frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 01 June 2023
DOI 10.3389/fnbot.2023.1196683

OPEN ACCESS

EDITED BY
Hang Su,
Fondazione Politecnico di Milano, Italy

REVIEWED BY
Xiaorui Liu,
Qingdao University, China
Shinsuke Yasukawa,
Kyushu Institute of Technology, Japan

*CORRESPONDENCE
Xin Ma
✉ mx_lnfa@163.com

RECEIVED 30 March 2023
ACCEPTED 15 May 2023
PUBLISHED 01 June 2023

CITATION
Cui Q, Liu P, Du H, Wang H and Ma X (2023) Improved multi-objective artificial bee colony algorithm-based path planning for mobile robots. *Front. Neurorobot.* 17:1196683.
doi: 10.3389/fnbot.2023.1196683

Improved multi-objective artificial bee colony algorithm-based path planning for mobile robots

Qiuyu Cui, Pengfei Liu, Hualong Du, He Wang and Xin Ma*

School of Mechanical Engineering and Automation, University of Science and Technology Liaoning, Anshan, Liaoning, China

Mobile robots are widely used in various fields, including cosmic exploration, logistics delivery, and emergency rescue and so on. Path planning of mobile robots is essential for completing their tasks. Therefore, Path planning algorithms capable of finding their best path are needed. To address this challenge, we thus develop improved multi-objective artificial bee colony algorithm (IMOABC), a Bio-inspired algorithm-based approach for path planning. The IMOABC algorithm is based on multi-objective artificial bee colony algorithm (MOABC) with four strategies, including external archive pruning strategy, non-dominated ranking strategy, crowding distance strategy, and search strategy. IMOABC is tested on six standard test functions. Results show that IMOABC algorithm outperforms the other algorithms in solving complex multi-objective optimization problems. We then apply the IMOABC algorithm to path planning in the simulation experiment of mobile robots. IMOABC algorithm consistently outperforms existing algorithms (the MOABC algorithm and the ABC algorithm). IMOABC algorithm should be broadly useful for path planning of mobile robots.

KEYWORDS

mobile robots, path planning, multi-objective artificial bee colony algorithm, Bio-inspired algorithm, multi-objective optimization

1. Introduction

As mobile robots are used in a wide range of applications in the home, defense, medical, and food service industries, they are being given more and more functions (Liu et al., 2022; Shi et al., 2023). Path planning is a key technique in mobile robots. Mobile robot path planning is to find an optimal collision-free path from the start point to the target point for a mobile robot in an environment with multiple obstacles according to a relevant criterion (Zhang et al., 2018). A good path-planning strategy ensures that mobile robots can complete their assigned tasks safely and efficiently.

Several path planning methods have been developed for application to mobile robot path planning, and these methods can be classified as potential field method, sampling method, and swarm intelligence method. The core of the potential field method is to consider the motion of a mobile robot in the environment as a kind of robot motion in a virtual artificial force field. Mohamed et al. (2011) improved the artificial potential field method to solve the problem of mobile robot path planning which is prone to fall into local optimum. Chen et al. (2015) improved the artificial potential field method by adding chaotic optimization methods to solve the local minima and target unreachability problems. Liu et al. (2021) proposed an adaptive path planning method with dual potential field fusion that can satisfy the path planning of mobile robots in the state space with different obstacles and different velocities. Lee et al. (2017) proposed NP-APF, which is a virtual target point in an obstacle-free environment to overcome the local minima and path inefficiency problems.

Hou et al. (2017) proposed an improved artificial potential field (IAPF) method that solves the problems of the traditional APF method in robot path planning under different conditions and avoids the trap problem caused by local minima.

The core of the sampling method is to select a finite number of unstructured points in the configuration space and to establish connections between these points. The RRT algorithm is a sampling-based path planning algorithm that addresses the drawbacks of the RRT algorithm not applying to narrow spaces, dynamic environments, and slow convergence (LaValle and Kuffner, 2001). Kuffner (2000) proposed the RRT-connect algorithm, which introduced the greedy expansion idea on top of the bidirectional RRT algorithm to improve the convergence speed of the algorithm. Karaman and Frazzoli (2011) improved the RRT* algorithm by adding asymptotically optimal properties to guarantee the quality of the paths while preserving the probabilistic completeness of the RRT algorithm. Nasir et al. (2013) proposed the RRT*-smart algorithm to achieve a sample-only search by heuristic sampling, which accelerates the convergence speed. Gammell et al. (2014) proposed the Informed-RRT* algorithm to limit the node sampling range and improve the optimal path convergence speed by generating a subset of ellipsoidal samples.

Since the above methods have many shortcomings, researchers have started to study mobile robot path planning based on Bio-inspired algorithms, mainly including particle swarm algorithms, ant colony algorithms, genetic algorithms, and neural networks, and have achieved many results (Dorigo et al., 1996; Ghatee and Mohades, 2009; Mo and Xu, 2015; Liu et al., 2020). For example, Gunji et al. (2019) successfully obtained a mobile robot roadmap by a hybrid algorithm like cuckoo search and bats, and the algorithm was simulated and tested by a mobile robot in a real workspace. Hosseininejad and Dadkhah (2019) have designed a novel cuckoo search algorithm for the problem of dynamic path planning. Fakoor et al. (2016) proposed a fuzzy control algorithm based on the Markov decision process in which the obstacle avoidance requirement of a mobile robot was successfully implemented. Wang et al. (2016) proposed a genetic bee colony algorithm for the global path-planning problem of mobile robots. Nazarahari et al. (2019) have designed a novel genetic algorithm for path planning of multiple mobile robots in a complex environment map. This algorithm performs path planning in such a way that the path length can be reduced with enhanced safety. The principle is to add several crossover factors and multiple variation operators to the basic genetic algorithm. Qu et al. (2013) proposed a novel genetic algorithm, and the results showed that the convergence speed of the algorithm was enhanced and its search efficiency was improved. And the algorithm can also be well-applied to global path planning for multiple mobile robots. Wang et al. (2012) proposed a hybrid algorithm combining fuzzy logic and neural network, which can successfully bring the mobile robot to the task point under an unknown map. Lamini et al. (2018) devised an improved homo-neighborhood intersection operator that can generate feasible paths with better fitness values, successfully improved the problem of premature convergence of the algorithm, and was well-used in mobile robot path planning. Contreras-Cruz et al. (2015) fused evolutionary planning algorithms in artificial bee colony algorithms and thus obtained a novel algorithm that refines the feasible paths. To verify the authenticity of the algorithm, they conducted test experiments using a mobile robot in a real working environment, and the results confirmed the authenticity of the algorithm and good performance of the algorithm.

In summary, we can see that a very large number of intelligent algorithms have been applied to solve problems related to path planning for mobile robots, and all of them have achieved good results. However, the current research is basically for the single-objective path planning problem, while the mobile robot path planning is a constrained multi-objective optimization problem (Jeddisaravi et al., 2016). We have to consider how to study a more practical and effective way to solve this multi-objective problem.

In 2005, Turkish scholars Karaboga and Basturk first proposed the artificial bee colony algorithm (ABC), the basic idea of which was inspired by the honey harvesting task of bee colonies (Karaboga, 2005). The ABC algorithm has been favored by many scholars due to its advantages such as fast convergence and excellent algorithm performance in solving the problem. As the artificial bee colony algorithm was extended for solving multi-objective problems, the MOABC algorithm was then widely used in various fields. However, multi-objective artificial swarm algorithm-based path planning for mobile robots is rarely reported, so the contribution of this paper is as follows.

1) An improved MOABC algorithm is proposed.
2) The improved MOABC algorithm is applied to solve the mobile robot path planning to verify the effectiveness of the IMOABC algorithm.

The rest of this paper is organized as follows. Section 2 presents the relevant background knowledge. Section 3 presents the improvement and validation of the MOABC algorithm. Section 4 applies the improved MOABC algorithm to mobile robot path planning and validates it with experimental simulations. Section 5 summarizes the entire text.

2. Related background knowledge

2.1. MOABC algorithm

Since its introduction in 2005, the Artificial Bee Colony algorithm (ABC) has been favored by many scholars due to its unique advantages. The basic MOABC algorithm was first proposed by Hedayatzadeh et al. (2010). The MOABC algorithm consists of seven main actions: initializing the honey bee population, foraging for honey bees, generating new food sources, evaluating food source mechanisms, onlooker bees foraging, scouting for bees foraging, and local optimization for external archive. One of the main roles of the external archive is to keep and maintain records of the optimal Pareto solutions found up to now.

The following are the specific steps of the MOABC algorithm implementation:

1. Initialization: the initialization of each parameter and external archive, the honeybee population is initialized according to Equation (1).

$$x_i^j = x_{min}^j + rand\,(0,1)\left(x_{max}^j - x_{min}^j\right) \tag{1}$$

Where is the maximum value of the food source in the direction of j; $x_{\min}^j$ is the minimum value of the food source in the direction of j.

2. Employed bee stage: a solution is randomly selected from the external archive. The current domain search is guided according to Equation (2), and the better solution is retained through the Pareto dominance relation.

$$v_i^j = x_i^j + \phi_i^j\left(x_i^j - x_k^j\right) \tag{2}$$

Wherex_k denotes an adjacent food source,denotes the current food source, $k \in \{1, 2, \cdots, SN\}$ and $k \neq i$. ϕ_i^j is a random number between $[-1, 1]$.

3. Calculate the corresponding following probabilities for all solutions according to Equation (3).

$$\begin{aligned} P_i &= \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \\ fit_i &= \frac{dom(i)}{SN} \end{aligned} \tag{3}$$

Where fit_i is the specific yield of the food source; $dom(i)$ is the number of feasible solutions available in the total feasible solutions for the i and SN is the number of feasible solutions.

4. Onlooker bees stage: a solution is chosen randomly by the following probability, and then the domain search is performed using Equation (3).
5. Scout bees stage: the solution whose *trial* reaches the *Limit* is discarded and the food source information is initialized again according to Equation (1).
6. Update the external archive: add the Pareto optimal solution from the current population to the external archive and update it according to the crowding distance clipping when its upper limit is reached.
7. Export all solutions of external archive.

2.2. Multi-objective mobile robot path planning model

In this paper, we only consider the static known environment and assume that the route of the mobile robot from the starting point to the endpoint in the map environment is a series of path nodes.

Then the path of the mobile robot *Path* can be represented using $Path = \left[Start = P_0, P_1, P_2, \cdots, P_n, P_{n+1} = End\right]$. Where *Start* and *End* are the starting point and the endpoint, respectively. P_i is the ith path node on the path and the two path nodes are considered to be connected by line segments. In this paper, two objective functions are proposed for the two performance metrics of path length and path safety, which are the path length function f_l and the path safety function f_l. The mobile robot path planning problem is transformed into solving the optimal set of solutions of a multi-objective function, whose mathematical model is shown in Equation (4):

$$\begin{cases} F\left(Path\right) = \left[Start = P_0, P_1, P_2, \cdots, P_n, P_{n+1} = End\right] \\ Minimize\ f_l\left(Path\right) = Length\left(Path\right) \\ Minimize\ f_s\left(Path\right) = Safety\left(Path\right) \end{cases} \tag{4}$$

Satisfaction.

$$P_0 = Start,\ P_{n+1} = End,\ Path \cap Obstacle = \emptyset \tag{5}$$

That is, the path *Path* cannot intersect with an obstacle, but is allowed to coincide with the edge of the obstacle. Where represents the set of feasible paths, $Path = \{Path_1, Path_2, ..., Path_m\}$.

In this paper, the adopted path safety function is defined as.

$$f_s = \begin{cases} f_{so}, f_{so} \leq distance \\ const, f_{so} \succ distance \end{cases} \tag{6}$$

Where *distance* represents the safe distance from the position of the obstacles in the working environment set in this paper; *Const* is a constant; f_{so} represents the shortest distance from the path of the mobile robot to all obstacles, and the definition of f_{so} is shown in Equation (7).

$$f_{so} = \min_{k=1}^{m}\left\{Dist^k \,\middle|\, Dist_{ij}^k = \sqrt{\left(PathX_i^k - so_j^x\right)^2 + \left(PathY_i^k - so_j^y\right)^2}\right\} \tag{7}$$

Where $i = 1, 2, \cdots, n$ represents the number of path nodes of each feasible path of the mobile robot; $j = 1, 2, \cdots, r$ is the number of obstacles in the working environment; $k = 1, 2, \cdots, m$ represents the number of feasible paths of the mobile robot; $Dist^k$ represents the distance of the mobile robot path nodes in each feasible path from the obstacles in the working environment.

The total length of the path *Path* can be expressed by Equation (8).

$$L\left(Path\right) = \sum_{i=0}^{n} d\left(P_i, P_{i+1}\right) \tag{8}$$

Where n denotes the number of path nodes, $d(P_i, P_{i+1})$ denotes the distance between the point P_i and the point P_{i+1}.

To further compare the performance advantages and disadvantages of the IMOABC algorithm in planning paths for mobile robots, path planning experiments were conducted with the single-objective artificial bee colony (ABC) algorithm under the same environmental map. The parameters in the ABC algorithm are set as follows: the bee colony size is 100 and the maximum number of iterations is 200. The cost equation is shown in Equation (9).

$$
\begin{aligned}
cost = & \sqrt{\left(X_g - X_i\right)^2 + \left(Y_g + Y_i\right)^2} \\
& + \left(collide * 5000 + oldpoint\_cost + collide\_to\_target\right) \\
& * 0.5 * sqrt\left(MM\hat{}2 + MM\hat{}2\right)
\end{aligned} \tag{9}
$$

Where X_i, Y_i are the coordinates of the food source and X_g, Y_g are the coordinates of the target point; *oldpoint_cost* is the value of the previously selected path; *collide_to_target* has two cases: Case 1: When a collision occurs, the collision-to-target value is equal to zero, and Case 2: When there is no collision, the collision-to-target value is equal to one.

2.3. Performance metrics for mobile robot path planning

This thesis evaluates the goodness of the feasible path obtained by the multi-objective optimization algorithm by the length of the path obtained by the optimization algorithm and the path safety.

2.3.1. Criteria for measuring path length

Among the many performance metrics for path planning, only the length size of the feasible path of the mobile robot optimized by the multi-objective optimization algorithm is considered. A long path is inferior to a short path.

2.3.2. Path safety metrics

Among the many performance metrics of path planning, only the path safety of the mobile robot optimized by the multi-objective optimization algorithm is considered. A path with a small path safety value is inferior to a path with a large path safety value. The formula for calculating the safety indicator for each route of the mobile robot is shown in Equation (10).

$$
S = min\left\{Dist \middle| Dist_{ij} = \sqrt{\left(PathX_i - so_j^x\right)^2 + \left(PathY_i - so_j^y\right)^2}\right\} \tag{10}
$$

Where $i = 1, 2, \cdots, n$, represents the number of path nodes; $j = 1, 2, \cdots, r$, represents the number of obstacles in the working map environment.

3. Improvement and validation of MOABC algorithm

3.1. Improvement of MOABC algorithm

While the MOABC algorithm has many advantages, it also has its disadvantages. First, since there is a greater demand for external archive space, the algorithm will have higher requirements for the objective environment. Second, although the algorithm takes all objective functions into account when optimizing, the degree of optimization of them will still be different in practical applications, which will make the results with a certain bias. Third, when solving multi-objective problems, the algorithm suffers from problems such as the tendency to fall into local optimality. In this paper, we propose an improved MOABC algorithm, namely IMOABC algorithm, for a series of problems existing in the above MOABC algorithm. Inspired by the ideas of algorithms such as NSGA-II and DNPSO, this paper combines the MOABC algorithm with a series of multi-objective optimization strategies (Deb et al., 2002; Zhou et al., 2017). It consists of four important strategies: the external archive pruning strategy, the non-dominated ranking and crowding distance strategy, the search strategy, and the Pareto optimal strategy to evaluate food source locations with multiple objectives and select non-dominated solutions, and is thus presented as follows.

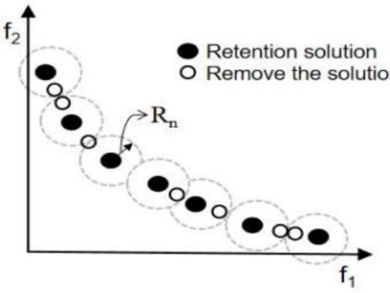

FIGURE 1
Schematic diagram of external file trimming.

3.1.1. External archive pruning strategy

To reduce the complexity of the algorithm computation and to obtain a uniformly distributed solution, the size of the external archive should not increase rapidly. Therefore, to prune external archive, this paper uses a technique based on the concept of domain radius for pruning external archive. The domain radius (Rn) of the non-dominated solution is considered and the Euclidean distance between the non-dominated solutions is calculated at the end of each iteration. If the distance between two non-dominated solutions is less than Rn, one of the non-dominated solutions is omitted. Figure 1 illustrates this process in the space of two objective functions.

3.1.2. Non-dominated sorting and congestion distance strategies

Inspired by the ideas of the NSGA-II algorithm, in this thesis we also introduce the concepts of fast non-dominated sorting and congestion distance (Deb et al., 2002). The individuals in the population that are not dominated by non-inferior solutions are defined as rank 1, then they are removed from the population and new non-inferior solutions are identified from the remaining individuals and defined as rank 2; the above process was repeated until all individuals in the population were given the appropriate rank. By storing the current solution and all individuals in the population in a hierarchical manner, the top-performing

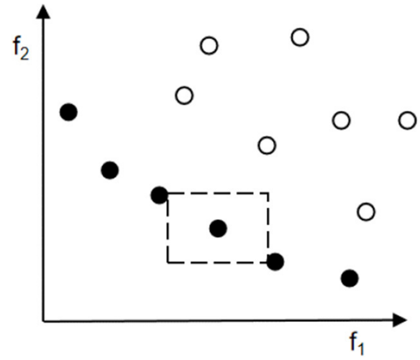

FIGURE 2
Diagram of congestion distance.

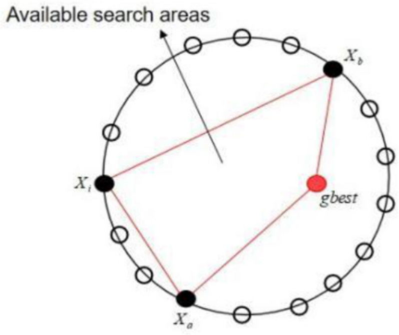

FIGURE 3
Global domain search operator.

individuals can have a higher probability of survival and can quickly improve the population ranking.

Once the number of non-inferior solutions exceeds the external archive size, the removal of individuals with small crowding distance in the external archive starts until the algorithm condition is satisfied (Akbari et al., 2012). As shown in Figure 2, regarding the two objective problems, the black dots in the figure are the non-inferior solutions in the population, and for the non-inferior solution x, the sum of the two sides of the rectangle consisting of $x + 1$ and $x - 1$ is computed; the final result is the crowding distance x_{dist} of the non-inferior solution x. The crowding distance of the boundary solutions (solutions with minimum and maximum objective function values) is infinite. When two solutions have the same rank, the superiority of the two solutions can be compared by the magnitude of the crowding distance, thus ensuring the convergence and diversity of the population. It can be seen that the individual x is superior to the individual y when and only when, $x_{rank} < y_{rank}$ or $x_{rank} = y_{rank}$ and $x_{dist} > y_{dist}$.

TABLE 1 Set each algorithm parameter.

Algorithm	Parameter setting
IMOABC	SN = 100 Limit = 100 maxCycle = 5,000 runtimes = 20
MOPSO	SN = 100 Limit = 100 maxCycle = 5,000 runtimes = 20 alpha = 0.1 beta = 4 gamma = 2
MOGWO	SN = 100 Limit = 100 maxCycle = 5,000 runtimes = 20 alpha = 0.1 beta = 4 gamma = 2
MOFA	SN = 100 Limit = 100 maxCycle = 5,000 runtimes =2 0
	Gamma = 2 beta 0 = 1 alpha = 0.1 alpha_damp = 0.9 mu = 0.1

TABLE 2 Statistical results of IGD value of 6 test functions of each algorithm.

Function		IMOABC	MOPSO	MOGWO	MOFA
UF1	Max	6.0977E-04	7.2238E-01	5.6043E-03	1.2877E-01
	Min	3.0968E-04	3.4987E-01	3.4772E-03	5.8557E-02
	Mean	3.9535E-04	5.8111E-01	4.5354E-03	9.6393E-02
	Std	9.8998E-05	1.1530E-01	6.8383E-04	2.6902E-02
UF2	Max	3.9503E-04	1.5286E-01	2.6193E-03	6.8151E-02
	Min	2.7110E-04	9.0947E-02	2.0654E-03	4.7688E-02
	Mean	3.1956E-04	1.1721E-01	2.2934E-03	5.7197E-02
	Std	4.0266E-05	1.6970E-02	1.4827E-04	6.3076E-03
UF3	Max	3.7099E-03	5.7834E-01	1.0777E-02	3.3310E-01
	Min	2.4375E-03	4.8281E-01	6.1589E-03	2.0697E-01
	Mean	2.9466E-03	5.4727E-01	9.2236E-03	3.0021E-01
	Std	3.3504E-04	2.6733E-02	1.5362E-03	4.1251E-02
UF4	Max	1.0123E-03	1.2913E-01	3.1823E-03	1.1771E-01
	Min	8.9674E-04	8.1743E-02	1.9092E-03	1.0491E-01
	Mean	9.4301E-04	1.0063E-01	2.1780E-03	1.1295E-01
	Std	3.8383E-05	1.3554E-02	3.7717E-04	3.9907E-03
UF5	Max	3.3773E-02	3.7239E+00	5.3183E-01	2.3188E-01
	Min	1.7787E-02	3.1088E+00	1.3546E-01	1.9165E-01
	Mean	2.3999E-02	3.3441E+00	2.9843E-01	2.0271E-01
	Std	4.6789E-03	1.7392E-01	1.3587E-01	1.2418E-02
UF6	Max	6.7739E-03	3.4929E+00	1.7238E-02	3.5559E-01
	Min	2.1384E-03	2.3373E+00	1.0387E-02	1.3182E-01
	Mean	4.3481E-03	3.0233E+00	1.2353E-02	1.7620E-01
	Std	1.3624E-03	3.4637E-01	1.8345E-03	6.6398E-02

3.1.3. Pareto optimal strategy

The Pareto solution set is a variety of feasible solutions for mobile robot path planning, which can also improve the efficiency of mobile robot path planning. Manual Pareto selection is subject to various subjective uncertainties, which can make the chosen solution not optimal. Fix a function η_i and the function η_i is denoted as

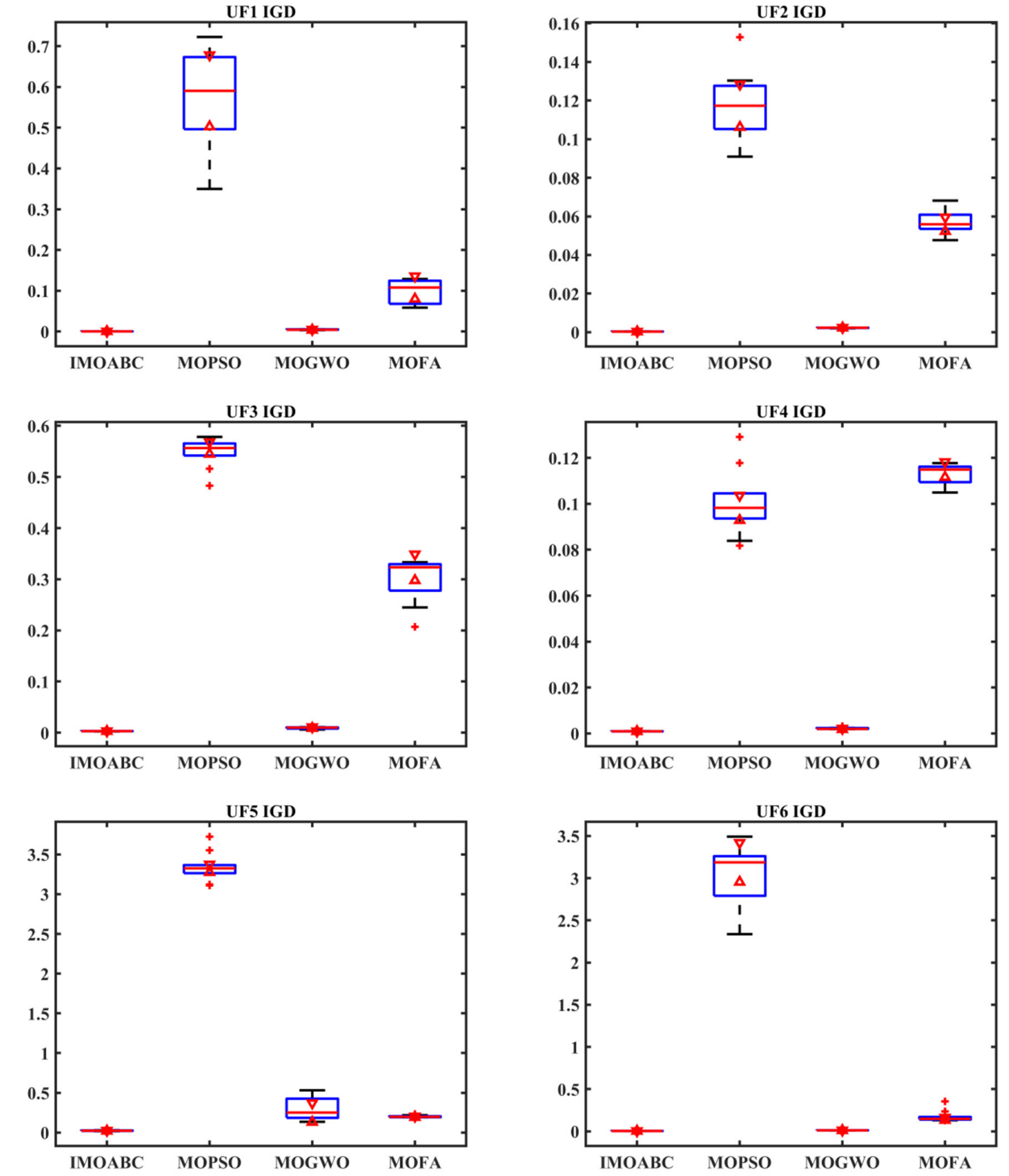

FIGURE 4
Test function IGD boxplot.

$$\eta_i = \begin{cases} 1, V_i \le V_i^{min} \\ \frac{V_i^{max} - V_i}{V_i^{max} - V_i^{min}}, V_i^{min} \le V_i \le V_i^{max} \\ 0, V_i \ge V_i^{max} \end{cases} \tag{11}$$

Where V_i^{max} and V_i^{min} are the maximum and minimum values of the ith objective function in the Pareto solution set, respectively; V_i is the ith objective function value. For each non-inferior solution k, the normalized affiliation function η_k is calculated by the following equation.

$$\eta_k = \frac{\sum_{i=1}^{Nob} \eta_i^k}{\sum_{j=1}^{M} \sum_{i=1}^{Nob} \eta_i^j} \tag{12}$$

Where M is the number of non-inferior solutions in the Pareto solution set and N_{ob} is the number of objective functions.

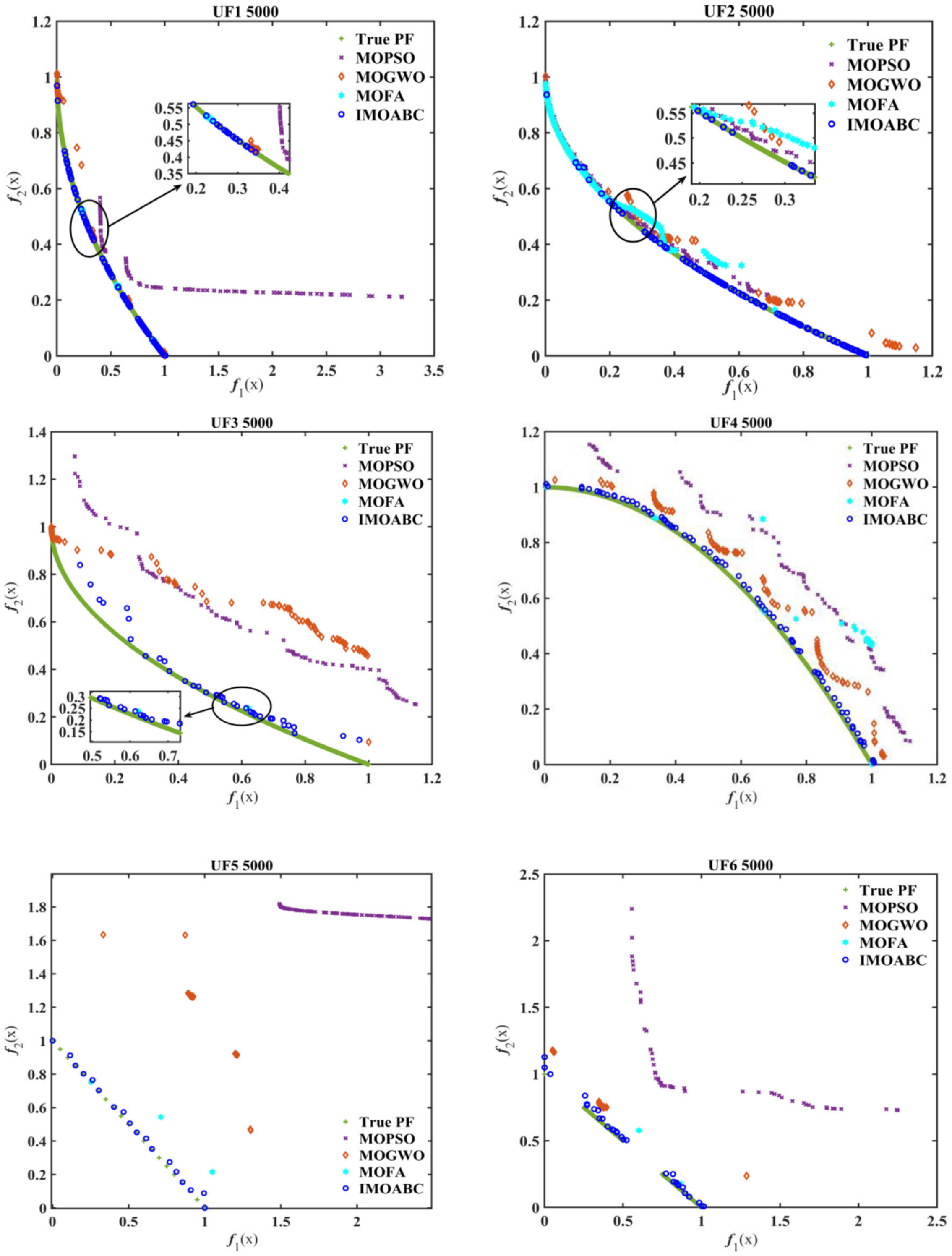

FIGURE 5
Test function Pareto frontier comparison graph.

The larger the value of η_k, the better the performance k of in coordinating multiple objective functions. The ranking of the Pareto solution set by the value of η_k gives the priority sequence of non-inferior solutions.

3.1.4. Search strategy

As one of the most representative variants, the DNPSO algorithm is characterized by the introduction of the global *NS* operator. Therefore, the search strategy

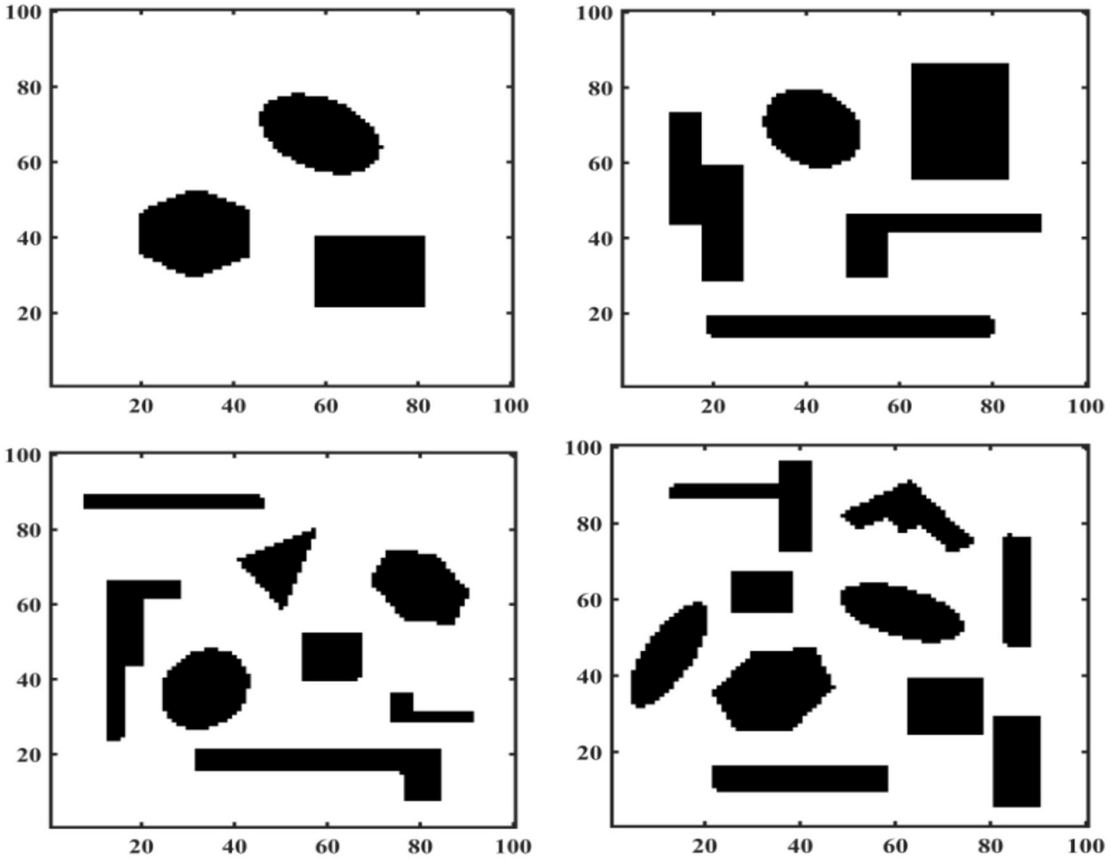

FIGURE 6
Mobile robot working environment.

in this paper also adds that operator, which can be expressed as:

$$v_i^j = r_1 x_i^j + r_2 x_{gbest} + r_3 \left(x_a^j - x_b^j \right) \tag{13}$$

Where r_1, r_2 and r_3 are three mutually exclusive numbers chosen at random from (0, 1), which must satisfy another condition: $r_1 + r_2 + r_3 = 1$; x_{gbest} is the global optimal solution for the whole population; and the indices *a* and *b* are mutually exclusive integers chosen at random from $(1, 2, \cdots, SN)$, which are different from the basic index *i*. It is important to note that r_1, r_2 and r_3 are regenerated in each generation, but they remain the same for all dimensions of each generation. Once an experimental solution has been generated, its associated food source must compete with it for entry into the next generation. This means that individuals with better fitness values have a chance to survive. An explicit illustration of the global *NS* operator is shown in Figure 3.

Thus, the basic flow of the IMOABC algorithm is shown below.

Step 1: Population initialization, randomly generate *SN* feasible solutions according to equation (1).
Step 2: Calculate the objective function values for all individuals and complete the initialization of the external archive set.
Step 3: In the employed bees stage, the employed bees guide the current domain search according to Equation (13).
Step 4: During the onlooker bees stage, the onlooker bees randomly select the nectar source according to Equation (3).
Step 5: Add all non-dominated solutions of the current population to the external archive according to the external archive pruning strategy.
Step 6: During the scout bees stage, a nectar source is randomly selected according to Equation (1).
Step 7: When the number of non-inferior solutions exceeds the *limit* number of external archives, fast non-dominated sorting is performed on solutions in external archives using the grouping of solutions and the
Step 8: Determine if the termination condition is met. If yes, output the Pareto front; otherwise go to step 3.
Step 9: The set of Pareto solutions is sorted by the Pareto optimal strategy, and then a priority list of solutions is derived.

3.2. Performance testing and analysis of an IMOABC algorithm

3.2.1. Test functions and performance metrics

To test the IMOABC algorithm proposed in this paper, simulation tests will be carried out for the single-objective

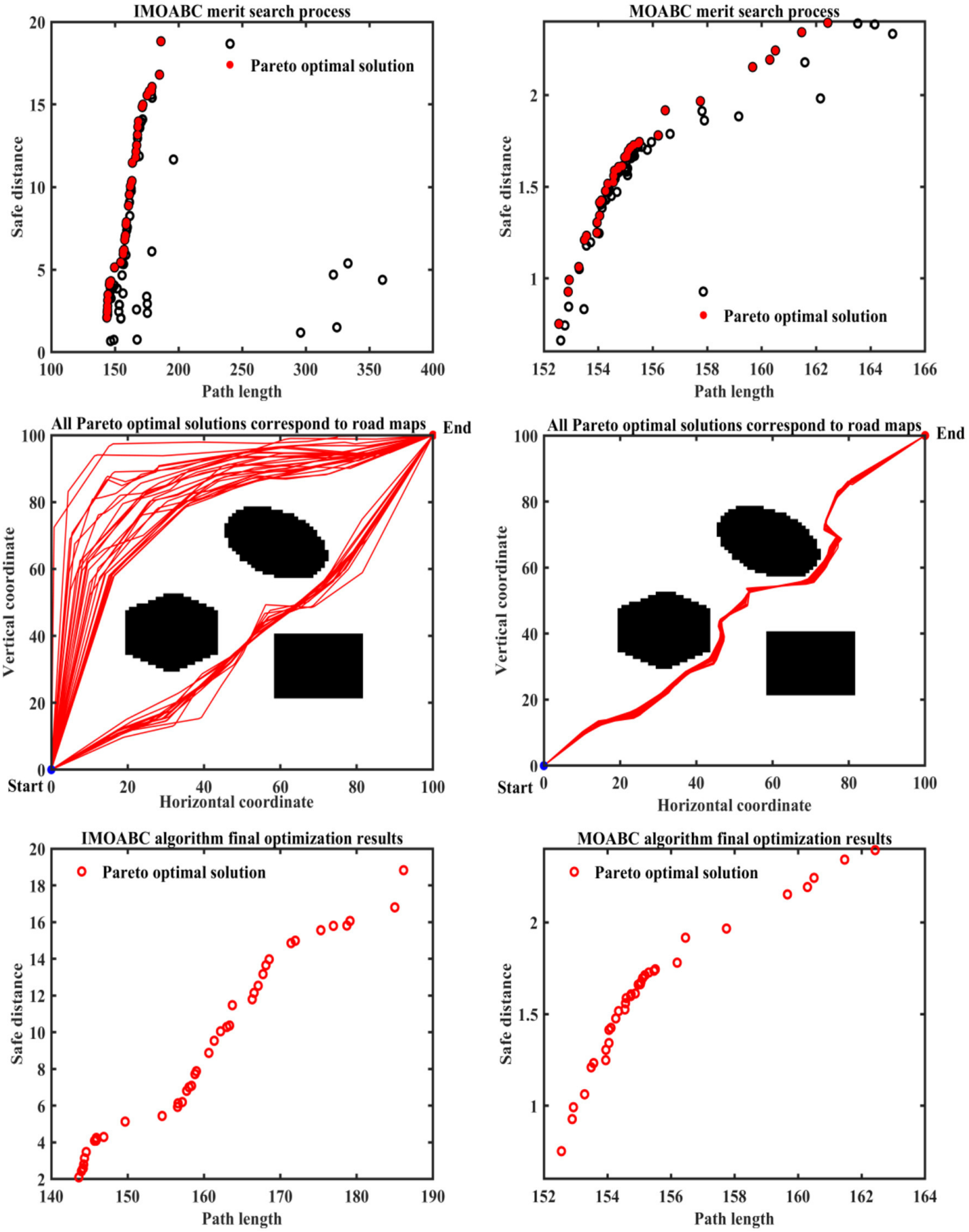

FIGURE 7
Experimental results of path planning based on IMOABC algorithm and MOABC algorithm in map 1 for case 1.

optimization problem. In this paper, six standard test functions UF1-UF6 are selected, and all six test functions have two optimization objectives. The inverse generation distance (IGD) is chosen as the evaluation criterion to compare the performance of the algorithms. The inverse generation distance (IGD) is used to evaluate the comprehensive performance of the algorithm (convergence and distributivity) and is defined as (Durillo and Nebro, 2011).

$$IGD\left(P,Q\right)=\frac{\sum_{v\in P}d\left(v,Q\right)}{\left|P\right|} \tag{14}$$

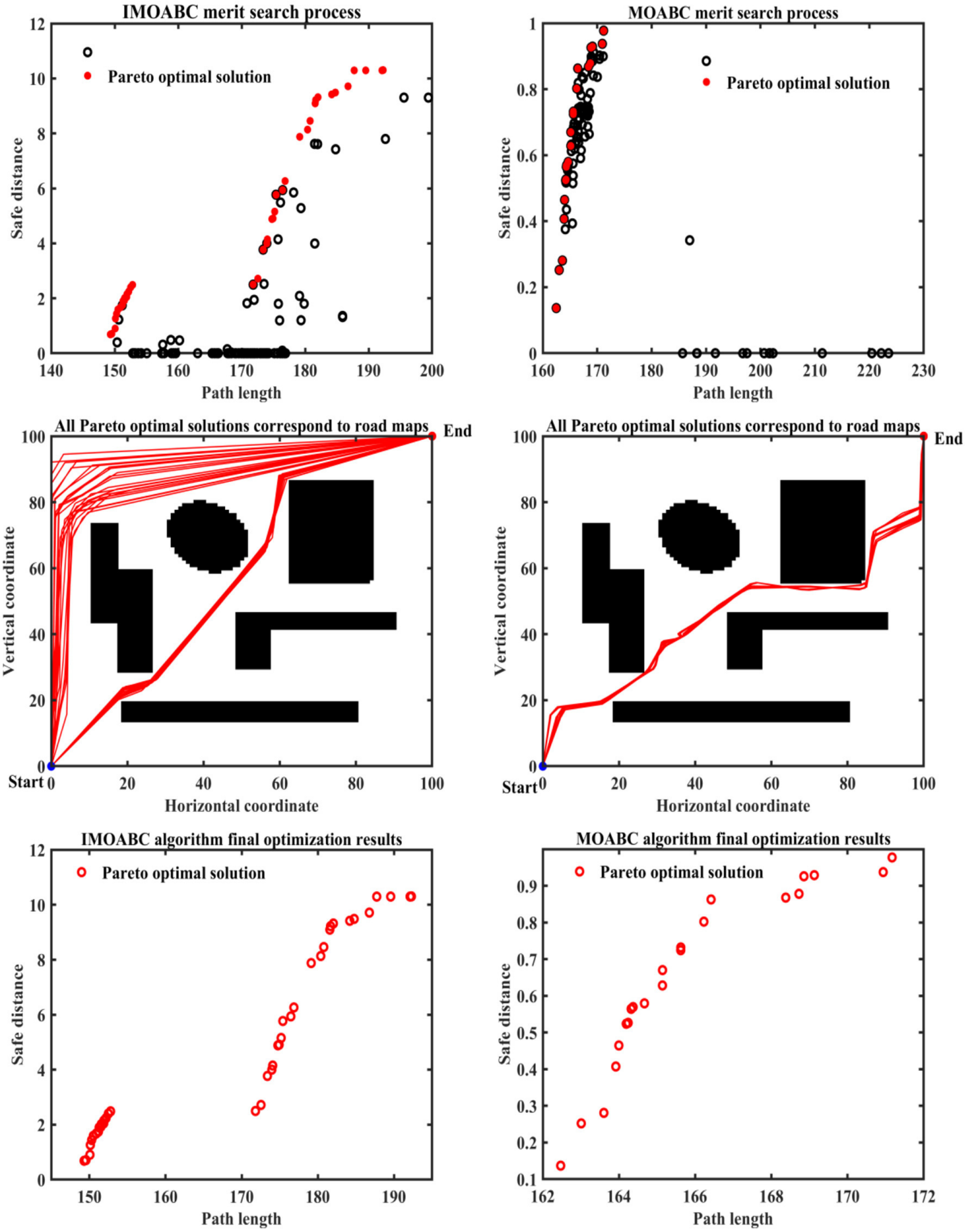

FIGURE 8
Experimental results of path planning based on IMOABC algorithm and MOABC algorithm in map 2 for case 1.

Where P is the set of points on the real Pareto front end and $|P|$ is the number of P. Q is the set of Pareto optimal solutions. And $d(v, Q)$ is the minimum Euclidean distance from the individual to the population Q in P. The smaller the value, the better the overall performance of the algorithm.

3.2.2. Test function simulation results and analysis

To test the performance of the IMOABC algorithm proposed in this paper, three typical multi-objective algorithms will be selected for comparison, i.e., they will be compared with MOPSO, MOGWO, and MOFA. To ensure the fairness of the algorithm testing, the parameter settings for each algorithm are given in

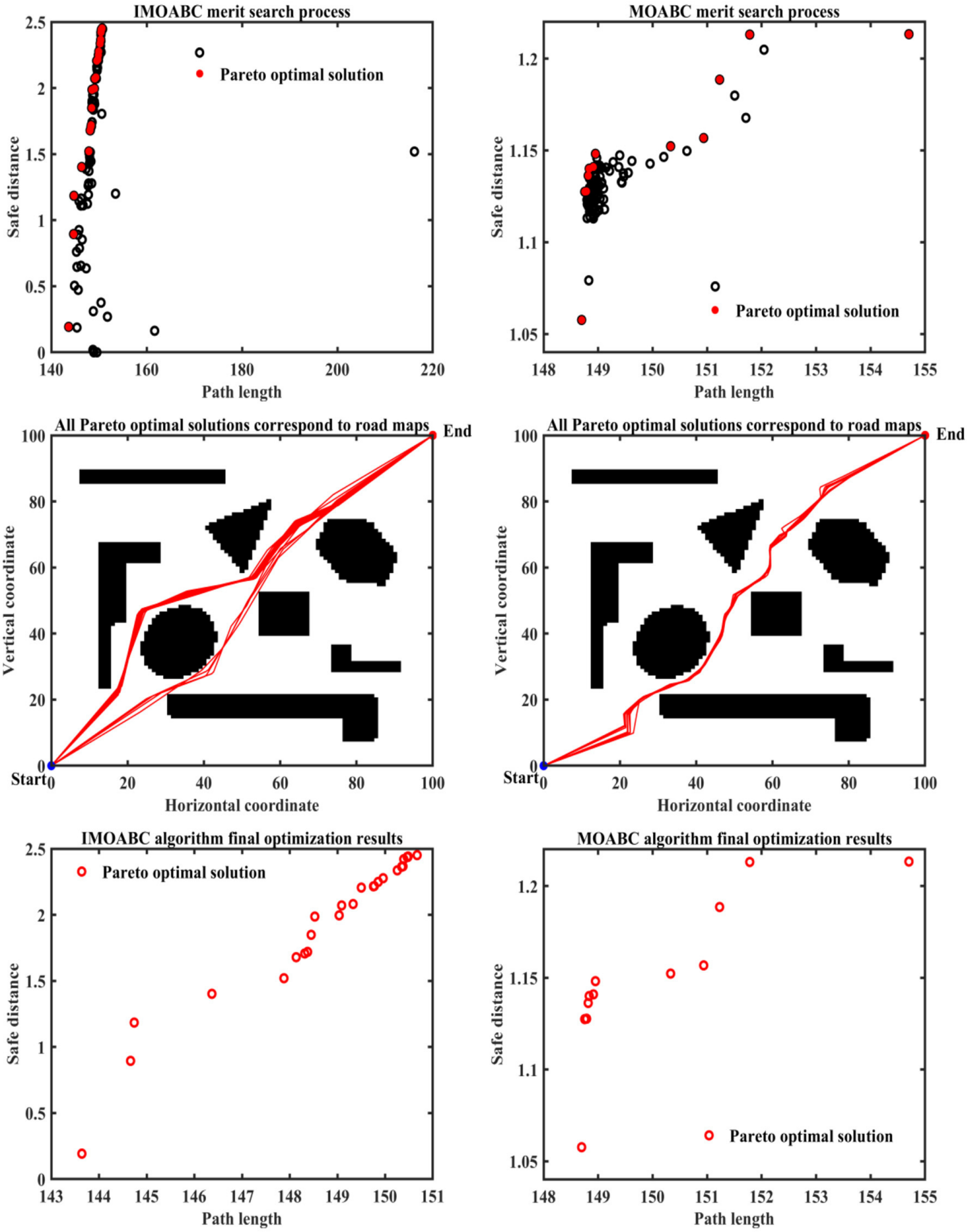

FIGURE 9
Experimental results of path planning based on IMOABC algorithm and MOABC algorithm in map 3 for case 1.

Table 1, and the algorithms are run 20 times separately and independently to reduce any unexpected errors in the algorithm's operation. By counting and comparing the maximum value (Max), minimum value (Min), mean value (Mean), and standard deviation (Std) of IGD for these four algorithms, the results are shown in Table 2, where the bolded items are the optimal values. At the same time, to have a better feeling of the performance comparison between each algorithm, we introduce box plots, which can more intuitively show the dispersion of the performance index data between each algorithm. We then plotted the values of the IGD metrics in Table 2 separately into the box plots shown in Figure 4.

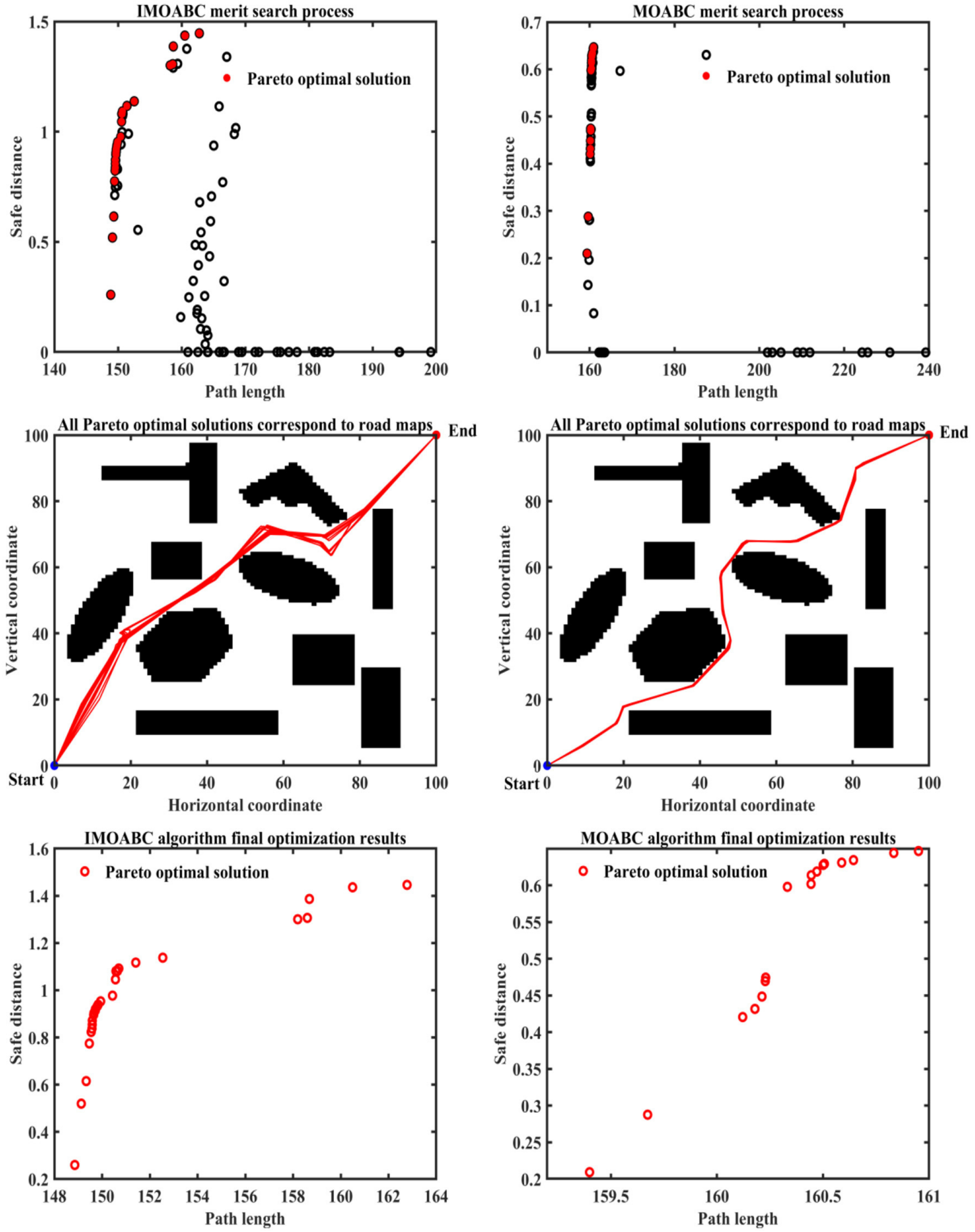

FIGURE 10
Experimental results of path planning based on IMOABC algorithm and MOABC algorithm in map 4 for case 1.

From the IGD of the integrated performance metrics in Table 2 and Figure 4, it can be seen that for these six test functions, the IMOABC algorithm outperforms the other three algorithms compared to the other three algorithms, and its IGD data values of integrated performance metrics are more concentrated. This shows that the IMOABC algorithm is also more stable in solving multi-objective optimization problems, further demonstrating the effectiveness of the IMOABC algorithm.

To visually and compare the performance of each algorithm, Figure 5 gives a comparative plot of the Pareto frontier for these four algorithms solving the six standard test functions.

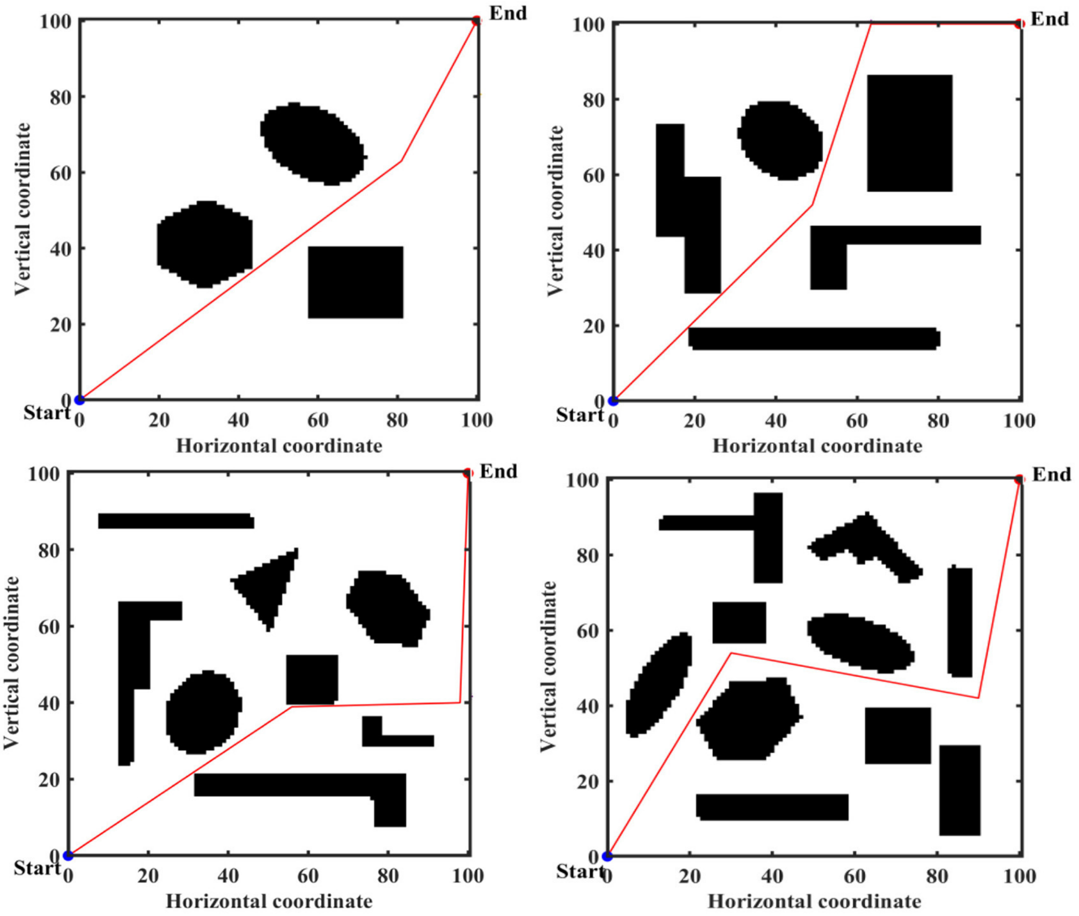

FIGURE 11
Path planning results of mobile robot in 4 different environments based on ABC algorithm for case 1.

From Figure 5 we can see more intuitively that the approximate Pareto front obtained by the IMOABC algorithm is closer to the real Pareto front than the other three algorithms. In particular, for the UF5 problem, the IMOABC algorithm shows a large improvement in the combined performance metrics compared to the remaining three algorithms. This further demonstrates that the IMOABC algorithm proposed in this paper not only yields a better approximation of the frontier, but also its convergence and distributivity metrics are superior and have been improved.

4. Simulation verification

4.1. Establishment of mobile robot optimization environment

Before a mobile robot searches for a path, it is important to first model the workspace in which it is located to facilitate identification and decision-making by the mobile robot. Environment modeling refers to the formal description of the realistic environment of a mobile robot through feature analysis and then replacing the obstacle information into a language that the computer can understand using a suitable approach. A variety of methods for environment modeling exist in the world, and the three most basic and commonly used methods are the raster method, the geometric feature method, and the topological method (Zhang et al., 2018).

There are advantages and disadvantages to each of these three approaches to constructing a work environment map. The object of this paper is mainly for land mobile robots, whose working environment is a two-dimensional plane, so the motion space of mobile robots is represented by a two-dimensional coordinate system. The mobile robot is represented by a mass point, and an image in BMP format is set up as the actual working environment of the mobile robot. The size of this BMP is 100*100, the black part is the obstacle area and the white part is the feasible area. The obstacle locations and shapes in the four different BMPs are shown in Figure 6. The obstacle shapes in the BMPs are set to geometric polygons to facilitate verification of the effectiveness of IMOABC algorithm-based path planning in complex and variable environments.

4.2. Experimental simulations and algorithm comparison

The effectiveness of the IMOABC algorithm proposed in this thesis has been confirmed in the previous section, so this section

TABLE 3 The first experiment is to compare the path planning data between each algorithm.

Performance indicators	Working environment		IMOABC	MOABC	ABC
Path length	Map 1	Max	186.144	162.428	**144.2091**
		Min	**143.605**	152.553	144.2091
	Map 2	Max	192.274	171.17	**158.1544**
		Min	**149.317**	162.469	158.1544
	Map 3	Max	**150.671**	154.703	170.2625
		Min	**143.641**	148.697	170.2625
	Map 4	Max	162.777	**160.95**	181.8178
		Min	**148.855**	159.4	181.8178
Path security	Map 1	Max	**18.827**	2.39384	0.588
		Min	**2.09754**	0.752478	0.588
	Map 2	Max	**10.3018**	0.977297	0.0868
		Min	**0.686048**	0.136834	0.0868
	Map 3	Max	**2.45146**	1.21333	0.238
		Min	0.19272	**1.05764**	0.238
	Map 4	Max	**1.44622**	0.646673	0.996
		Min	0.259739	0.209394	**0.996**
Number of possible routes	Map 1		42	36	1
	Map 2		43	11	1
	Map 3		26	12	1
	Map 4		27	17	1

The meaning of the bold values is the optimal value.

TABLE 4 The second experiment is to compare the path planning data between each algorithm.

Performance indicators	Working environment		IMOABC	MOABC	ABC
Path length	Map 1	Max	183.036	166.022	**150.4454**
		Min	**146.967**	155.866	150.4454
	Map 2	Max	200	166.033	**151.1716**
		Min	**142.005**	146.41	151.1716
	Map 3	Max	199.679	**149.641**	173.1685
		Min	**144.449**	146.547	173.1685
	Map 4	Max	193.988	161.705	**159.2363**
		Min	**150.584**	159.319	159.2363
Path security	Map 1	Max	**18.6376**	4.09792	0.9205
		Min	**1.11984**	0.974716	0.9205
	Map2	Max	**10.2965**	2.83367	0.575
		Min	0.00602009	0	**0.575**
	Map 3	Max	**8.46905**	1.04332	1.115
		Min	0	0.479438	**1.115**
	Map 4	Max	**3.32034**	0.890569	0.0625
		Min	0	**0.652126**	0.0625
Number of possible routes	Map 1		**39**	29	1
	Map 2		**25**	4-15	1
	Map 3		**32**	27	1
	Map 4		14	**30**	1

The meaning of the bold values is the optimal value.

will explore the application of the IMOABC algorithm to the mobile robot path planning problem and verify its effectiveness. Here, we have conducted simulation experiments using Matlab 2018b, and the two selected objective functions are path length and path safety. Firstly, the effectiveness of the IMOABC algorithm applied to mobile robot path planning in a two-dimensional working environment with complex and variable obstacles is verified, followed by a comparative analysis with path planning based on the MOABC algorithm and ABC algorithm. Also, to verify the robustness of the algorithm, two experiments were conducted in the same experimental environment by changing the starting and ending coordinates of the mobile robot path planning, with the starting coordinates of case 1 being (0, 0) and the ending coordinates being (100, 100); the starting coordinates of case 2 being (0, 100) and the ending coordinates being (100, 0).

4.2.1. Simulation and verification for case 1

To verify the effectiveness of IMOABC algorithm-based path planning, path planning experiments are first conducted in four environments with different obstacles. The initialization algorithm parameters are: the bee colony size is set to 100, the maximum number of iterations of the algorithm is set to 200, the four different 2D working environment pixels constructed using BMPs are uniformly set to 100*100, and the starting coordinates of the experimentally set path planning are (0, 0) and the ending coordinates are (100, 100). The results are shown in Figures 7–11, where the mobile robot harvested several feasible routes. Mobile robots can select routes from planning results based on environmental conditions and their preferences. As can be seen, the mobile robot can perfectly avoid complex and variable obstacles and find the minimum path distance from the starting point to the endpoint.

From Table 3 and Figures 7–11 we can find that the path length of the mobile robot based on the IMOABC algorithm is the shortest in all four environments. That is, the path length performance of path planning based on the IMOABC algorithm is better than the path length performance of path planning based on the MOABC algorithm and ABC algorithm.

From Table 3 and Figures 7–11 we can find that the path safety of the IMOABC algorithm-based path planning is maximum in the map 1 and map 2 environments. That is, the path safety of path planning based on the IMOABC algorithm is better than the path safety performance of both MOABC algorithm-based and ABC algorithm-based path planning. In the map 3 environment, we can find from Figure 9 that the safety performance of path planning based on the IMOABC algorithm is the most superior.

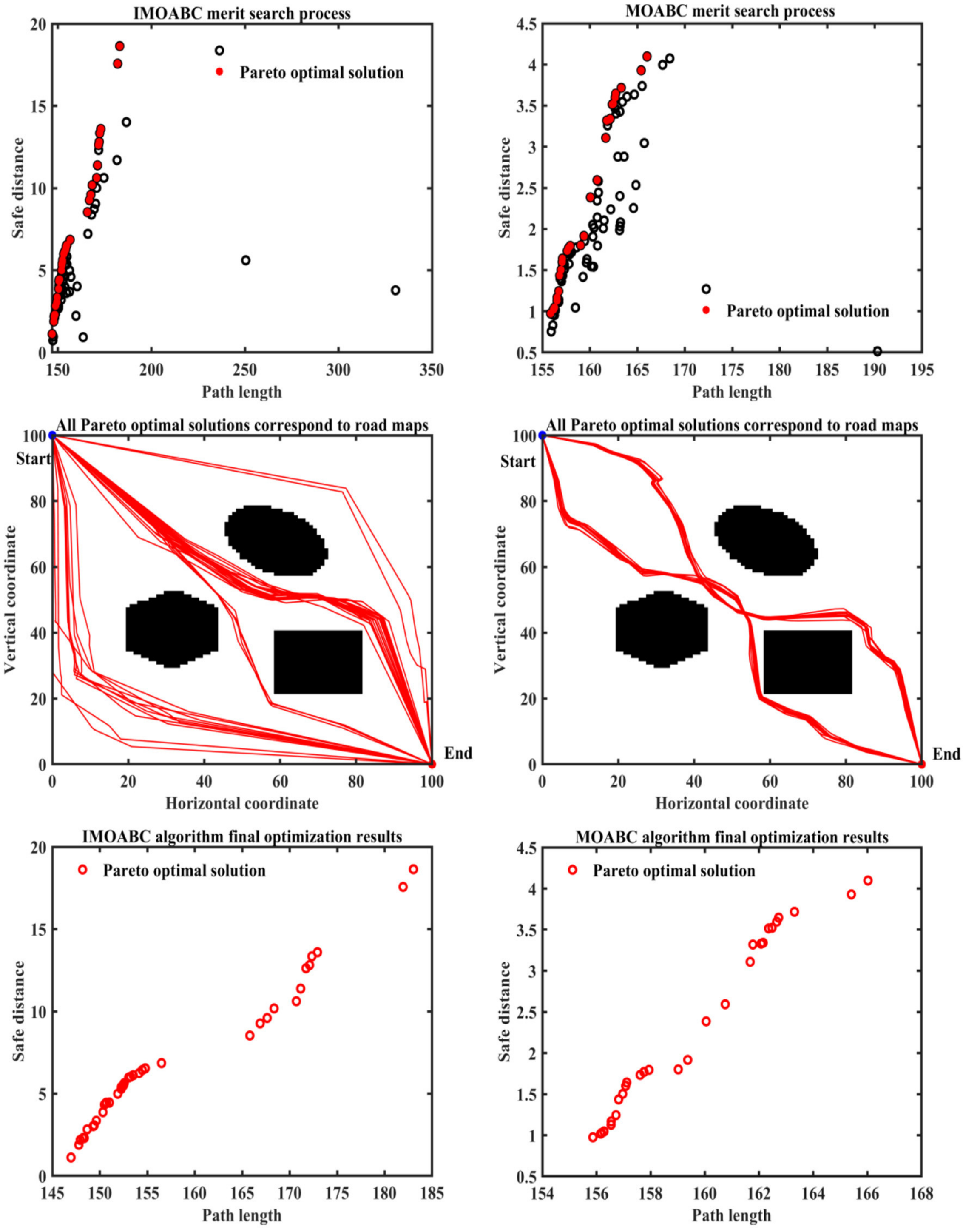

FIGURE 12
Experimental results of path planning based on IMOABC algorithm and MOABC algorithm in map 1 for case 2.

In the map 4 environment, the path safety of path planning based on the IMOABC algorithm is higher than the path safety of path planning based on the MOABC algorithm and slightly inferior to the path safety of path planning based on the ABC algorithm when the path is shortest, but the shortest path of path planning based on ABC algorithm is much inferior to the shortest path of path planning based on IMOABC algorithm. In summary, it is still the IMOABC algorithm-based path planning that has the best safety performance.

From Table 3 and Figures 7–11 we can find that the number of feasible routes for path planning based on the IMOABC algorithm is more than that of the MOABC-based algorithm

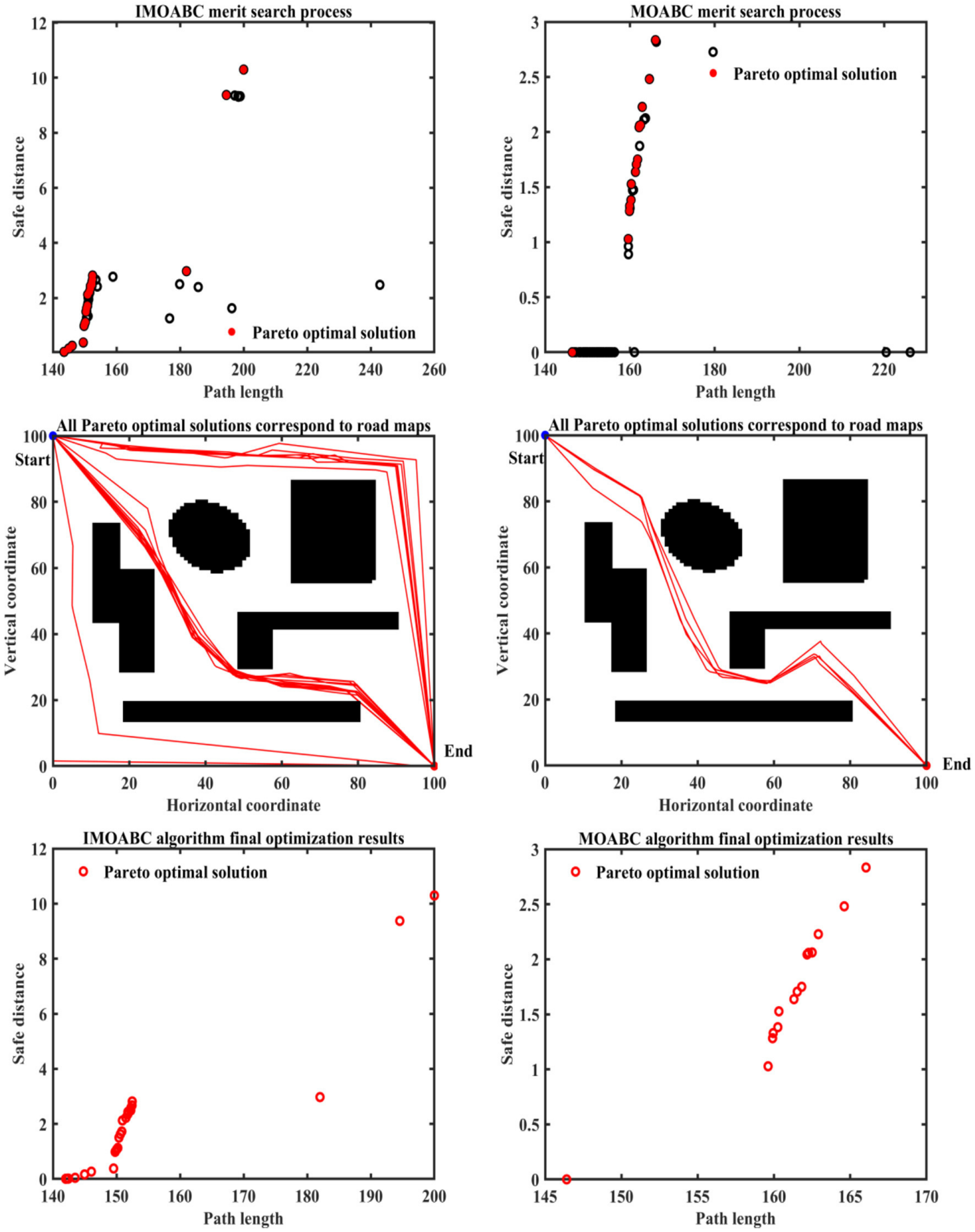

FIGURE 13
Experimental results of path planning based on IMOABC algorithm and MOABC algorithm in map 1 for case 2.

and ABC-based algorithm in these four environments. And the IMOABC algorithm-based path planning feasible routes are a little more dispersed than the MOABC algorithm and ABC algorithm-based path planning feasible routes, which means that the IMOABC algorithm-based path planning has more options in choosing feasible routes according to their situation. None of the path planning based on the MOABC algorithm in the map 2 environment can completely draw the number of feasible routes. Therefore, the IMOABC algorithm-based path planning for mobile robots is the most superior in terms of the number of feasible routes.

In summary, the experimental results demonstrate the superiority of IMOABC algorithm-based path planning for mobile

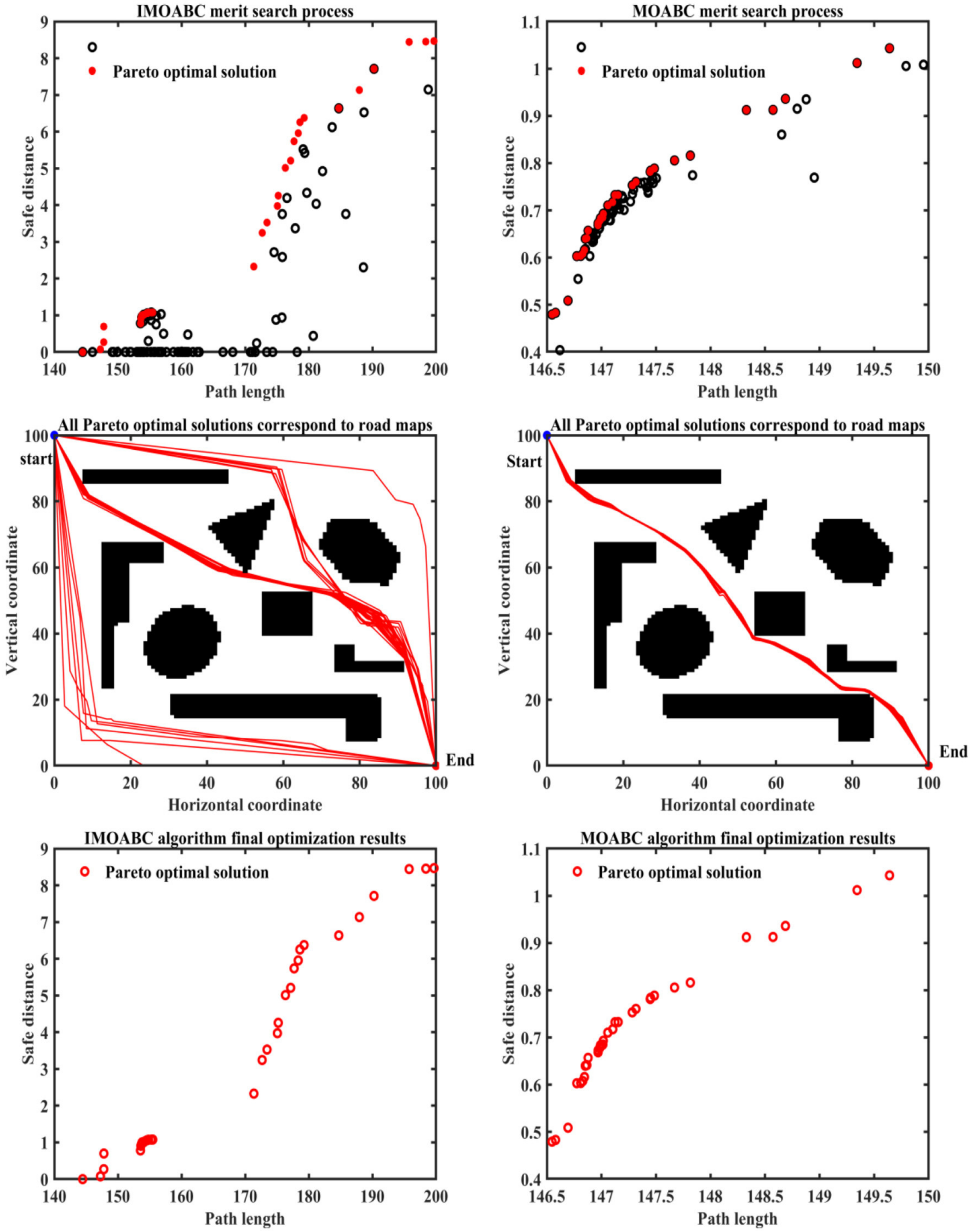

FIGURE 14
Experimental results of path planning based on IMOABC algorithm and MOABC algorithm in map 3 for case 2.

robots in terms of three aspects: path length, path safety, and the number of feasible routes.

4.2.2. Simulation and verification for case 2

To verify the robustness of the IMOABC algorithm applied to the mobile robot, we conduct a second experiment on the mobile robot, where the starting coordinates are set to (0, 100) and the ending coordinates are (100, 0). The results of the path planning experiment are as follows.

From Table 4 and Figures 12–16 we can find that the path length of the IMOABC algorithm-based path planning is the shortest in all four environments. That is, the path length

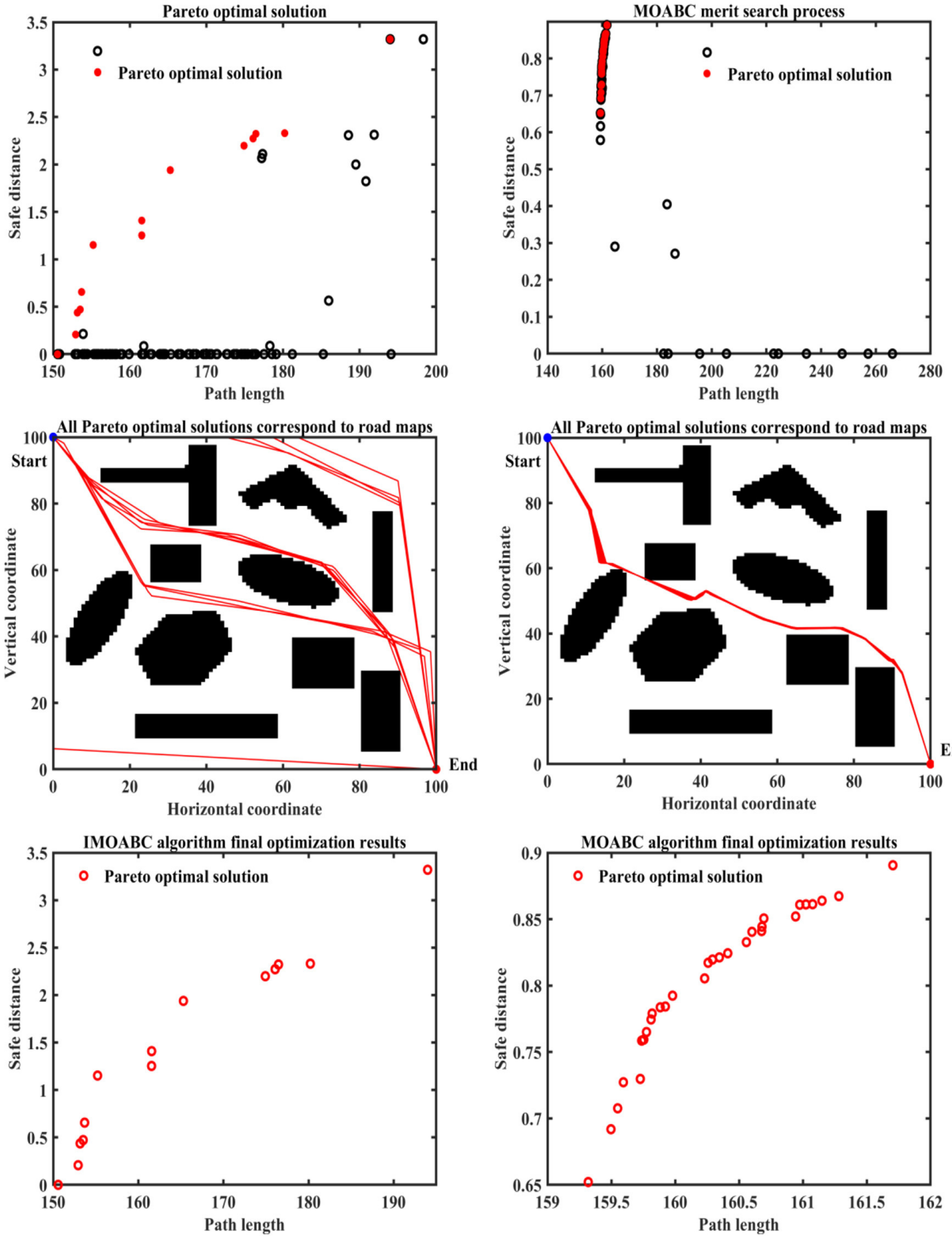

FIGURE 15
Experimental results of path planning based on IMOABC algorithm and MOABC algorithm in map 4 for case 2.

performance of path planning based on the IMOABC algorithm is better than the path length performance of path planning based on the MOABC algorithm and ABC algorithm.

From Table 4 and Figures 12–16 we can find that the security of IMOABC algorithm-based path planning is highest only in the map 1 environment for the shortest path length. That is, the path safety of path planning based on the IMOABC algorithm is better than the path safety of path planning based on the MOABC algorithm and ABC algorithm. In the map 2 environment, it is clear from Figure 13 that the safety performance of the IMOABC algorithm-based path planning is the most superior. In the map 3 environment, the security of path planning based on

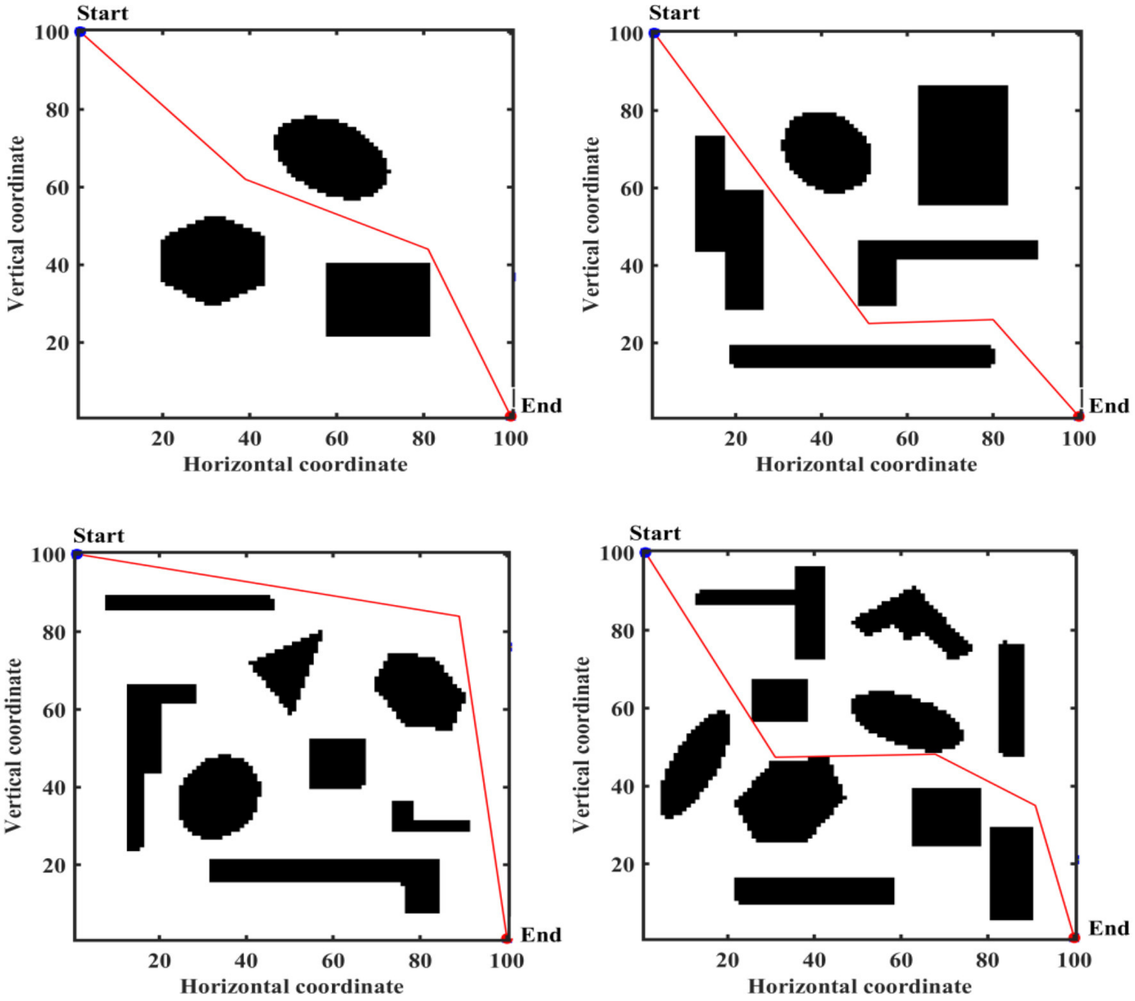

FIGURE 16
Path planning results of mobile robot in 4 different environments based on ABC algorithm for case 2.

the IMOABC algorithm is not as good as the security of path planning based on the MOABC algorithm, but better than the security of path planning based on the ABC algorithm. In the map 4 environment, the best safety performance of path planning based on the IMOABC algorithm can be found in Figure 15. In summary, we can find that the path safety of IMOABC algorithm-based path planning is generally better than the security of MOABC and ABC algorithm-based path planning.

From Table 4 and Figures 12–16 we can find that the number of feasible routes for path planning based on the IMOABC algorithm is more than that based on the MOABC algorithm and ABC-based algorithm in map 1, map 2, and map 3 environments. And the IMOABC algorithm-based path planning feasible routes are a little more dispersed than the MOABC and ABC algorithm-based path planning feasible routes, which means that the IMOABC algorithm-based path planning has more options in choosing feasible routes according to its situation. In the map 4 environment, although the number of feasible routes of IMOABC algorithm-based path planning is inferior to that of MOABC algorithm-based path planning in the experiment, the feasible routes of MOABC algorithm-based path planning are very concentrated, which is inferior to the feasible routes of IMOABC algorithm-based path planning that can provide many choices, so IMOABC algorithm-based path planning for mobile robots is still the most superior in terms of the number of feasible routes.

In summary, we can find that the experimental results still demonstrate the superiority of the IMOABC algorithm-based mobile robot path planning and its robustness in terms of path length, path safety, and the number of feasible routes when the starting and ending points of the mobile robot are changed.

5. Conclusion

This thesis focuses on optimizing the path-planning problem of mobile robots based on the IMOABC algorithm. Mobile robot path planning is a complex optimization problem. First of all, it is necessary to ensure that the mobile robot does not collide with obstacles in the working environment map, but also to ensure that the mobile robot can go from the starting point to the endpoint smoothly and that the performance indicators of the path planning should be as good as possible. In this thesis, an IMOABC algorithm is proposed in conjunction with the MOABC algorithm for solving mobile robot path planning problems. Firstly, the external archive pruning strategy, non-dominated sorting and crowding distance strategy, search strategy, and Pareto optimal strategy are introduced into the MOABC algorithm, which leads to the IMOABC algorithm. The IMOABC algorithm is also compared with three other similar algorithms for verification, and the results show the effectiveness and stability of the IMOABC algorithm. The IMOABC algorithm is then applied to mobile robot path planning based on two performance optimization metrics selected for path planning, and the results show that the algorithm can complete the given path planning task. And compare the experiments with two other algorithms (MOABC algorithm and ABC algorithm) on the Matlab platform to verify the effectiveness and efficiency of the IMOABC algorithm in mobile robot path planning. Our current limitation is simulation validation, and future work will be conducted in physical experiments.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

QC completed the improvement of the algorithm and analysis. QC and HD wrote the first draft. XM proposed the idea and directed the experiment. PL assisted in the debugging of the algorithm. HW provided facilities and equipment. XM helped the complete the revised manuscript. All authors contributed to the article and approved the submitted version.

Funding

This study was fully supported by Liaoning Provincial Department of Education Basic Research Projects for Higher Education Institutions, China (No. LJKZ0301); the Scientific Research Foundation of the Education Department of Liaoning Province, China (No. 2017LNQN22); and Young Teachers Foundation of University of Science and Technology Liaoning, China (No. 2017QN04).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Akbari, R., Hedayatzadeh, R., Ziarati, K., and Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm Evol. Comput.* 2, 39–52. doi: 10.1016/j.swevo.2011.08.001

Chen, W., Wu, X., and Lu, Y. (2015). An improved path planning method based on artificial potential field for a mobile robot. *Cybernet. Inform. Technol.* 15, 181–191. doi: 10.1515/cait-2015-0037

Contreras-Cruz, M. A., Ayala-Ramirez, V., and Hernandez-Belmonte, U. H. (2015). Mobile robot path planning using artificial bee colony and evolutionary programming. *Appl. Soft Comput.* 30, 319–328. doi: 10.1016/j.asoc.2015.01.067

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 182–197. doi: 10.1109/4235.996017

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybernet Part B)* 26, 29–41. doi: 10.1109/3477.484436

Durillo, J. J., and Nebro, A. J. (2011). jMetal: a Java framework for multi-objective optimization. *Adv. Eng. Softw.* 42, 760–771. doi: 10.1016/j.advengsoft.2011.05.014

Fakoor, M., Kosari, A., and Jafarzadeh, M. (2016). Humanoid robot path planning with fuzzy Markov decision processes. *J. Appl. Res. Technol.* 14, 300–310. doi: 10.1016/j.jart.2016.06.006

Gammell, J. D., Srinivasa, S. S., and Barfoot, T. D., (2014). Informed, R. R. T.:, "Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Chicago, IL, USA: IEEE), 2997–3004. doi: 10.1109/IROS.2014.6942976

Ghatee, M., and Mohades, A. (2009). Motion planning in order to optimize the length and clearance applying a Hopfield neural network. *Expert Syst. Appl.* 36, 4688–4695. doi: 10.1016/j.eswa.2008.06.040

Gunji, B., Deepak, B., Saraswathi, M. B. L., Mogili, U. R. (2019). Optimal path planning of mobile robot using the hybrid cuckoo-bat algorithm in assorted environment. *Int. J. Intell. Unmanned Syst.* 7, 35–52. doi: 10.1108/IJIUS-07-2018-0021

Hedayatzadeh, R., Hasanizadeh, B., Akbari, R., and Ziarati, K. (2010). "A MOABC for optimizing multi-objective problems," in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Vol. 5(Chengdu, China: IEEE), V5-277–V5-281.

Hosseininejad, S., and Dadkhah, C. (2019). Mobile robot path planning in dynamic environment based on cuckoo optimization algorithm. *Int. J. Adv. Robot. Syst.* 16, 1729881419839575. doi: 10.1177/1729881419839575

Hou, P., Pan, H., and Guo, C. (2017). Simulation research for mobile robot path planning based on improved artificial potential field method recommended by the AsiaSim. *Int. J. Model. Simul. Sci. Comput.* 8, 17500465. doi: 10.1142/S1793962317500465

Jeddisaravi, K., Alitappeh, R. J., and Guimarães, F. G. (2016). "Multi-objective mobile robot path planning based on a search," in *2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)* (Mashhad, Iran: IEEE), 7–12. doi: 10.1109/ICCKE.2016.7802107

Karaboga, D. (2005). *An Idea Based on Honey Bee Swarm for Numerical Optimization*. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey.

Karaman, S., and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *Int. J. Rob. Res.* 30, 846–894. doi: 10.1177/0278364911406761

Kuffner, J. J. (2000). "LaValle S M. RRT-connect: an efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Vol. 2 (San Francisco, CA, USA: IEEE), 995–1001.

Lamini, C., Benhlima, S., and Elbekri, A. (2018). Genetic algorithm based approach for autonomous mobile robot path planning. *Proc. Comput. Sci.* 127, 180–189. doi: 10.1016/j.procs.2018.01.113

LaValle, S. M., and Kuffner, J. J. Jr. (2001). Randomized kinodynamic planning. *Int. J. Robot. Res.* 20, 378–400. doi: 10.1177/02783640122067453

Lee, D., Jeong, J., Kim, Y. H., and Park, J. B. (2017). "An improved artificial potential field method with a new point of attractive force for a mobile robot," in *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)* (Shanghai, China: IEEE), 63–67. doi: 10.1109/ICRAE.2017.8291354

Liu, J., Chen, Z., Zhang, Y., and Li, W. (2020). "Path planning of mobile robots based on improved Genetic algorithm," in *Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence* (Shanghai: Association for Computing Machinery), 49–53. Available online at: https://dl.acm.org/doi/proceedings/10.1145/3438872

Liu, X., Jiang, W., Su, H., Qi, W., and Ge, S. S. A. (2022). Control strategy of robot eye-head coordinate gaze behavior achieved for minimized neural transmission noise. *IEEE/ASME Transact. Mechatr*. 28, 956–966. doi: 10.1109/TMECH.2022.3210592

Liu, Z., Yuan, X., Huang, G., Wang, Y., and Zhang, X. (2021). Two potential fields fused adaptive path planning system for autonomous vehicle under different velocities. *ISA Trans.* 112, 176–185. doi: 10.1016/j.isatra.2020.12.015

Mo, H., and Xu, L. (2015). Research of biogeography particle swarm optimization for robot path planning. *Neurocomputing* 148, 91–99. doi: 10.1016/j.neucom.2012.07.060

Mohamed, E. F., El-Metwally, K., and Hanafy, A. R. (2011). "An improved Tangent Bug method integrated with artificial potential field for multi-robot path planning," in *2011 International Symposium on Innovations in Intelligent Systems and Applications* (Istanbul, Turkey: IEEE), 555–559. doi: 10.1109/INISTA.2011.5946136

Nasir, J., Islam, F., Malik, U., and Hasan, Y (2013). RRT*-SMART: a rapid convergence implementation of RRT. *Int. J. Adv. Robot. Syst.* 10, 299. doi: 10.5772/56718

Nazarahari, M., Khanmirza, E., and Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Syst. Appl.* 115, 106–120. doi: 10.1016/j.eswa.2018.08.008

Qu, H., Xing, K., and Alexander, T. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing* 120, 509–517. doi: 10.1016/j.neucom.2013.04.020

Shi, Y., Li, H., Fu, X., Luan, R., Wang, Y., Wang, N., et al. (2023). Self-powered difunctional sensors based on sliding contact-electrification and tribovoltaic effects for pneumatic monitoring and controlling. *SSRN Electr. J.* 110. doi: 10.2139/ssrn.4340707

Wang, L., Yang, S. X., and Biglarbegian, M. (2012). "A fuzzy logic based bio-inspired system for mobile robot navigation," in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (Hamburg, Germany: IEEE), 219–224. doi: 10.1109/MFI.2012.6343040

Wang, S., Li, H., and Zhang, Y. (2016). Path planning of mobile robot based on genetic bee colony algorithm. *DEStech Transc Comput. Sci. Eng.* 16, 615–620. doi: 10.12783/dtcse/cmee2016/5357

Zhang, H., Lin, W., and Chen, A. (2018). Path planning for the mobile robot: a review. *Symmetry* 10, 450. doi: 10.3390/sym10100450

Zhou, X., Wang, H., Wang, M., and Wan, J. (2017). Enhancing the modified artificial bee colony algorithm with neighborhood search. *Soft Comput.* 21, 2733–2743. doi: 10.1007/s00500-015-1977-x