

Article

Application of Improved Butterfly Optimization Algorithm in Mobile Robot Path Planning

Rongjie Zhai ^{1,2}, Ping Xiao ^{1,*}, Da Shu ¹, Yongjiu Sun ³ and Min Jiang ²

¹ School of Mechanical Engineering, Anhui Polytechnic University, Wuhu 241000, China; 2200120174@stu.ahpu.edu.cn (R.Z.); sd@ahpu.edu.cn (D.S.)

² Institute of Launch Dynamics, Nanjing University of Science and Technology, Nanjing 210094, China; jm_anhui@163.com

³ Department of Robot Design, Anhui Pullen Intelligent Equipment Co., Ltd., Wuhu 241000, China; sunyongjiu1@126.com

* Correspondence: tlxp95@ahpu.edu.cn

Abstract: An improved butterfly optimization algorithm (IBOA) is proposed to overcome the disadvantages, including slow convergence, generation of local optimum solutions, and deadlock phenomenon, of the optimization algorithm in the path planning of mobile robots. A path-planning grid model is established based on an improved obstacle model. First, the population diversity is improved by introducing kent mapping during population position renewal in the normal butterfly optimization algorithm (BOA) to enhance the global search ability of the butterfly population. Second, an adaptive weight coefficient is introduced in the renewal process of each generation to increase the convergence speed and accuracy. An opposition-based learning strategy based on convex lens imaging is introduced to help the butterfly population jump out of the local optimum. Finally, a mutation strategy is introduced to solve the path planning problem. On this basis, two path simplification strategies are proposed to make up for the shortcomings of planning paths in grid maps. The shortest path lengths solved by IBOA, BOA, and GA in the 20×20 map are 30.97, 31.799, and 31.799, respectively. The numbers of iterations for the shortest paths searched by IBOA, BOA, and GA are 14, 24, and 38 in that order. The shortest path lengths solved by IBOA, BOA and GA in the 40×40 map are 63.84, 65.60, and 65.84, respectively. The number of iterations for the shortest paths searched by IBOA, BOA and GA are 32, 40, and 46, respectively. Simulation results show that IBOA has a strong ability to solve robot path planning problems and that the proposed path simplification strategy can effectively reduce the length of the optimal path in the grid map to solve the path planning problem of mobile robots. The shortest paths solved by IBOA in 20×20 and 40×40 maps are simplified to lengths of 30.2914 and 61.03, respectively.

Keywords: improved butterfly optimization algorithm; kent mapping; adaptive weight coefficient; path simplification strategy; path planning



Citation: Zhai, R.; Xiao, P.; Shu, D.; Sun, Y.; Jiang, M. Application of Improved Butterfly Optimization Algorithm in Mobile Robot Path Planning. *Electronics* **2023**, *12*, 3424. <https://doi.org/10.3390/electronics12163424>

Academic Editor: Giuseppe Prencipe

Received: 5 July 2023

Revised: 29 July 2023

Accepted: 9 August 2023

Published: 13 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, mobile robotics has garnered great research interest. Research on mobile robots includes navigation and positioning, motion control, and path planning. Path planning is one of the core topics of mobile robot research. Robot path planning is a technique that enables a robot to find an optimal path from a starting point to a target point in a static environment without colliding with obstacles [1–5].

With the working environment of mobile robots becoming increasingly complex, the requirements of path-planning algorithms are getting higher and higher. The traditional path-planning strategy can not be satisfied with the high demand for robot path planning. In order to adapt to the complex application environment and application requirements of mobile robots, there is an urgent need to design path-planning algorithms with shorter solution paths and faster iterative processes in complex environments.

Different path-planning methods for mobile robots have been proposed by several scholars.

The first step in solving the mobile robot path planning problem is to set up a robot movement map model. Scholars have proposed different research methods to establish an environment model for mobile robots. An environment modeling method that collects and integrates the obstacle information between the starting and the goal through a neural network is proposed [6]. Although modeling the environment model through neural networks is accurate, the integration of information requires extensive training of the neural networks, which may also need to be retrained when performing tasks on new maps. Therefore, this method has a large workload and is suitable only when a high accuracy of the moving position is required, and the robot has been in a stable map environment for a long time. Compared with other environmental modeling methods, the grid method is more convenient in the modeling process, and the storage of each point information is simpler in the operation process, which can help mobile robots read environmental information quickly. Therefore, the grid method is widely used to model the environment in robot path-planning problems [7–11]. The grid method of environment modeling has a wide range of applications in path planning problems, and it can be well combined with meta-heuristic algorithms to solve the path planning problems of mobile robots, so the grid method is used in this study to model the environment. But the obstacle modeling in the grid map still needs further processing.

Different types of algorithms have been introduced to solve the path-planning problem, and they have achieved good results.

An enhanced particle swarm optimization algorithm was used to help mobile robots generate the optimal collision-free path. Simulation and experimentation yielded good results [12]. A method for fitting a smooth robot moving path and a smoothing criterion for the robot path is proposed [13]. In the research, parameter adaptive strategy and greedy strategy are proposed to provide ideas for this research. A novel path-planning algorithm was proposed based on jump point search and Bessel curves. The experimental results showed that the algorithm could achieve an optimal robot motion path between the initial point and the target point [14]. Many algorithms in the field of path planning have been inspired by biological technology, including the ant colony algorithm, bee colony algorithm, genetic algorithm (GA), whale optimization algorithm, dragonfly algorithm, and ant lion optimizer, to solve robot path planning, obstacle avoidance, and path optimization problems [15–19]. The grey wolf optimization (GWO) algorithm has been used to search for the optimal path in the mobile robot path-planning problem. The effect of random parameters on the GWO algorithm was shown to be reduced by calculating the arithmetic mean of the three optimal individuals of the population [20]. The path-planning scheme proposed in the literature is as follows. First, a two-dimensional map is established to simulate the robot's mobile environment; then, the optimal path is calculated by the GWO, and the coordinates of the key points in the optimal path recorded; finally, the key points in the map are connected to generate the robot movement path. This provides a general idea for this study. The storage, computation and updating of path information with different numbers of grids is one of the keys to solving path planning problems by using metaheuristic algorithms.

Among the mobile robot path-planning algorithms proposed by the above scholars, metaheuristic algorithms showed the characteristics of fast solution speed and high solution accuracy. However, the metaheuristic algorithms proposed by some scholars still have some disadvantages for robot path planning, such as problems of low convergence speed and accuracy, which need to be further optimized.

A butterfly optimization algorithm (BOA) was proposed. Through different tests, it was verified that the BOA had a higher solving ability than other advanced metaheuristic algorithms for most problems. The BOA has a more stable solution performance than other algorithms [21]. By introducing kent mapping [22], self-adaptive weight, and adversarial learning strategy, the algorithm is improved from different angles to improve

the convergence speed and convergence accuracy of the BOA. The adaptability of BOA to path planning problems could be improved by introducing a population variation strategy. The improved butterfly optimization algorithm (IBOA) and GA are applied to the path-planning problem, and the path length obtained by the IBOA is shorter and the number of iterations is lower. Two simplification strategies were proposed to optimize the path collaboratively according to the optimal path found by the algorithm, thereby further shortening the path length. In the static obstacle environment, the IBOA and the simplified strategy were applied to the mobile robot path-planning simulation experiment, and good results were obtained, which verified the superiority and feasibility of the algorithm.

The remainder of this paper is organized as follows. Section 2 models the environment through a grid approach and proposes an obstacle extension method to model obstacles. Compared with traditional obstacle models, the proposed obstacle model guarantees that the mobile robot will not collide with an obstacle when it passes over the vertices of the obstacle grid. The traveling safety of the robot is guaranteed with the size of the obstacle model as small as possible. Section 3 presents three ways to improve the BOA. These three methods improve the population diversity, convergence speed, and ability of the algorithm to jump out of the local optimal solution. The solving powers of the BOA and IBOA are compared through six benchmark functions. In comparison to the normal BOA, IBOA showed faster solving speed and higher solving accuracy. In Section 4, IBOA is applied to the mobile robot path-planning problem. The method of storing paths in the form of cell data used in this paper could effectively store the path information with the different number of variables and unify the processing to complete the algorithm solving. Compared with BOA and GA, the shortest path length for the solution of IBOA is shorter and the number of iterations required for the solution of IBOA is smaller. Two methods for simplifying the path were proposed to simplify the optimal path obtained by the algorithm. This simplification addresses the unavoidable problem of planning paths in a grid graph. Section 5 presents conclusions and discussions drawn from the findings. Compared with the path simplification strategy in literature [11], the proposed simplification strategy can be applicable to paths of different sizes and has an overlapping property that allows the successive deletion of redundant points between necessary points.

2. Environment Modeling

Establishing a suitable environment model is critical for mobile robot path planning. The grid method is easy to use and can effectively represent the layout of an environment. Therefore, the grid method is a common environment modeling method for solving robot path-planning problems [23,24].

This environment model adopts complete traversal modeling. The entire environment is first probed, and the coordinate and contour information of all obstacles are recorded in the mobile robot database so that the robot can plan a collision-free and shortest path. Such environment modeling enables the robot to have a basic understanding of global information and find an optimal path in the global environment [25].

In a static environment, the size, number, and position of obstacles are known and do not change during the robot's motion. The planning time of the robot and environmental information is closely related to the size and number of grids. Therefore, in this study, the external circle of the robot projected on the ground was used as the robot in the two-dimensional map, and the diameter of the external circle was D . To plan the route accurately and avoid collisions between the robot and obstacles, the side length of a single grid is taken as D .

Based on this, a grid map was established, as shown in Figure 1. The black grid represents the obstacle module, and the white grid represents the movable area. The lower-left corner S is the starting point, and the upper-right corner E is the target point.

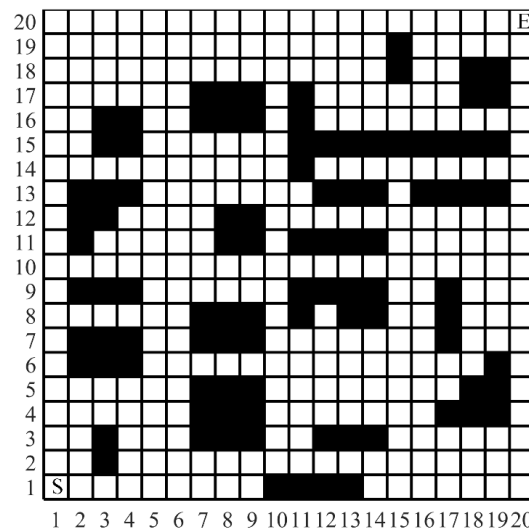


Figure 1. Grid method environment modeling.

As shown in Figure 1, the positive direction of the x-axis is defined from left to right, the positive direction of the y-axis is defined from bottom to top, and the grid length is defined as the unit length used to establish a two-dimensional coordinate plane. The grid is numbered using Equation (1), which is used to plan the movement route of the robot.

$$num = (x - 1) + (y - 1) * x_{max}, \tag{1}$$

where num denotes the number of grids, x denotes the number of columns in the current grid, y denotes the number of rows in the current grid, and x_{max} denotes the total number of columns in the grid map.

The obstacle models were obtained by expanding the real obstacle, as shown in Figure 2, which is a comparison diagram of the preliminary expansion treatment of the obstacle module.

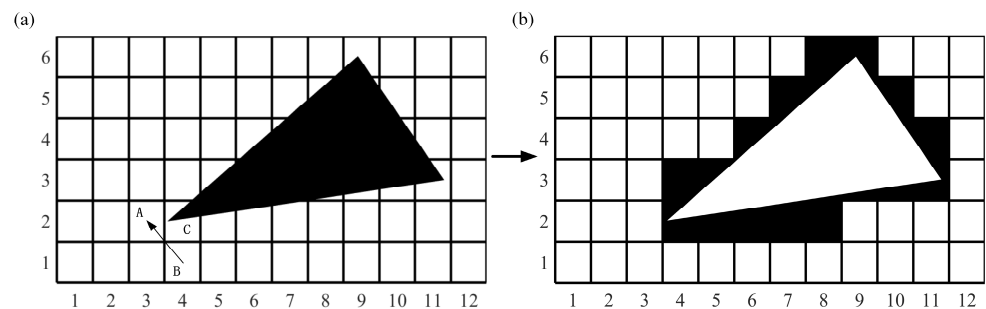


Figure 2. Comparison diagram of preliminary expansion treatment of obstacle models: (a) The initial obstacle model; (b) The obstacle module after the first expansion process.

All grids involved in the obstacle are set to black. When the planned robot path includes the path denoted by $B \rightarrow A$ in Figure 2, the robot may still come into contact with the obstacle module. Therefore, the obstacle model required a second expansion process.

First, the edge grid of the obstacle model must be determined. As shown in Figure 3, if the black grid C is adjacent to two sides connected to two free grids A and B , and the free grids A and B are in the diagonal positions of the 2×2 square grid, it is necessary to determine the relationship between the distance L and $D/2$. L is the distance from the point on the obstacle in the grid to the line connecting the center points of grids A and B .

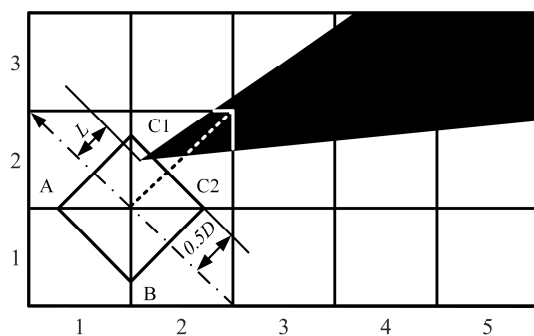


Figure 3. Analysis diagram of the second step expansion processing of obstacles.

As shown in Figure 3, if the distance $\min(L) > D/2$, then the free grid remains blank. Otherwise, the black grid C is divided into two triangular areas, C1 and C2, according to the dashed diagonal line, as shown in Figure 3. The section with $L < D/2$ on the obstacle model will be named the out-of-bounds part, and the free grid that is connected to the triangle area where the out-of-bounds part is located will be changed to a black grid. If the out-of-bounds part is in two triangular areas at the same time, the free grids on the robot motion path that connect the two triangular areas will all become black grids.

As shown in Figure 3, the out-of-bounds part of the obstacle only exists in the C1 triangle area, so grid A needs to be set as a black grid to avoid collision with the obstacle. Code.

3. Improved Butterfly Optimization Algorithm

Each butterfly in the BOA emits a specific fragrance. At the same time, individual butterflies can perceive each other’s fragrances. According to the fragrance concentration, each butterfly mainly renews the position of the butterfly population through two search methods during the search process. Equation (2) for the fragrance concentration of butterflies is as follows:

$$f = cI^a, \tag{2}$$

where f is the odor intensity magnitude, c is the perceptual form, I is the stimulus intensity, and a is the power exponent. The ranges of a and c were $[0, 1]$. In the BOA, a is 0.1, and the initial value of c is 0.01.

The selection of the search method was determined by a random parameter within $[0, 1]$. The first search method used was a global search. These butterflies produce fragrances at their positions. By comparing the fragrance of each butterfly, the butterfly with the strongest fragrance was selected and its position was regarded as the optimal position. In the global search stage, the butterfly population moves towards the butterfly with the strongest fragrance to achieve the purpose of renewing the position of the butterfly population. The specific search method was as Equation (3):

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i, \tag{3}$$

where x_i^t is the solution vector x_i for i th butterfly in iteration number t . Here, g^* is the globally optimal individual in the current iteration. The fragrance of the i th butterfly is denoted by f_i and r is a random number in $[0, 1]$.

The second search method is local search. Considering other factors, such as the distance between butterfly individuals, the algorithm performs a local search using Equation (4).

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i, \tag{4}$$

where x_j^t and x_k^t represent the solution vector of the i th and k th butterfly in the solution space. And r is a random number in $[0, 1]$.

Although the BOA has a fast search speed in the global search, once a butterfly individual with a strong fragrance appears, other butterflies will move toward it quickly. The algorithm solution process has a poor diversity of solution results and is easily trapped

in local extremes. The local search depends only on two random solution vectors, so the algorithm can easily miss the better solution. The following improvement strategies were proposed to solve the aforementioned problems of the BOA.

3.1. Kent Mapping

Chaos theory is characterized by randomness, non-repeatability, ergodicity, and regularity. Compared to the stochastic search, chaos theory can perform a comprehensive and thorough search in the search space. The introduction of chaos theory into the metaheuristic algorithm could effectively increase the diversity of the population and improve the solving power. The introduction of chaotic mapping in the BOA improves the convergence accuracy of the algorithm, ensures the diversity of butterfly positions, improves the global search ability of the algorithm, and avoids falling into the local optimal solution in the process of the algorithm operation.

There are many types of chaotic mappings, and kent mapping has good uniform traversal, which can effectively improve the diversity of butterfly populations. Therefore, in this study, kent mapping is chosen to optimize the random parameter r in the BOA position renewal process so that the butterfly individual position distribution is more random. Incorporating the mapped parameter r into the operation can significantly improve the ability of the BOA to jump out of the local optimum. In addition, the fixed mapping number is set to avoid the overall slow iteration speed while improving the local optimal ability in the early search. The specific mathematical model is as follows:

During the first 1/3 population position iteration process, the parameter u is set to 0.6, and the value of r is solved by Equation (5).

$$r = \begin{cases} (1-r)/(1-u) & r < u \\ r/u & r \geq u \end{cases} \quad (5)$$

The butterfly position renewal process was optimized by incorporating the mapped r into the position renewal calculation of the butterfly population.

3.2. Adaptive Inertia Weights

The addition of adaptive inertia weights can effectively improve the convergence speed and accuracy of the optimization algorithm [26].

By analyzing the principle of the BOA, it can be seen that the individual butterfly mainly renews its position through its current position and the position of the current optimal solution. This renewal method can easily cause BOA to fall into a local optimum. The convergence speed of BOA cannot be effectively controlled. Therefore, in this study, the adaptive weight coefficient w is added in the process of butterfly position renewal to improve the algorithm's ability to jump out of the local optimum in the early stage and speed up the convergence speed and accuracy of the algorithm in the later stage. The mathematical expression of the adaptive weight coefficient w is given by Equation (6):

$$w = 2.5(\sin(\frac{\pi}{2}(\frac{t}{N_iter})^{0.8}) - 3). \quad (6)$$

The improved Equation (3) is shown in Equation (7):

$$x_i^{t+1} = x_i^t + (r^2 \times w \times g^* - x_i^t) \times f_i. \quad (7)$$

The improved Equation (4) is shown in Equation (8):

$$x_i^{t+1} = x_i^t + (r^2 \times w \times x_j^t - x_k^t) \times f_i. \quad (8)$$

3.3. Opposition-Based Learning Strategy Based on Convex Lens Imaging

To ensure the diversity of the entire population and avoid the problem of “premature” convergence and low optimization accuracy of the algorithm, an opposition-based learning strategy based on convex lens imaging is introduced into BOA, and it is applied to the current optimal individual to generate new individual [27].

The principle of searching for an opposing individual of the optimal butterfly individual based on an opposition-based learning strategy based on convex lens imaging is as follows. Suppose that in a one-dimensional space, there is a butterfly individual P with height h on the coordinate axis interval $[lb, ub]$. The projection of P on the x-axis is X (X is the global optimal individual), and the value range of the global solution is set to $[ub, lb]$. A convex lens with a focal length of F is placed at the base point position O (the base point position in this study is $(lb + ub)/2$), and the individual butterfly P obtains an inverted image P* with a height of h^* through the convex lens. The imaging of the convex lens produces an opposing individual X* on the X-axis, and the imaging principle is shown in Figure 4.

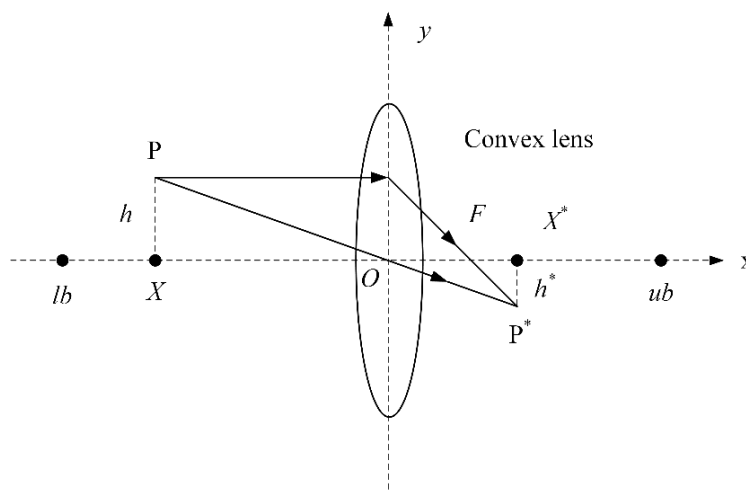


Figure 4. Schematic of opposition-based learning strategy based on convex lens imaging.

In Figure 4, the global optimal butterfly individual X finds the corresponding opposing individual X* through the base point O, and Equation (9) is obtained according to the principle of convex lens imaging.

$$\frac{h}{h^*} = \frac{(ub + lb)/2 - X}{X^* - (ub + lb)/2} \tag{9}$$

Equation (10) is obtained from the Equation (9) transformation.

$$\frac{h}{h^*} = \eta, \tag{10}$$

where η is the scaling factor, that is, the corresponding proportional relationship between the object and image.

According to Equations (9) and (10), Equation (11) for X* can be obtained:

$$X^* = \frac{ub + lb}{2} + \frac{ub + lb}{2 \cdot \eta} - \frac{X}{\eta} \tag{11}$$

In the calculation process of BOA, the scaling factor η was used as a microregulatory factor to strengthen the local development capability of the algorithm, and the optimal η was determined by comparing different values. Through opposition-based learning of the global optimal individual in the algorithm, the solution accuracy of the algorithm is greatly increased. Figure 5 shows a flowchart of IBOA.

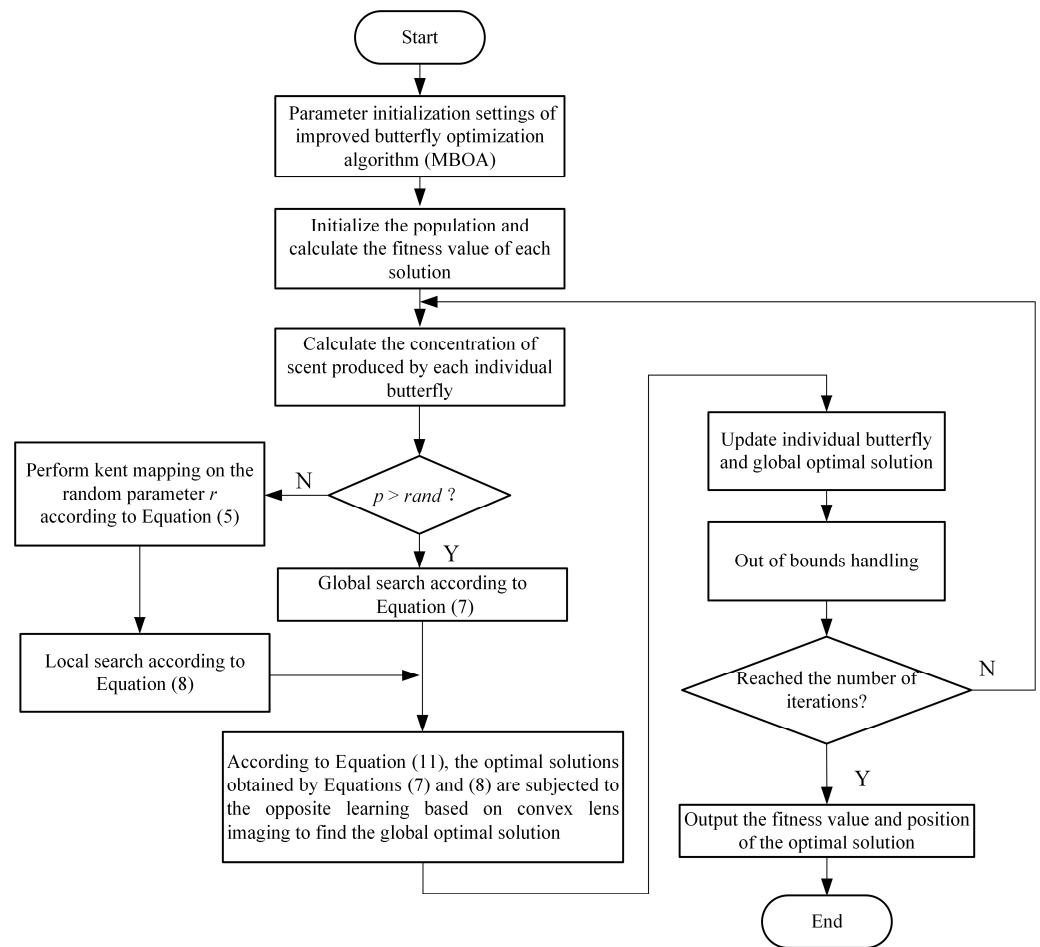


Figure 5. Flow chart of IBOA.

3.4. Comparison of BOA with IBOA

To verify the optimization effect of the IBOA proposed in this paper, the six general benchmark functions of metaheuristic algorithms are solved using the BOA and IBOA algorithms. These benchmark functions contain unimodal and multimodal functions. Unimodal functions can evaluate the solution speed of the algorithm. Multimodal functions can evaluate the ability of the algorithm to jump out of the local optimal solution during the solution. By analyzing the solution results of the two algorithms, their superiority and stability were compared.

The relevant information of the test function used this time is shown in Table 1.

Table 1. Benchmark functions used in this study.

Function	V_no	Range	fmin
$F_1(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	100	[-100, 100]	0
$F_2(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	100	[-100, 100]	0
$F_3(x) = \sum_{i=1}^n x_i^2$	100	[-100, 100]	0
$F_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	100	[-5.12, 5.12]	0
$F_5(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - e(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 2 + \exp(1)$	100	[-32, 32]	0
$F_6(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	100	[-600, 600]	0

Figure 6 depicts the convergence curves of BOA and IBOA on 50 iterations. It can be observed from Figure 6 that the IBOA exhibits a faster convergence speed and higher convergence accuracy in the process of solving different benchmark functions.

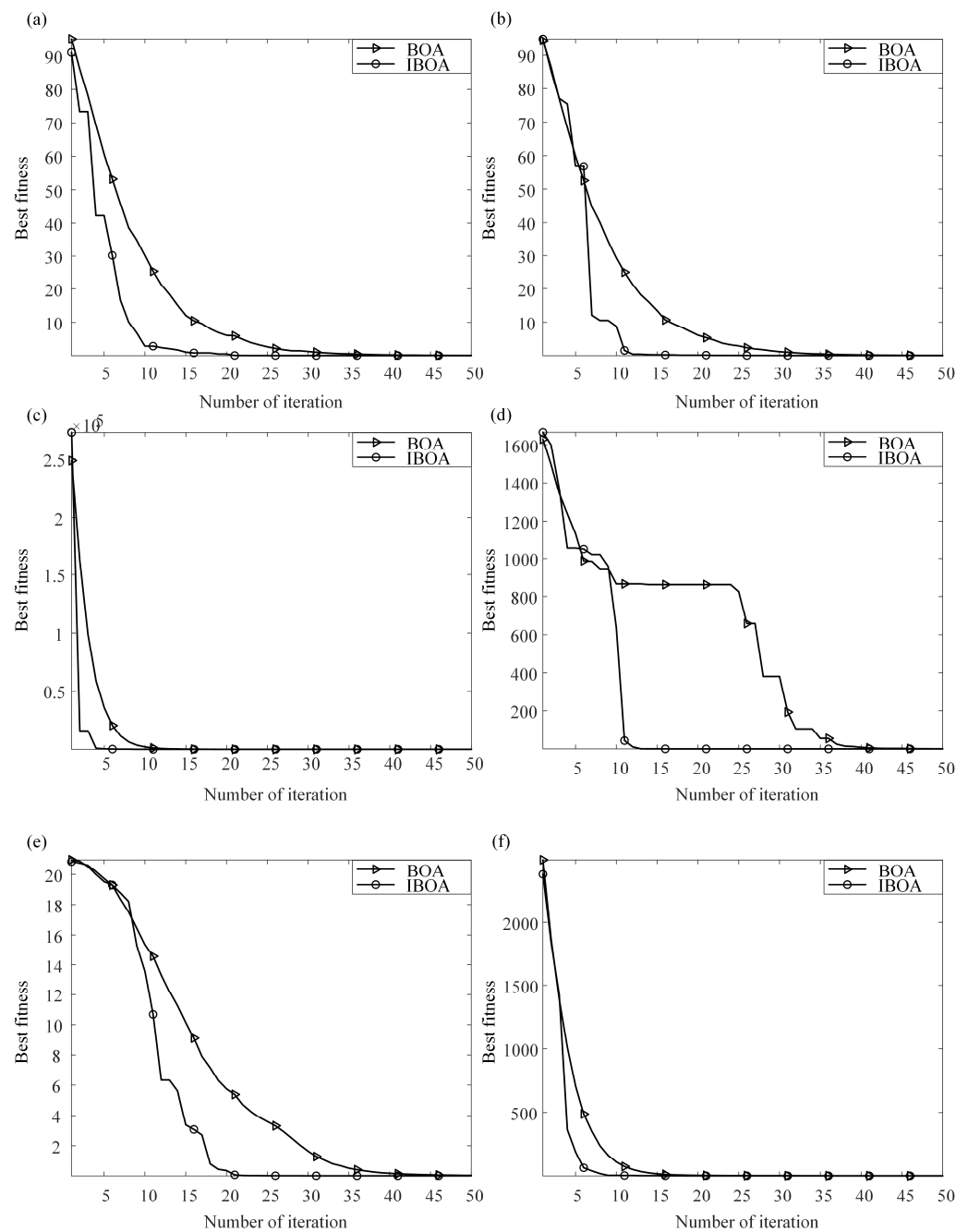


Figure 6. Comparison of the solving procedures for the benchmark functions F1-F6: (a) Solution process of the datum function F1; (b) Solving process of the reference function F2; (c) Solution process of the reference function F3; (d) Solution process of the reference function F4; (e) Solution process of the reference function F5; (f) Solution process of the reference function F6.

To verify the effectiveness of the IBOA strategy, the six benchmark functions were solved 20 times by the BOA and IBOA. In this study, the mean and standard deviation of 20 solutions of the two algorithms for each benchmark function were calculated. The overall accuracy of the algorithm’s solution is observed by comparing the means, and the stability of the algorithm’s solution is determined by comparing the standard deviations. The solution results are shown in Table 2.

Table 2. Comparison of the solution results obtained from the benchmark function.

F	BOA		IBOA	
	ave	std	ave	std
F1	2.4432×10^{-5}	2.5048×10^{-6}	3.7334×10^{-9}	1.74×10^{-9}
F2	0.0018123	0.00013418	9.406×10^{-7}	4.6296×10^{-7}
F3	3.035×10^{-5}	2.2703×10^{-6}	3.9923×10^{-9}	2.6031×10^{-9}
F4	2.2523	7.8826	1.8149×10^{-7}	5.9147×10^{-7}
F5	0.036973	0.0036808	0.00012988	0.00027588
F6	0.017841	0.0032376	9.0913×10^{-7}	1.0834×10^{-6}

It can be seen from the mean and standard deviation of the solution results of multiple benchmark functions, that the IBOA has a higher accuracy than the traditional BOA solution, indicating that the IBOA solution is more accurate, and the IBOA solution ability is more powerful. The standard deviation of the results of the IBOA solution is generally small, indicating that the solution ability of the IBOA is more stable. Therefore, it can be shown that IBOA has a faster convergence rate and more accurate and stable solutions.

4. Path Planning Based on IBOA

4.1. Application of IBOA in Path Planning

The process of solving the mobile robot path-planning problem using the BOA is as follows [28]. The path change process is shown in Figure 7.

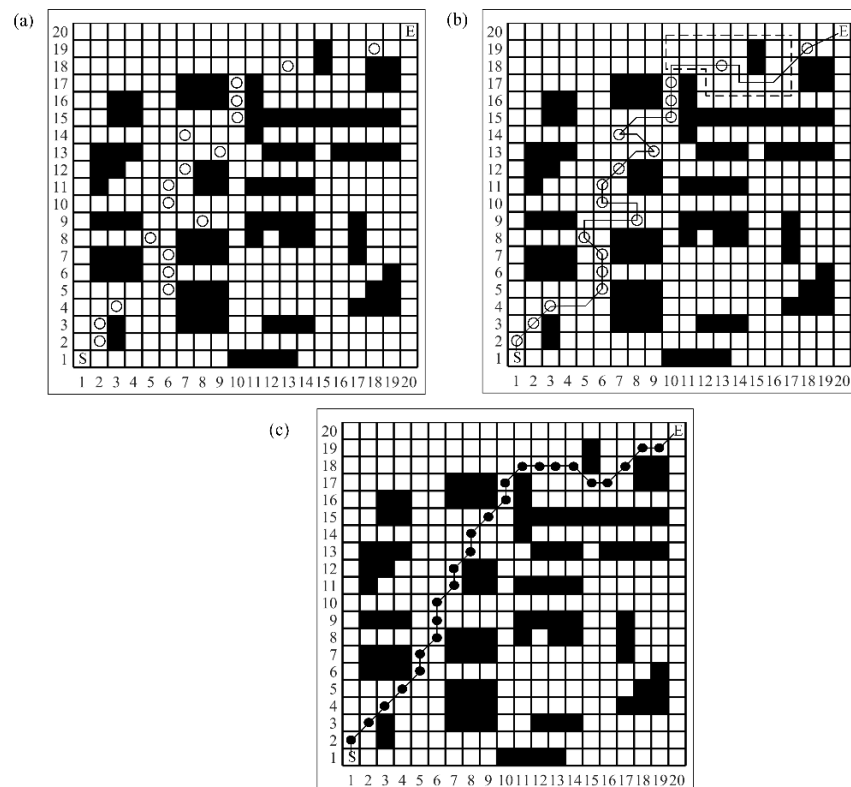


Figure 7. Schematic of process of obtaining the shortest path by the IBOA: (a) First generation path; (b) Second generation path; (c) Third generation path.

First, in the 20×20 grid, as shown in Figure 7a, a free grid was selected as the point through which the robot passes in each of the 18 rows in the middle of the starting and target points. These necessary grids were represented by hollow circles. These 20 grids were referred to as individual butterfly clusters in IBOA.

However, the path formed by these 20 grids could not be directly used as the path of the robot. Therefore, first-generation routing must be interpolated. All two discontinuous points in the path are interpolated until all adjacent grids in the path are continuous, and the algorithm outputs a path for the robot to move.

The second-generation paths after interpolation of the first-generation paths are shown as solid lines in Figure 7b. The length of each path in the second-generation path set is calculated and used as the fitness value of the individual butterfly.

For storing the path information, the cell array was applied to store the path information of different lengths. For reading and computing path information, the numeric array was applied to read specific path information from the cell array for computation and iteration.

Although the number of grids contained in each path in the second-generation path set is inconsistent, it causes problems in the overall position update of the IBOA population. However, the interpolation method employed follows a strict logic. Therefore, the first-generation paths correspond one-to-one with the second-generation paths. When IBOA is operated, the renewing result of the first-generation path composed of 20 grids is the same as the renewing result of the second-generation path.

However, as shown by the dashed box in Figure 7b, some detour paths may appear during the second-generation path. To avoid the detour path included in the final solution path, the second-generation path was processed. This study introduces the mutation process of GA to improve the BOA. The third-generation path is solved. The third generation path was shown in Figure 7c. The edge length of the grid is assumed to be 1 m.

To verify the effectiveness of the improved strategy in solving the path planning problem, IBOA, GA, and BOA were compared. Figure 8a depicts the convergence curves of the shortest path lengths for IBOA and BOA after 50 iterations. The number of algorithm populations and the maximum number of iterations are 200 and 50, respectively. As shown in Figure 8a, IBOA has the fastest search for the optimal path and the shortest length of the optimal path. The shortest paths solved by the three optimization algorithms in grid maps are shown in Figure 8b.

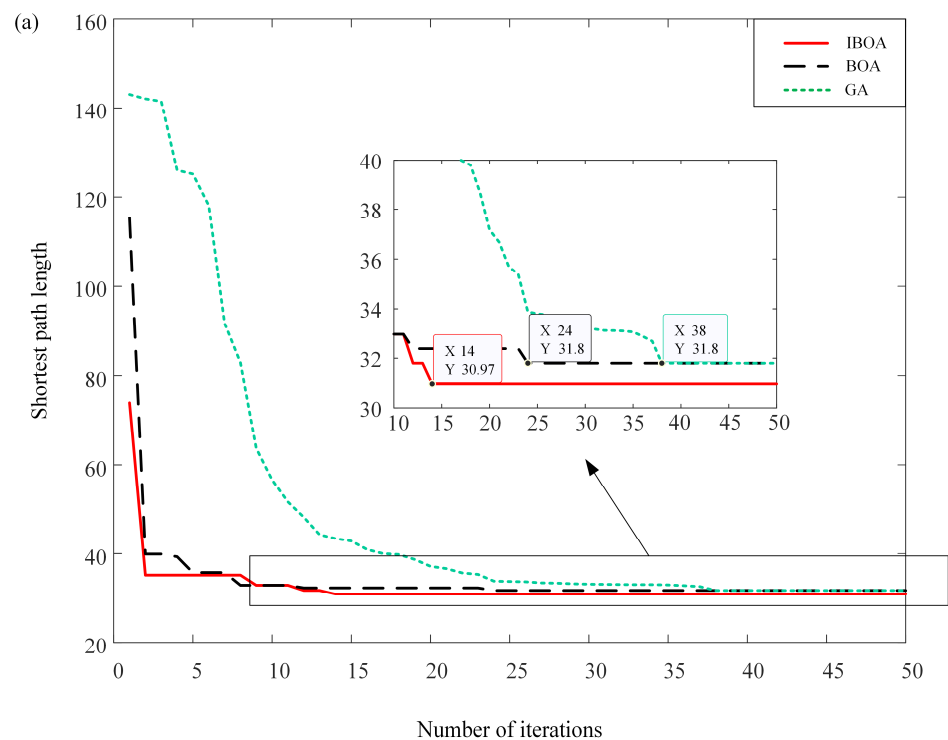


Figure 8. Cont.

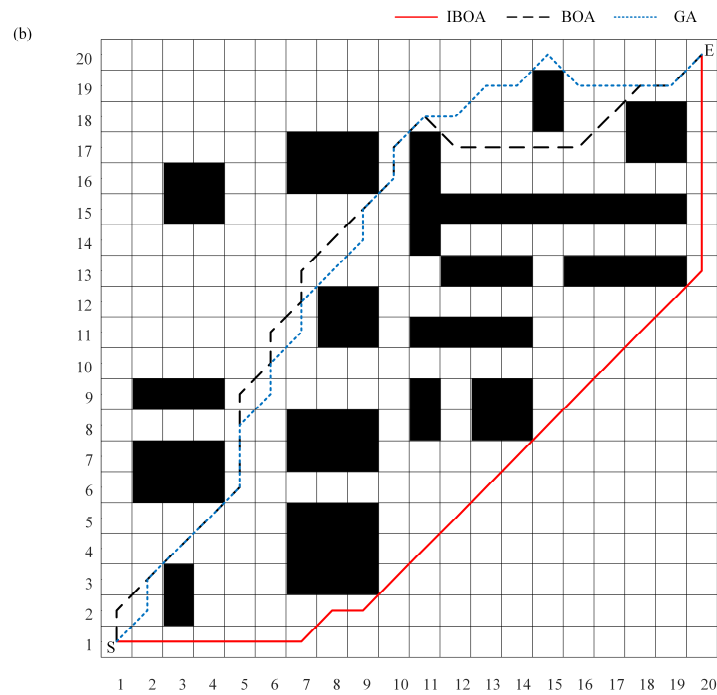


Figure 8. Solution results for IBOA, BOA, and GA: (a) Shortest path convergence curves of the IBOA, BOA, and GA solutions; (b) Shortest paths of IBOA, BOA, and GA solutions.

The shortest path lengths searched by IBOA, BOA, and GA are successively 30.97 m, 31.799 m, and 31.799 m in Figure 8a. The number of iterations for the shortest paths searched by IBOA, BOA, and GA are 14, 24, and 38 in that order.

In order to further investigate the solving capability of the IBOA algorithm, this study solved the shortest path of a more complex map by IBOA, BOA, and GA. The map is a combination of the maps in reference [3] and reference [18]. Thus, the map is 40×40 in size and has a more complex environment. The results of the solution are shown in Figure 9.

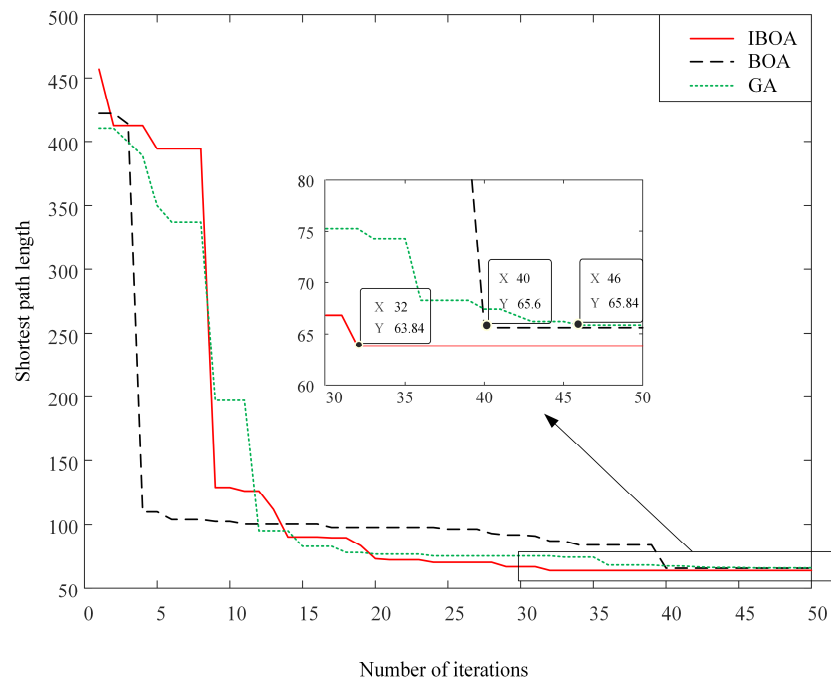


Figure 9. Shortest path convergence curves of IBOA, BOA, and GA solutions in 40×40 maps.

The population size of the algorithm needs to be increased to 4000 because the map is substantially larger. The number of iterations for the shortest paths searched by IBOA, BOA, and GA are 32, 40, and 46 in that order. The shortest paths solved by IBOA, BOA, and GA in the 40×40 map are shown in Figure 10.

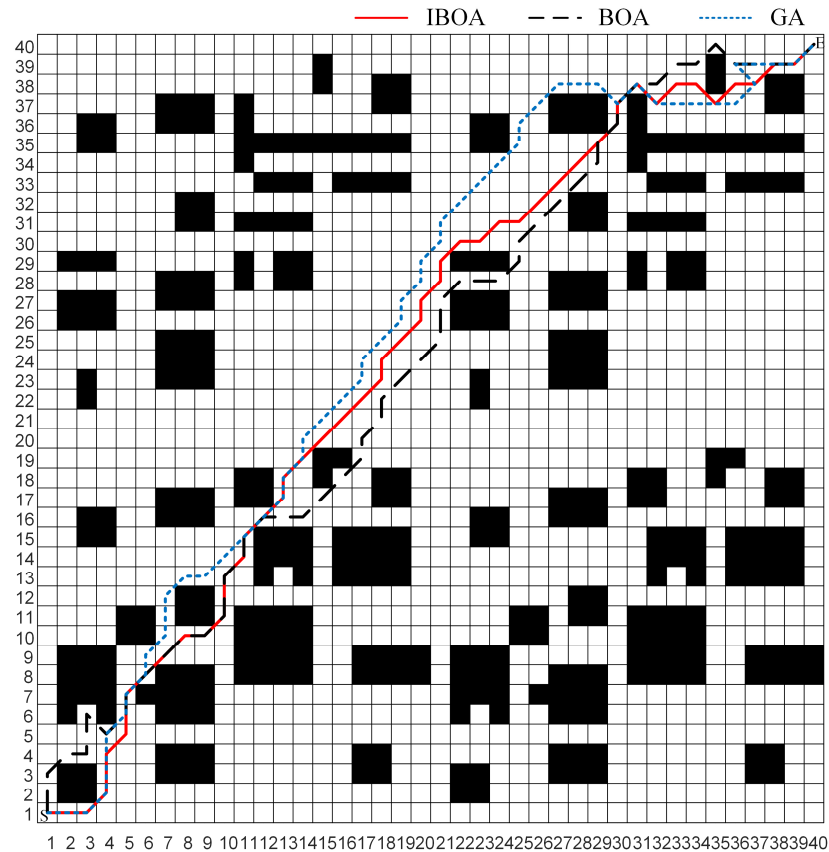


Figure 10. Shortest paths of IBOA, BOA, and GA solutions.

Despite the increased complexity of the map environment, the IBOA solution still has the shortest path lengths and the fastest solution speed.

In order to better verify the ability of the algorithms to solve the shortest path, each algorithm solved the path 10 times. The minimum/maximum/average/median values of the shortest path lengths solved by the algorithms multiple times were compared. The comparison results are shown in Table 3.

Table 3. Comparison of multiple solution results for 20×20 maps.

Value	IBOA	BOA	GA
Max	31.7989898700000	36.6274170000000	36.9705627500000
Min	30.9705627500000	31.7989898700000	31.7989898700000
Average	31.7161471580000	33.7989898730000	33.1504617360000
Median	31.7989898700000	33.2132034400000	32.3847763100000

The minimum/maximum/mean/median values of the shortest path lengths solved by IBOA multiple times are smaller than those of BOA and GA.

A comparison of the results of multiple solutions for the 40×40 map is shown in Table 4.

Table 4. Comparison of multiple solution results for 40×40 maps. The results from the comparison show that the path of the IBOA solution is the shortest.

Value	IBOA	BOA	GA
Max	68.5269119345812	67.9411254969543	73.0121933088197
Min	63.8406204335659	65.5979797500000	65.8406204300000
Average	65.7778787337689	66.8766594032423	69.4440692218820
Median	65.8406204335659	66.7695526217004	69.6984848100000

4.2. Solution of Redundant Routes in Path Planning

To reduce the energy consumption of the robot during movement, it is necessary to shorten the path length as much as possible. Therefore, this study proposes two simplification strategies to minimize the path length [29,30].

Strategy 1: The methods for determining the simplistic path are mainly divided into three cases, as shown in Figure 11a.

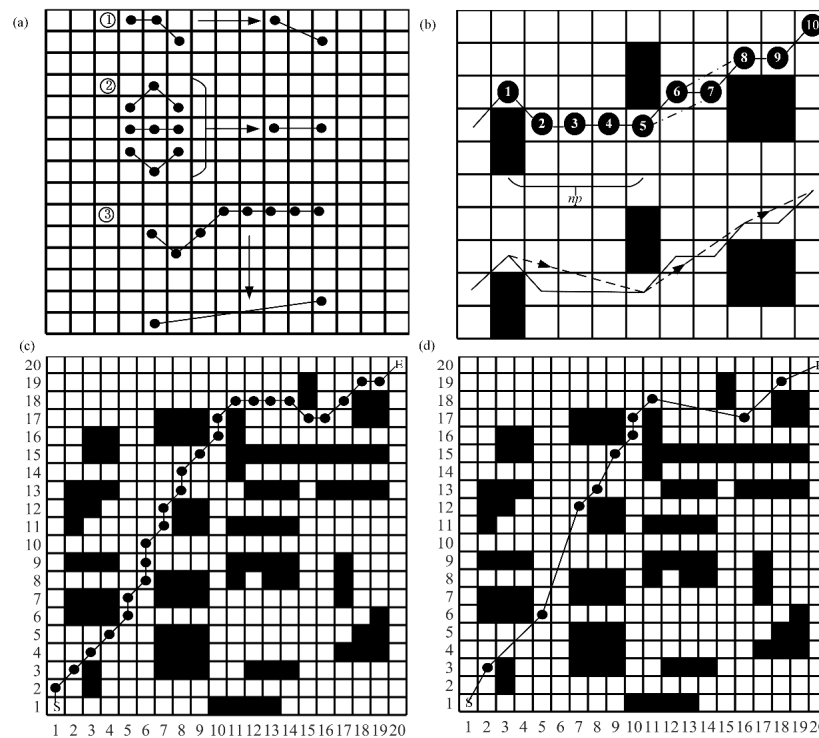


Figure 11. Two path simplification strategies: (a) Simplified schematic of strategy 1; (b) Simplified schematic of strategy 2; (c) The shortest path solved by IBOA; (d) Shortest path searched by IBOA after combining two simplification strategies.

The end of the path is defined as a P_{end} point, and the current point is assumed to be $P(i)$, where $i(i < P_{end} - 1)$ is the position of the current point in the path.

Case 1: When the line connecting $P(i)$ and $P(i + 2)$ is the diagonal line of a 3×2 rectangular frame if the 6 grids in the rectangular frame are all free, then $P(i + 1)$ will be judged that the point can be deleted, and the matrix is judged to be a reducible rectangle.

Case 2: When $P(i)$ and $P(i + 2)$ are in the same row, it is necessary to determine whether the grids in the 3×3 matrix whose rectangular midline is the connecting line between $P(i)$ and $P(i + 2)$ are free. If the nine grids are all free, the matrix is judged to be a reduced rectangle, and $P(i + 1)$ is judged to be a point that can be deleted.

Case 3: When two consecutive points and more, that is, when the points $P(i)$ to $P(i + nc)$, $nc < P_{end} - 1 - i$, satisfy one of the above two cases, then the points $P(i + 1)$ to $P(i + nc + 1)$ can be judged as removable points.

This simplification strategy applies to both the horizontal and vertical directions of the path.

Strategy 2: The main process simplifies the path when some of the grids in the $n \times 2$ rectangular grids composed of three consecutive points in the path are obstacle grids. Grids were calculated between the first and third points.

The horizontal direction was simplified as an example, as shown in Figure 11b. In the lower path diagram in Figure 11b, the fine solid line is the original path, and the dashed line is the simplified path. When the robot passes three points, 1→2→3, point 1 is called the simplified initial point, and the number n_{in} of the horizontal grids between points 1 and 3 is calculated. When the number of continuous horizontal free grids at point 1 in the direction of point 3 reaches np and the number of continuous horizontal free grids at point 3 in the direction of point 1 also reaches np , the condition for removing point 2 is satisfied, and the path can be simplified. n_{in} and np are calculated as follows:

$$n_{in} = \frac{|x_3 - x_1| - 1}{2}, \quad (12)$$

where x_1 is the number of columns of the simplified initial point, and x_3 is the number of columns of the 3rd point.

$$np = \text{ceil}(0.5n_{in}). \quad (13)$$

As shown in Figure 11, when the robot passes through the four points 5→6→7→8, it can be found that both the 5→6→7 and 6→7→8 segments satisfy the path simplification conditions. At this time, the path is in a simplified path superposition situation because point 4 can be deleted when the 5→6→7 path segment satisfies the simplified conditions and point 5 can be deleted when the 6→7→8 road segment satisfies the path simplified conditions. Therefore, Section 5→6→7→8 is simplified to 5→8. As shown in Figure 11b, the 5→8 route exists in the simplified paths 5→7 and 6→8, so the feasibility of 5→8 can be guaranteed. Extending this rule, when m simplified initial points appear in the path consecutively, let $Po(ii)$ be the first simplified initial point, $Po(ii + m - 1)$ is the m th simplified initial point, and ii is the sequence number of points in the robot's motion path only needs to keep the points $Po(ii)$ and $Po(ii + m + 1)$. Points in the middle were deleted to simplify the path.

This simplification strategy also applies to both the horizontal and vertical paths. Schematics of the two simplification strategies are shown in Figure 11.

Two simplification strategies are introduced into the IBOA for path planning. Figure 11c shows the shortest path solved by IBOA. The path length is 31.799 m. Figure 11d depicts the shortest path finally solved by the IBOA. The path length is 30.2914 m, and the path length is significantly reduced.

In order to verify the generality of the application of the proposed simplification strategy, it was applied to the fusion environment model with a size of 40×40 as described in the previous section. In Figure 12, the IBOA_2g curve is the simplified robot movement path and the IBOA curve is the path solved by IBOA.

Figure 13 shows the overall flow chart of robot path planning.

First, the algorithm generated an initial population containing 50 individuals. Each of these individuals is composed of one blank grid selected from any one of the rows of the grid map. The number of variables contained in a single individual is equal to that of rows in the grid map. Next, each individual is interpolated to make it a feasible path that each route grid is connected to. Individuals that cannot be interpolated to a feasible path would be discarded. The above process was repeated, and the loop jumped out when the set number of iterations was reached. Finally, the shortest path solved by the algorithm was simplified by the proposed path simplification strategies to obtain the final path.

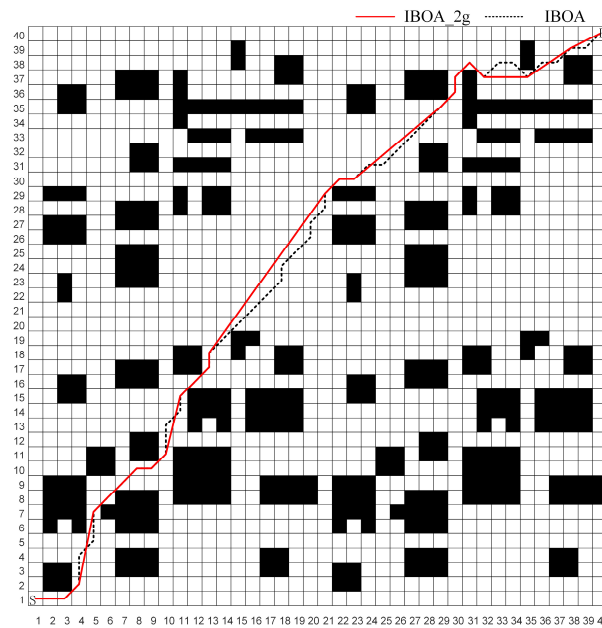


Figure 12. The shortest path searched in a 40 × 40 map by IBOA combining two simplified strategies.

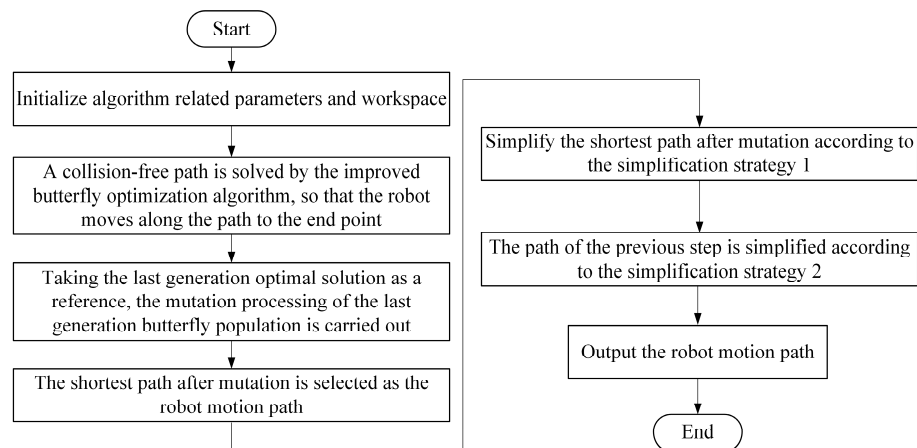


Figure 13. Flow chart of robot path planning.

5. Discussion

In the path planning problem, the accurate obstacle model can effectively avoid the collision between the mobile robot and the obstacle. At present, most of the different algorithms can solve shorter paths, but the accuracy and speed of the shortest path still need to be improved. Therefore, the quadratically inflated obstacle model, the IBOA, and the two path simplification strategies were proposed.

In the traditional obstacle collision model, there is a risk of collision with an obstacle when the mobile robot passes over the vertices of the obstacle grid. The obstacle model proposed in this paper ensures that a mobile robot does not collide with an obstacle while passing through the vertices of the obstacle grid. The traveling safety of the robot is guaranteed with the size of the obstacle model as small as possible.

For the general BOA, although it has a faster solution speed, it is easy to fall into the local optimum. For the improvement of the optimization algorithm, many scholars have used chaos theory and the design of appropriate adaptive weights to modify the algorithm. Therefore, in this study, kent mapping and convex lens inverse imaging theory were introduced to improve the ability of the algorithm to jump out of the local optimum. In addition, the convergence speed of IBOA was effectively adjusted by the designed adaptive weight.

A mobile robot moving in a raster map must pass through the center of the raster, and, therefore, the passing rasters must be in contact with each other. This will generate some redundant points in the planned path. To address this part of the problem, the literature [11] proposed a path simplification strategy which was applied to small-scale path simplification. The path simplification strategy designed in this study is suitable for path simplification in a small range and can also remove redundant sections in a large range.

Combined with the above scheme, this research was completed to investigate the path-planning algorithm for mobile robots. A safer obstacle model was designed in this paper. This research effectively improved the solving ability of BOA and, therefore, improved the solving speed as well as the solving accuracy of the robot path planning algorithm. The path simplification strategies designed in this paper effectively solved the problem of redundant points in the planned paths under grid maps.

Different algorithms have their own merits, and the path planning algorithms will be further improved by comparing them with more algorithms. Domestic and foreign scholars also tend to fuse more than one algorithm. In addition, further research on the speed and energy management of the moving mobile robot on the planned path is also the work needed in the future.

6. Conclusions

The further simplification of the shortest path solved by the algorithm can effectively avoid the redundant points caused by the grid map. In this study, a quadratic inflated obstacle model is proposed. Compared with the inflated model in the literature [25], the proposed quadratic inflated obstacle model could better avoid contact between the mobile robot and the obstacles, which ensures the safety of the mobile robot traveling according to the planned path. For the improvement of the traditional BOA, this paper designs adaptive weight giving for the BOA and introduces the kent mapping as well as the convex lens-based backward learning strategy into BOA for the first time. The effectiveness of the improvement strategies was verified by testing different test functions. Applying the IBOA algorithm to the mobile robot path planning problem, the IBOA algorithm solves the shortest path with higher accuracy and faster iteration speed compared with BOA and GA. In both the 20×20 map and the 40×40 map, the IBOA algorithm shows stronger solving ability, and as the complexity of the environment increases, the advantage of the algorithm IBOA algorithm solving becomes more and more obvious. In the 40×40 map, the IBOA algorithm converges in 32 generations, and the shortest path is 63.84, while BOA and GA converge in 40 and 46 generations, respectively, and their shortest paths are 65.60 and 65.84. From the analysis of the max/min/average/median values of paths many times solved, IBOA has a significant advantage. For the solved shortest path, the two simplification strategies proposed in this paper can simplify the redundant sections in the path. Compared with the path simplification strategies in the literature [11], the main way of the proposed simplification strategies is to remove the redundant points in the path. The proposed simplification strategy has the superposition property, which can remove the redundant paths between two points that can be directly connected. The simplified length of the shortest path solved by IBOA is 61.03.

Author Contributions: All authors contributed extensively to this work; R.Z. and P.X. proposed the key ideas, analyzed the key contents using a simulation, and wrote the manuscript; D.S. and Y.S. obtained the financial support for the project leading to this publication; M.J. modified the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the financial support of the Industrial Collaborative and Innovative Special Fund Co-sponsored by Anhui Polytechnic University and Jiujiang District of Wuhu City (No. 2021cyxtb5 and No. 2022cyxtb3), National Natural Science Fund (No.52171148), Jiangsu Funding Program for Excellent Postdoctoral Talent (No. 2022ZB244), Project funded by China Postdoctoral Science Foundation (No. 2022TQ0159), Project funded by China Postdoctoral Science Foundation (No. 2022M721624), and Key Research and Development Projects in Anhui Province (NO.2022a05020007).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sathiya, V.; Chinnadurai, M. Evolutionary algorithms-based multi-objective optimal mobile robot trajectory planning. *Robotica* **2019**, *37*, 1363–1382. [[CrossRef](#)]
2. Wang, J.; Meng, M.Q.H.; Khatib, O. Optimal motion planning for mobile robots. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 2063–2073. [[CrossRef](#)]
3. Li, K.; Hu, Q.; Liu, J. Path planning of mobile robot based on improved multiobjective genetic algorithm. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 8836615. (In Chinese) [[CrossRef](#)]
4. Tan, C.S.; Mohd-Mokhtar, R.; Arshad, M.R. A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access* **2021**, *9*, 119310–119342. [[CrossRef](#)]
5. Chi, W.; Wang, C.; Wang, J.; Meng, M.Q.H. Risk-DTRRT-based optimal motion planning algorithm for mobile robots. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1271–1288. [[CrossRef](#)]
6. Bozek, P.; Karavaev, Y.L.; Ardentov, A.A.; Yefremov, K.S. Neural network control of a wheeled mobile robot based on optimal trajectories. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 172988142091607–172988142091616. [[CrossRef](#)]
7. Zahid, T.; Kausar, Z.; Shah, M.F.; Saeed, M.T.; Pan, J. An intelligent hybrid control to enhance applicability of mobile robots in cluttered environments. *IEEE Access* **2021**, *9*, 50151–50162. [[CrossRef](#)]
8. Alireza, M.; Vincent, D.; Tony, W. Experimental study of path planning problem using EMCOA for a holonomic mobile robot. *J. Syst. Eng. Electron.* **2021**, *32*, 1450–1462. [[CrossRef](#)]
9. Liu, X.; Wang, W.; Li, X.; Liu, F.; He, Z.; Yao, Y.; Ruan, H.; Zhang, T. MPC-based high-speed trajectory tracking for 4WIS robot. *ISA Trans.* **2022**, *123*, 413–424. [[CrossRef](#)]
10. Ali, H.; Gong, D.; Wang, M.; Dai, X. Path planning of mobile robot with improved ant colony algorithm and MDP to produce smooth trajectory in grid-based environment. *Front. Neurobot.* **2020**, *14*, 44–56. [[CrossRef](#)]
11. Sánchez-Ibáñez, J.R.; Pérez-Del-Pulgar, C.J.; García-Cerezo, A. Path planning for autonomous mobile robots: A review. *Sensors* **2021**, *21*, 7898. [[CrossRef](#)]
12. Paikray, H.K.; Das, P.K.; Panda, S. Optimal multi-robot path planning using particle swarm optimization algorithm improved by sine and cosine algorithms. *Arab. J. Sci. Eng.* **2021**, *46*, 3357–3381. [[CrossRef](#)]
13. Parque, V.; Miyashita, T. Smooth curve fitting of mobile robot trajectories using differential evolution. *IEEE Access* **2020**, *8*, 82855–82866. [[CrossRef](#)]
14. Zhang, B.; Zhu, D. A new method on motion planning for mobile robots using jump point search and Bezier curves. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 172988142110192–172988142110204. [[CrossRef](#)]
15. Gul, F.; Mir, I.; Abualigah, L.; Sumari, P.; Forestiero, A. A consolidated review of path planning and optimization techniques: Technical perspectives and future directions. *Electronics* **2021**, *10*, 2250–2287. [[CrossRef](#)]
16. Muthukumar, S.; Sivaramakrishnan, R. Optimal path planning for an autonomous mobile robot using dragonfly algorithm. *Int. J. Simul. Model.* **2019**, *18*, 397–407. [[CrossRef](#)] [[PubMed](#)]
17. Li, X.; Wang, L. Application of improved ant colony optimization in mobile robot trajectory planning. *Math. Biosci. Eng.* **2020**, *17*, 6756–6774. [[CrossRef](#)] [[PubMed](#)]
18. Zhang, Z.; Wan, Y.; Wang, Y.; Guan, X.; Ren, W.; Li, G. Improved hybrid A* path planning method for spherical mobile robot based on pendulum. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 172988142199295–172988142199308. [[CrossRef](#)]
19. Zhou, H.; Xiong, H.L.; Liu, Y.; Tan, N.; Chen, L. Trajectory planning algorithm of UAV based on system positioning accuracy constraints. *Electronics* **2020**, *9*, 250–270. [[CrossRef](#)]
20. Precup, R.; Voisan, E.; Petriu, E.M.; Tomescu, M.L.; David, R.; Szedlak-Stinean, A.; Roman, R. Grey wolf optimizer-based approaches to path planning and fuzzy logic-based tracking control for mobile robots. *Int. J. Comput. Commun. Control* **2020**, *15*, 1–17. [[CrossRef](#)]
21. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [[CrossRef](#)]
22. Chen, Z.Q.; Zhou, Q.; Yuan, Z.Z. Research on the Digital Fountain Codes and Decodes Algorithm Based upon Kent Mapping. *J. Syst. Sci. Math. Sci.* **2011**, *31*, 731–741. (In Chinese)
23. Falcó, A.; Hilario, L.; Montés, N.; Mora, M.C.; Nadal, E. A path planning algorithm for a dynamic environment based on proper generalized decomposition. *Mathematics* **2020**, *8*, 2245–2255. [[CrossRef](#)]
24. Kim, H.; Kim, B.K. Energy-optimal transport trajectory planning and online trajectory modification for holonomic robots. *Asian J. Control* **2021**, *23*, 2185–2200. [[CrossRef](#)]
25. Xu, L.; Fu, W.H.; Jiang, W.H.; Li, Z.T. Mobile robots path planning based on 16-directions 24-neighborhoods improved ant colony algorithm. *Control Decis.* **2021**, *36*, 1137–1146. (In Chinese)

26. Wang, W.J.; Tao, Q.; Cao, Y.T.; Wang, X.H.; Zhang, X. Robot time-optimal trajectory planning based on improved cuckoo search algorithm. *IEEE Access* **2020**, *8*, 86923–86933. [[CrossRef](#)]
27. He, Q.; Luo, S.H. Chimp optimization algorithm based on hybrid improvement strategy and its mechanical application. *Control Des.* **2021**, 1–11. (In Chinese) [[CrossRef](#)]
28. Yang, H.; Qi, J.; Miao, Y.; Sun, H.; Li, J. A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8557–8566. [[CrossRef](#)]
29. Muhammad, A.; Ali, M.A.H.; Turaev, S.; Shanono, I.H.; Hujainah, F.; Zubir, M.N.M.; Faiz, M.K.; Faizal, E.R.M.; Abdulghafor, R. Novel algorithm for mobile robot path planning in constrained environment. *Comput. Mater. Continua.* **2022**, *71*, 2697–2719. [[CrossRef](#)]
30. Sun, Y.; Zhang, C.; Sun, P.; Liu, C. Safe and smooth motion planning for Mecanum-wheeled robot using improved RRT and cubic spline. *Arab. J. Sci. Eng.* **2020**, *45*, 3075–3090. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.