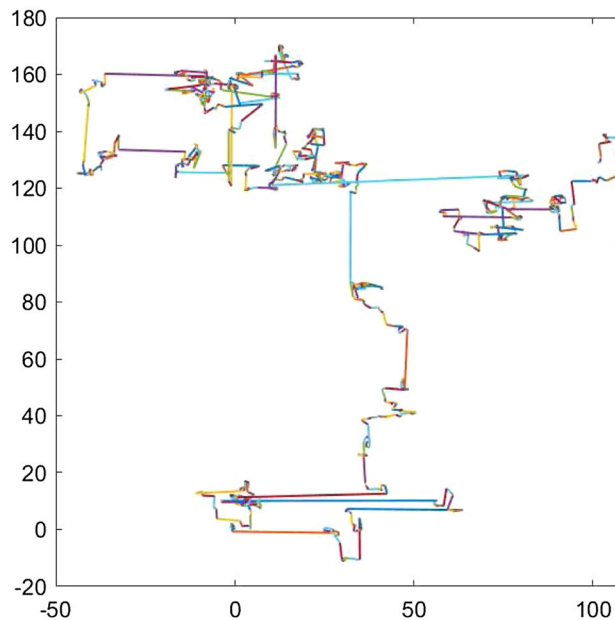


Fig. 3 Levy flight

Algorithm 2 Pseudo-code of proposed ISSA**Input:**

T :the maximum iterations
 pop :the size of population
 NP :the number of producers
 NA :the number of anti-predators

Output: X_{best}, f_{best}

Initialize a population;
while $t < T$ **do**
 Sort the population by fitness values and find the current best and worst individual;
 for $i = 1 : NP$ **do**
 update the producers' location with Equation (2) and Equation (5);
 end for
 for $i = (NP + 1) : n$ **do**
 update the scroungers' location with Equation (8) and Equation (9);
 end for
 for $j = 1 : n$ **do**
 update the anti-predators' location with Equation (4);
 end for
 $t = t + 1$;
end while
return X_{best}, f_{best}

4 Formulation of path planning problem

This section describes the process and structure of solving local path planning problems with ISSA. In the local path planning problem, the robot does not know the global environment information and only performs obstacle avoidance based on a limited sensing range. When there are obstacles nearby, the robot needs to maintain a safe distance from the nearest obstacle. Considering the robot's drive-to-target behavior and keep-safe behavior, the direction of each robot action can be determined.

4.1 Keep-safe behavior

Let d_{obs} be the distance between the robot and the obstacle, X be the position of the robot, and X_{obs}^i be the position of obstacle i . Then the distance between the robot and the nearest obstacle can be calculated by the Euclidean distance as follows:

$$d_{obs} = \min \|X - X_{obs}^i\| \quad (14)$$

The robot should keep the maximum safe distance from the nearest obstacle to avoid collision with the obstacle. When $d_{obs} \leq 0$, the adaptation degree should be infinite; when $d_{obs} \leq d_{safe}$ (safe distance), the larger d_{obs} is, the lower the adaptation degree value should be; when $d_{obs} > d_{safe}$, it means that the robot is in a safe position and does not need to avoid the obstacle. The specific expression is as follows:

$$f_1 = \begin{cases} \inf & \text{if } d_{obs} \leq 0 \\ \frac{1}{d_{obs}} & \text{if } 0 < d_{obs} \leq d_{safe} \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

4.2 Drive-to-target behavior

In this behavior, the robot needs to reach the goal along the shortest possible path. This goal can be expressed by the following equation:

$$f_2 = d_{goal} = \|X - goal\| \quad (16)$$

4.3 Fitness function

In order to develop an objective function to combine the above two objectives, this paper uses the weighted sum method to combine them into a fitness function applicable to the path planning problem:

$$f = w_1 * f_1 + w_2 * f_2 \quad (17)$$

where w_1, w_2 are weighting factors that can be adjusted according to the problem.

4.4 Implementing local path planning with ISSA

The robot starts from the starting position and advances towards the target area. After each move, it searches whether there is any obstacle in the vicinity, and if not, it continues to move in the target direction; otherwise, it calls the ISSA algorithm to find the direction of the next move. The above process is continued until the robot reaches the target position. The following are the specific steps:

Step 1: Initialize the environment and the starting and ending points of the robot.

Step 2: Detect if there are obstacles around, and if not, advance to $[x_{new}, y_{new}]$ in the direction of the target.

$$\begin{aligned} x_{new} &= x + stepsize * \cos \theta_{goal} \\ y_{new} &= y + stepsize * \sin \theta_{goal} \end{aligned} \quad (18)$$

Step 3: If an obstacle is detected, the ISSA algorithm is invoked to find the direction θ with optimal adaptation and proceed in this direction:

$$\begin{aligned}x_{new} &= x + \text{stepsize} * \cos \theta \\y_{new} &= y + \text{stepsize} * \sin \theta\end{aligned}\quad (19)$$

Step 4: Repeat step 2 or step 3 continuously according to the detection result until the target position is reached.

5 Experiments and discussion

In order to verify the performance of the proposed path planning algorithm, benchmark function experiments and simulation experiments were implemented. The experimental environment is a computer with windows operating system, i5 processor and 16GB RAM.

5.1 CEC2017 benchmark test

In this experiment, the performance of the proposed algorithm is tested on the CEC2017 function set. The obtained results are compared with several intelligent optimization algorithms such as basic SSA, Particle swarm optimization(PSO)(Duman et al 2021a), Harris hawks optimization(HHO), Whale optimization algorithm(WOA) (Guvenc et al 2021), and Moth-flame optimization(MFO) (Duman et al 2021b). cec2017 benchmark functions can be classified into four categories: single-mode (F1-3), multimode (F4-10), hybrid (F11-20), and composition (F21-30). To be fair, all algorithms are implemented in the same test environment. In this experiment, the problem dimension, population size and maximum number of computations (maxFES) are set to 10,30 and 100000, respectively. Since the swarm intelligence algorithm is randomized, each experiment is repeated 30 times in order to avoid the error brought by randomness on the experimental results. The experimental results are shown in the Table 1, and some of the convergence curves are shown in Fig. 4.

As can be seen from the table, the ISSA algorithm proposed in this paper is the best in dealing with unimodal problems, such as F1-3. This is mainly due to the superiority of the proposed algorithm in terms of local search performance. ISSA is ahead of other algorithms in most of the multimodal problems and second only to PSO algorithm in a few functions. The multimodal problems mainly test the algorithm's ability to jump out of local optimum and global search. This can be seen in the ability to jump out of local traps brought by the progressive foraging added by ISSA and the global search ability enhanced by the discoverer improvement. As for the hybrid problem, the proposed algorithm has a large advantage on most functions, which

fully demonstrates ISSA's ability in balanced exploration and development. Finally, it can be seen in the results of the combinatorial problems that ISSA's operators have stable and excellent performance on most of the problems.

Table 1 lists the error statistics of the fitness values obtained by ISSA and some common algorithms in the test functions of the CEC2017 benchmark suite. To better understand and interpret the data in the table, the corresponding box-line plots are plotted below, as in Fig. 5. a test function is selected from four problem types: single-mode, multimode, hybrid, and combined, whose box-plots are shown below.

As can be seen from the box-plots, the proposed ISSA algorithm shows a more stable search performance in the CEC2017 test function and can converge to smaller error values. Moreover, the improved algorithm greatly reduces the probability of falling into a local optimum solution, and the premature convergence problem is significantly improved. In conclusion, according to the data shown in the box-plots, ISSA effectively improves the global exploration ability and local exploitation ability of the algorithm, and handles the balance between exploration and exploitation well, and solves the prematureness problem of the algorithm.

To more accurately assess the superiority of the proposed algorithm, the Wilcoxon signed-rank test was used to examine the statistical differences in the algorithm results. The obtained results are shown in Table 2.

From the data in the table, it is clear that the proposed ISSA algorithm is statistically significant compared to several other compared algorithms. This further proves the superiority of ISSA.

The complexity of an algorithm is an important factor that affects the usability and functionality of the algorithm. Some algorithms can converge to a relatively good level, but if the response time is too long, they cannot be called a fully optimized algorithm. Therefore, this paper will give the algorithm complexity of the tested algorithms, and discuss and analyze the experimental results (Table 3).

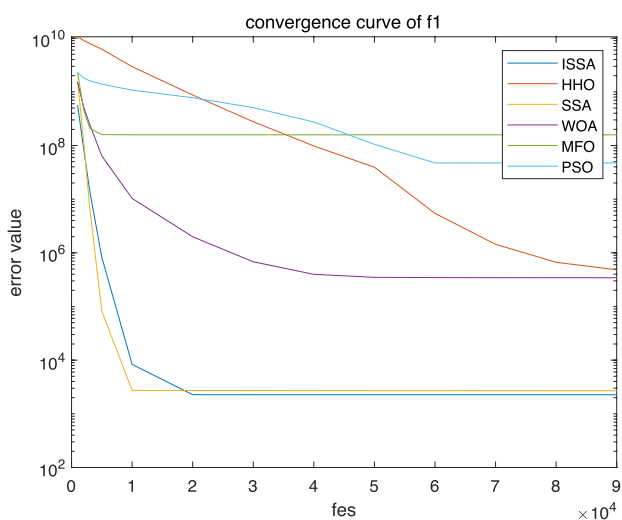
To calculate the complexity of the algorithm, this paper refers to the definition document of IEEE CEC 2017 and defines the algorithm complexity as follows.

$$\text{complexity} = (\text{mean}(T2) - T1)/T0 \quad (20)$$

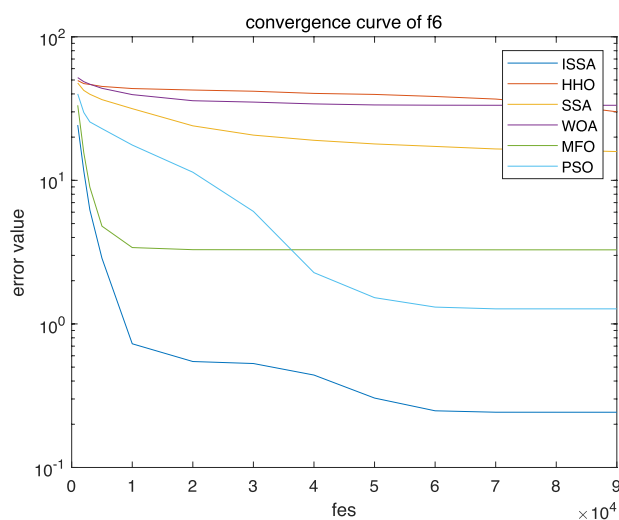
Where the parameter T0 represents the algorithm calculation time for the test procedure specifically defined in CEC 2017, T1 indicates the time required to consume 200000 times for the evaluation of the function F18 alone, F18 is the test problem defined in CEC 2017. t2 is the time required to calculate the function F18 using the algorithm, and the test is repeated 5 times and then averaged. The experimental results are shown in the following table.

Table 1 CEC2017 test results

Avg	PSO	HHO	WOA	MFO	SSA	ISSA
F1	4.73E+07	4.83E+05	3.44E+05	1.58E+08	2.69E+03	2.27E+03
F2	5.15E+08	1.22E+02	8.77E+03	7.05E+09	9.92E-05	7.24E-05
F3	0.00E+00	1.92E+00	3.97E+02	9.57E+03	0.00E+00	0.00E+00
F4	2.19E+01	1.58E+01	2.04E+01	6.08E+00	1.95E+00	4.40E+00
F5	1.86E+01	4.78E+01	5.43E+01	3.31E+01	3.92E+01	1.98E+01
F6	1.27E+00	3.00E+01	3.33E+01	3.28E+00	1.59E+01	2.42E-01
F7	1.88E+01	8.23E+01	7.66E+01	4.37E+01	9.11E+01	3.26E+01
F8	1.66E+01	3.14E+01	3.78E+01	3.25E+01	3.15E+01	1.57E+01
F9	5.99E+00	4.97E+02	3.79E+02	1.42E+02	7.12E+02	5.26E+00
F10	4.62E+02	9.32E+02	1.06E+03	9.66E+02	1.03E+03	7.29E+02
F11	4.55E+01	7.15E+01	1.28E+02	1.58E+02	4.90E+01	2.44E+01
F12	2.98E+05	2.12E+06	5.38E+06	4.07E+05	1.81E+04	1.58E+04
F13	7.86E+03	1.29E+04	1.40E+04	1.24E+04	9.12E+03	7.61E+03
F14	7.05E+01	1.22E+02	2.61E+02	5.24E+03	1.21E+02	7.15E+01
F15	1.59E+02	2.07E+03	3.70E+03	6.10E+03	2.10E+02	1.10E+02
F16	7.79E+01	2.96E+02	2.72E+02	1.56E+02	2.89E+02	6.38E+01
F17	7.05E+01	7.75E+01	8.92E+01	6.66E+01	1.10E+02	4.75E+01
F18	1.31E+04	1.35E+04	1.74E+04	2.10E+04	5.23E+03	4.57E+03
F19	1.51E+02	7.81E+03	2.71E+04	9.35E+03	3.76E+02	1.90E+02
F20	4.92E+01	1.83E+02	1.54E+02	8.34E+01	7.61E+01	3.00E+01
F21	2.13E+02	2.15E+02	1.99E+02	2.15E+02	1.74E+02	1.10E+02
F22	1.48E+02	1.99E+02	1.81E+02	1.06E+02	1.09E+02	9.76E+01
F23	3.33E+02	3.79E+02	3.55E+02	3.29E+02	3.48E+02	3.20E+02
F24	3.50E+02	4.17E+02	3.68E+02	3.62E+02	3.54E+02	3.44E+02
F25	4.34E+02	4.37E+02	4.44E+02	4.38E+02	4.33E+02	4.28E+02
F26	6.66E+02	9.33E+02	7.50E+02	4.93E+02	7.99E+02	3.68E+02
F27	4.25E+02	4.55E+02	4.37E+02	3.94E+02	4.10E+02	3.94E+02
F28	6.03E+02	5.51E+02	5.87E+02	4.93E+02	5.44E+02	5.15E+02
F29	3.17E+02	3.93E+02	4.23E+02	3.38E+02	3.91E+02	3.04E+02
F30	5.74E+05	3.68E+05	7.07E+05	5.36E+05	2.80E+05	3.45E+05

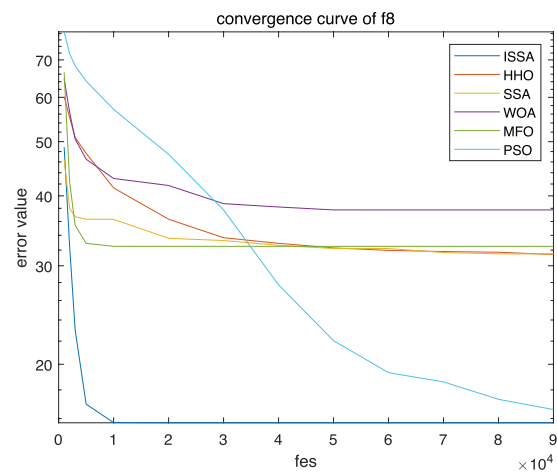


(a) f1

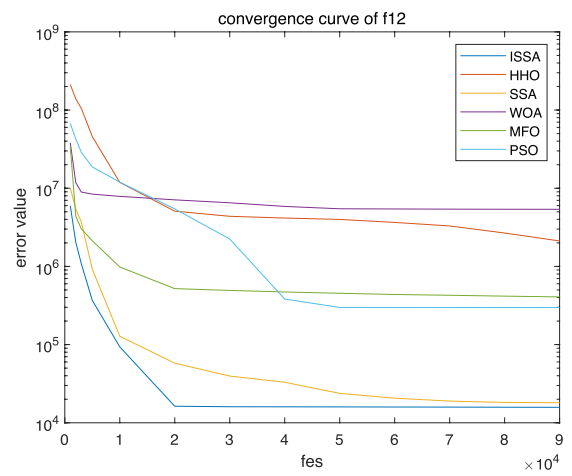


(b) f6

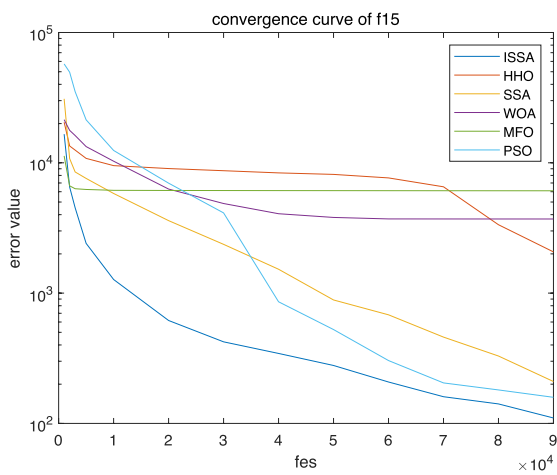
Fig. 4 A part of convergence curve results



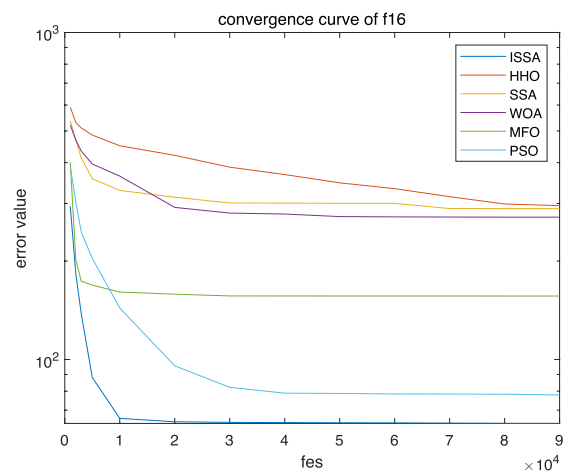
(a) f8



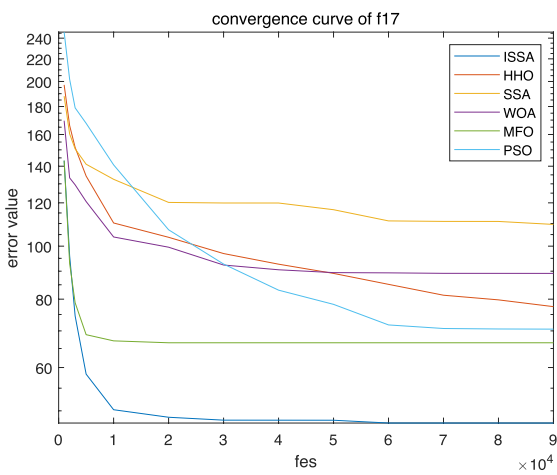
(b) f12



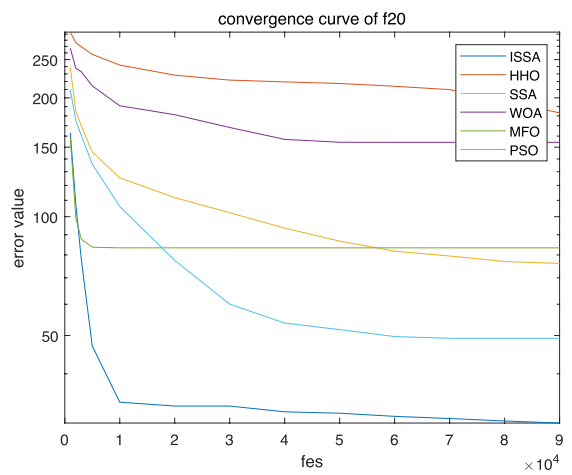
(c) f15



(d) f16

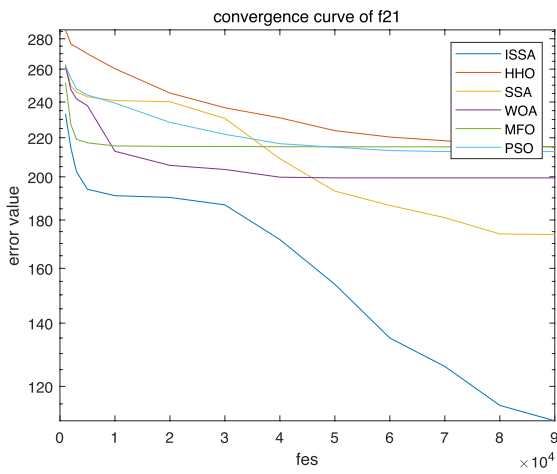


(e) f17

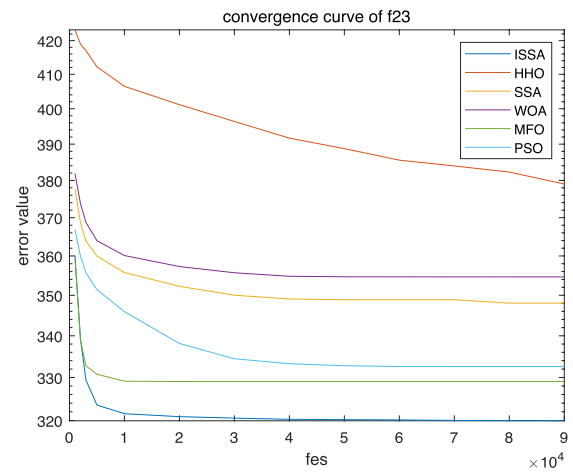


(f) f20

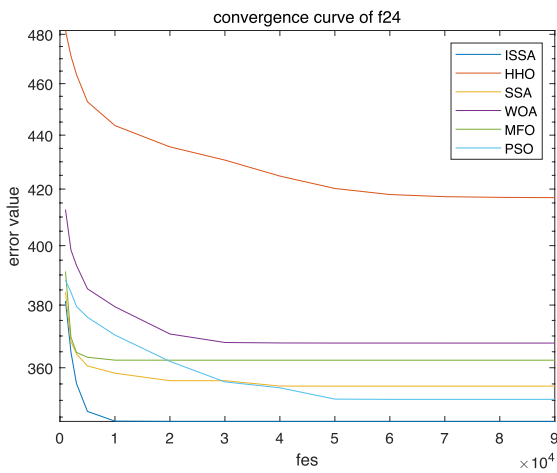
Fig. 4 (continued)



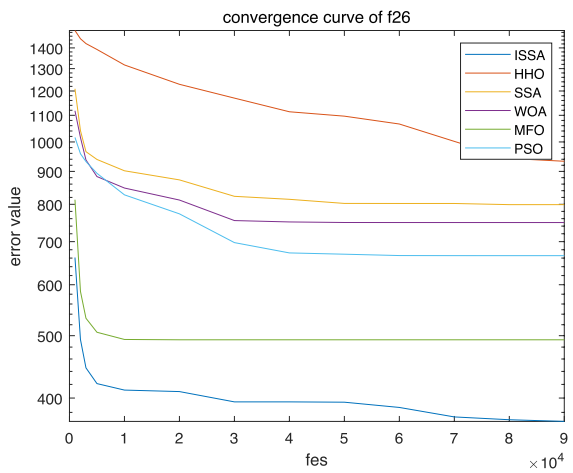
(a) f21



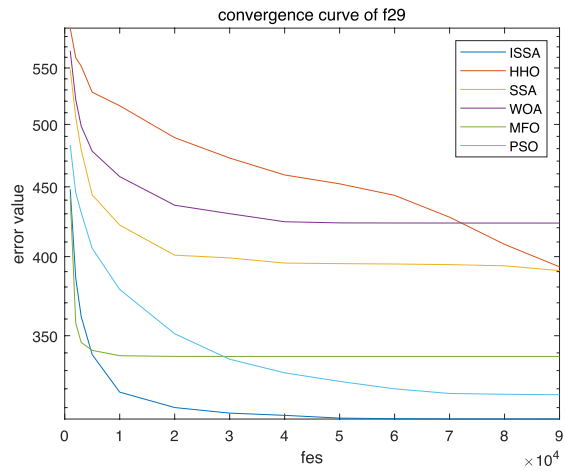
(b) f23



(c) f24



(d) f26



(e) f29

Fig. 4 (continued)

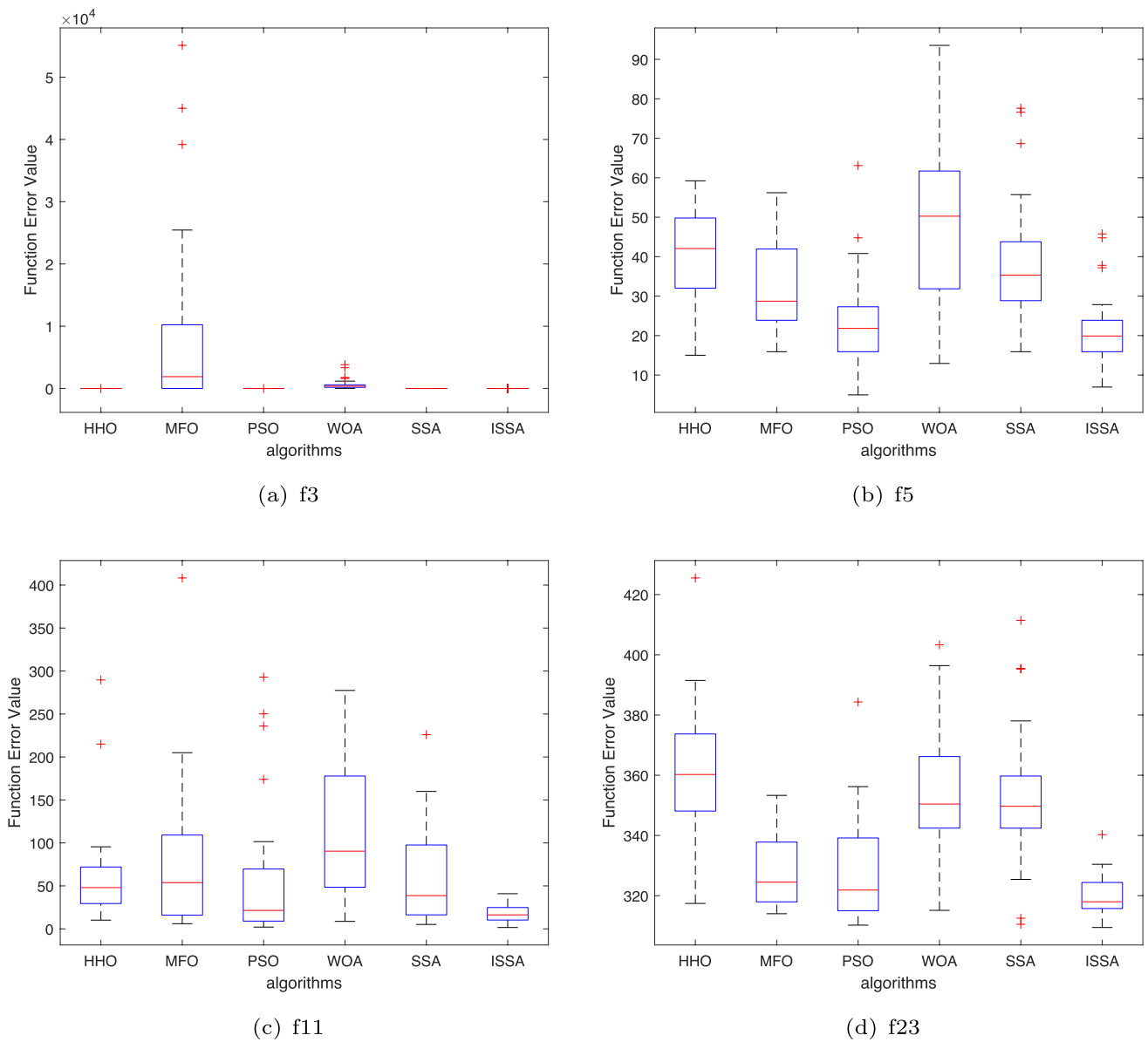


Fig. 5 A part of box-plot results

From the data in the table, we can see that the MFO algorithm performs well in the index of time complexity, the complexity of the sparrow search algorithm is 55.1923, and the complexity of the improved ISSA algorithm is higher, but within an acceptable range. The reason for the high time complexity is that the FDB selection method does not find a suitable programming method, which causes each individual to calculate the distance to the rest of the population in each iteration, and then consumes too much time. After temporarily discarding the FDB selection method, the time complexity of ISSA comes to a good magnitude.

5.2 Simulation experiments

To verify the feasibility and practicality of the proposed algorithm, an uncertain environment consisting of circular and polygonal obstacles is considered in this paper, as shown in Fig. 6. The starting point [30 30] and the ending point [450 450] are set in the complex scenario to test the feasibility and robustness of the algorithm on the local path planning problem. The proposed ISSA algorithm is experimented with SSA and PSO in the same scenarios and the obtained results are compared. Since the intelligent optimization algorithm is stochastic, the execution

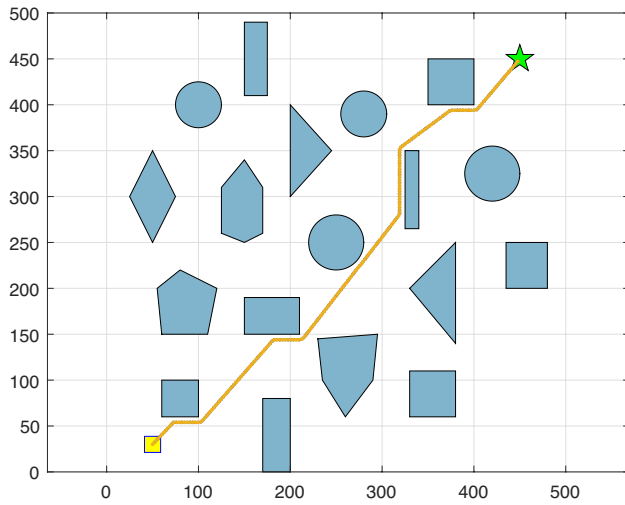


Fig. 6 Map with circular and polygonal obstacles

Table 2 Wilcoxon signed rank test results

Comparison	p	h
PSO	6.10E-04	1
HHO	1.73E-06	1
WOA	1.86E-06	1
MFO	6.34E-06	1
SSA	5.51E-05	1

Table 3 complexity test results

Algorithm	Complexity
T0	0.0152
T1	0.3683
PSO	78.1042
HHO	113.4969
WOA	25.7491
MFO	4.1267
SSA	55.1923
ISSA	117.6780/62.1127(No FDB)

is repeated 10 times to obtain the average path length, standard deviation of path length, average path smoothness and average computation time. Where the path length can be expressed by the following equation:

$$length = \sum_{i=1}^n dist(p_i, p_{i-1}) \tag{21}$$

where $dist()$ denotes the Euclidean distance between two points and n denotes the total number of points that form

Table 4 simulation results

Algorithm	ISSA	SSA	PSO
Average path length	6.30E+02	6.41E+02	6.31E+02
Standard deviation of path length	4.10E-13	4.07E+00	1.20E+00
Average path smoothness	6.53E+01	8.36E+01	7.16E+01
Average computation time	1.71E+01	1.26E+01	1.31E+01

the path. The path smoothness can be expressed as the sum of the deflection angles between adjacent points as follows:

$$smoothness = \sum_{i=1}^n |\theta_i| \tag{22}$$

where θ_i denotes the steering angle between two adjacent vectors in the path.

The results of the test are shown in Table 4.

From the table 4, it can be seen that the ISSA algorithm can obtain paths with shorter length and better smoothness compared to SSA and PSO. However, the proposed ISSA algorithm is not as good as the other algorithms in terms of average computation time. This may be due to the fact that the multiple operators introduced by ISSA lead to more operational steps required in the iteration of the algorithm, so that more time is consumed. Overall, the performance of the ISSA algorithm is excellent and robust for local path planning in uncertain environments compared to the other two algorithms.

6 Conclusion

In this paper, a novel SSA-based intelligent optimization algorithm is proposed for solving local path planning problems in uncertain environments. A new fitness function is formulated to orient the robot to the shortest safe distance. In this paper, complex uncertain scenarios are considered with multiple circular and polygonal obstacles placed in the environment. The results obtained from benchmark function experiments and simulated local path planning experiments show that the performance of the proposed ISSA algorithm outperforms other algorithms.

References

Aras S, Gedikli E, Kahraman HT (2021) A novel stochastic fractal search algorithm with fitness-distance balance for global numerical optimization. *Swarm Evol Comput* 61(100):821

Caraffini F, Kononova AV, Corne D (2019) Infeasibility and structural bias in differential evolution. *Inf Sci* 496:161–179

Duman S, Kahraman HT, Guvenc U et al (2021a) Development of a lévy flight and fdb-based coyote optimization algorithm for

- global optimization and real-world acopf problems. *Soft Comput* 25(8):6577–6617
- Duman S, Kahraman HT, Guvenc U et al (2021b) Development of a lévy flight and fdb-based coyote optimization algorithm for global optimization and real-world acopf problems. *Soft Comput* 25(8):6577–6617
- Guvenc U, Duman S, Kahraman HT et al (2021) Fitness-distance balance based adaptive guided differential evolution algorithm for security-constrained optimal power flow problem incorporating renewable energy sources. *Appl Soft Comput* 108(107):421
- Heidari AA, Mirjalili S, Faris H et al (2019) Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 97:849–872
- Hy Z, Wm L, Ax C (2018) Path planning for the mobile robot: a review. *Symmetry* 10(10):450
- Kahraman HT, Aras S, Gedikli E (2020) Fitness-distance balance (fdb): a new selection method for meta-heuristic search algorithms. *Knowl-Based Syst* 190(105):169
- Kononova AV, Corne DW, De Wilde P et al (2015) Structural bias in population-based algorithms. *Inf Sci* 298:468–490
- Liu G, Shu C, Liang Z et al (2021) A modified sparrow search algorithm with application in 3D route planning for UAV. *Sensors* 21(4):1224
- Mohanty PK (2020) An intelligent navigational strategy for mobile robots in uncertain environments using smart cuckoo search algorithm. *J Ambient Intell Humaniz Comput* 11(12):6387–6402
- Ong KM, Ong P, Sia CK (2021) A carnivorous plant algorithm for solving global optimization problems. *Appl Soft Comput* 98(106):833
- Ouyang C, Qiu Y, Zhu D (2021) Adaptive spiral flying sparrow search algorithm. *Sci Program*. <https://doi.org/10.1155/2021/6505253>
- Patle B, Pandey A, Parhi D et al (2019) A review: on path planning strategies for navigation of mobile robot. *Def Technol* 15(4):582–606
- Pattnaik S, Mishra D, Panda S (2021) A comparative study of meta-heuristics for local path planning of a mobile robot. *Eng Optim*. <https://doi.org/10.1080/0305215X.2020.1858074>
- Song B, Wang Z, Zou L (2021) An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree bezier curve. *Appl Soft Comput* 100(106):960
- Sood M, Panchal VK (2020) Meta-heuristic techniques for path planning: recent trends and advancements. *Int J Intell Syst Technol Appl* 19(1):36–77
- Tang Y, Li C, Li S et al (2021) A fusion crossover mutation sparrow search algorithm. *Math Probl Eng*. <https://doi.org/10.1155/2021/9952606>
- Tuerxun W, Chang X, Hongyu G et al (2021) Fault diagnosis of wind turbines based on a support vector machine optimized by the sparrow search algorithm. *IEEE Access* 9:69307–69315
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
- Wu G, Mallipeddi R, Suganthan PN (2017) Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report
- Xu F, Li H, Pun CM et al (2020) A new global best guided artificial bee colony algorithm with application in robot path planning. *Appl Soft Comput* 88(106):037
- Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. *Syst Sci Control Eng* 8(1):22–34
- Zhao D, Yu H, Fang X et al (2020) A path planning method based on multi-objective cauchy mutation cat swarm optimization algorithm for navigation system of intelligent patrol car. *IEEE Access* 8:788–803

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.