



Research article

Multi-objective path planning for mobile robot with an improved artificial bee colony algorithm

Zhenao Yu, Peng Duan*, Leilei Meng, Yuyan Han and Fan Ye

School of Computer Science, Liaocheng University, Liaocheng 52059, China

* **Correspondence:** Email: duanpeng@lcu.edu.cn.

Abstract: Effective path planning (PP) is the basis of autonomous navigation for mobile robots. Since the PP is an NP-hard problem, intelligent optimization algorithms have become a popular option to solve this problem. As a classic evolutionary algorithm, the artificial bee colony (ABC) algorithm has been applied to solve numerous realistic optimization problems. In this study, we propose an improved artificial bee colony algorithm (IMO-ABC) to deal with the multi-objective PP problem for a mobile robot. Path length and path safety were optimized as two objectives. Considering the complexity of the multi-objective PP problem, a well-environment model and a path encoding method are designed to make solutions feasible. In addition, a hybrid initialization strategy is applied to generate efficient feasible solutions. Subsequently, path-shortening and path-crossing operators are developed and embedded in the IMO-ABC algorithm. Meanwhile, a variable neighborhood local search strategy and a global search strategy, which could enhance exploitation and exploration, respectively, are proposed. Finally, representative maps including a real environment map are employed for simulation tests. The effectiveness of the proposed strategies is verified through numerous comparisons and statistical analyses. Simulation results show that the proposed IMO-ABC yields better solutions with respect to hypervolume and set coverage metrics for the later decision-maker.

Keywords: path planning; multi-objective optimization; improved artificial bee colony algorithm; hybrid initialization strategy; variable neighborhood local search

1. Introduction

Mobile robots have been gaining widespread attention as intelligent products. They are applied in many fields, such as industrial manufacturing, agricultural production, medical services, smart home, and automatic driving [1,2]. In our daily life, home service robot is a type of mobile robot that performs a wide range of tasks, including patrolling, taking care of the elderly and children, delivering goods to

the server, and so on [3–5]. Thus, path planning (PP) has become essential for home service robots to complete service tasks. PP problem involves finding one or more feasible paths to the target location from the starting position with a given configuration space. In the real application, robust PP methods could minimize the robot's operation time, while saving labor and material resources. Currently, PP has been one of the most researched problems in robot technology [6, 7].

The classical methods for solving the PP problem include the probabilistic roadmap planner (PRM) [8], rapidly exploring random tree (RRT) algorithm and its variants [9, 10], artificial potential field methods [11], A* algorithm [12] and Dijkstra algorithm [13]. These algorithms have achieved some typical applications in many different fields. However, most of them optimize only one objective, such as path length. Real scenarios bring many constraints, including dangerous obstacles in the environment and structural constraints of the mobile robot itself. Hence, a solution that considers only a single objective is not always accurate. In our opinion, path length and other objectives (e.g., path safety) should be simultaneously considered. The path length is related to the motion time of the mobile robot, from the starting point to the target point. Path safety is related to the concept of a collision-free path that represents a path's safety degree. The solution of a single-objective PP problem is uniquely determined, while a multi-objective PP problem yields a set of solutions with advantages and disadvantages that cannot be directly compared. These solutions are called the Pareto optimal solution set [14–16].

Swarm intelligence optimization algorithms [17–22] are currently attracting wide attention from scholars, who have applied them to deal with PP problems. For example, Masehian et al. [23] combined the PSO algorithm with PRM to propose a novel multi-objective PP framework that optimizes path length and path smoothness objectives. Davoodi et al. [24] presented an improved NSGA-II to solve path length and path clearance in discrete space. The improved NSGA-II used an innovative family of path refiner operators, in addition to standard genetic operators. Duan et al. [25] proposed a multi-objective PP optimization method was mainly implemented by an improved firefly algorithm that optimized path length and path smoothness objectives. Pareto-dominance was employed to balance the performance of the optimized solutions. In addition, an elite record library was created to reserve the non-dominance solutions. Guo et al. [26] investigated a multi-objective PP problem based on a hybrid algorithm. They weighted path length and path safety objectives using the entropy method to obtain a short and safe path. Gong et al. [27] proposed an improved global PP method using a multi-objective particle swarm optimization algorithm that considered path length and path danger level. In this algorithm, a self-adaptive mutation operation was mainly designed to improve the feasibility of a new path. Similarly, Zhang et al. [28] investigated the PP problem based on the particle swarm optimization algorithm in an uncertain environment. They selected path risk level and path length the two objectives to be optimized and mostly introduced a membership function to evaluate the riskiness of the path and several new operators to improve the algorithm performance. Mac et al. [29] introduced a hierarchical global PP method that designs a three-level structure, which optimizes the path length and the path smoothness, consequently realizing an optimal path. However, most of these studies used schemes that combined multiple objectives into a weighted function, despite claiming that they solved the multi-objective PP problem. These methods produced different optimal solutions according to the weighted value distribution, which are difficult to precisely predict in real problems. In other words, the optimal solution obtained by the weighted multi-objective optimization algorithm may be a better one in a particular situation, but a worse one in another case.

Many works dealt with the PP problem in a multi-objective simultaneous optimization manner, producing multiple non-dominated solutions in a single run instead of balancing the weight coefficients of each objective. For instance, Hidalgo-Paniagua et al. [30] proposed the multi-objective firefly algorithm to solve the multi-objective PP problem, in which path length, path safety, and path smoothness related to the robot's energy consumption are simultaneously optimized. Pareto dominance was employed to evaluate the solution fitness. Similarly, Hidalgo-Paniagua et al. [31] also investigated another algorithm, called the shuffled frog-leaping algorithm, to deal with the multi-objective PP problem. Although state-of-the-art algorithms have achieved considerable achievements, a gap remains in the multi-objective PP problem. In our opinion, path length and path smoothness are linearly correlated in the PP problem. They are usually regarded as two objectives that are not completely in conflict. In our experimental verification, we found that the path with a shorter path length has better smoothness. On the contrary, path length and path safety are two admitted conflicting objectives. In addition, existing evolutionary strategies used by the previous algorithms focused on the macro update of the path segment, and lacked an effective local search strategy. What's more, most of them lacked an effective elimination and regeneration mechanism for stagnated solutions over a period of time.

Proposed by Karaboga in 2005, the artificial bee colony (ABC) algorithm is a swarm intelligence optimization algorithm that aims to solve multivariate function optimization problems [32]. The ABC algorithm imitates the bees' behavior of finding the best honey source and has strong global optimization-seeking ability, fast convergence speed, and fewer parameters. It has been applied to many applications, including the flow shop scheduling problem [33–36] and neural network parameter optimization [37–39], achieving excellent results. It has also been applied to solve the mobile robot PP problem. Based on the ABC algorithm, Safari and Mahjoob [40] proposed a novel method used to optimize the path length objective in a real-time PP problem. They focused on designing a rule for building an initial collision-free path and utilized a swarm algorithm for path optimization. Bhattacharjee et al. [41] presented an improved ABC algorithm to minimize the path length of multiple robots. Using the ABC algorithm, a local trajectory planning scheme was developed to optimally obtain the next position of all robots in the world map from their current positions, such that the local paths of n robots become sufficiently small. The ultimate goal was to minimize the path length of all robots. However, few scholars have used the ABC algorithm to deal with the multi-objective PP problem.

Considering the characteristic of the ABC algorithm and the problems mentioned above, we proposed herein an improved ABC algorithm to solve the multi-objective PP problem of the home service robot, which uses the Pareto dominance to evaluate the deeply evolved solution sets. The decision-maker could select an appropriate scheme from these candidate solutions according to the current specific application requirements. In this study, path length and path safety are considered as two optimized objectives. The contributions of this study are as follows:

- 1) A new hybrid initialization strategy is proposed. This strategy uses the RRT algorithm and the space segmentation method to jointly generate a high-quality initial population. The RRT explores the configuration space through a tree structure from the starting point and incrementally extends the tree of collision-free paths, which generates initial solutions randomly, so as to enhance the distribution of the population. The space segmentation method utilizes the line between the starting and the target points as the axis to conduct space segmentation, which ensures the feasibility of the solutions, meanwhile, this method is conducive to finding these initial solutions with better objective characteristics.
- 2) A variable neighborhood local search strategy is developed. In each iteration, the proposed variable

neighborhood local search strategy changes searching neighborhoods of path nodes according to the closest distance between the current node and the obstacle. This operation can effectively ensure the reachability of the line segments on both sides of the node, so as to avoid lots of time-consuming collision detection.

3) A new global search strategy is presented. During the iteration, solutions with long-time evolutionary stagnation are removed from the population. For new individuals, this global search strategy mainly completes two aspects of work. Firstly, the individuals in the Pareto non-dominated solution set are used to carry out crossover operation to form new ones, which inherits the excellent characteristics of the current solutions. Secondly, new solutions are also randomly generated to ensure the global search ability of the algorithm. Compared with the original ABC algorithm, this strategy significantly accelerates the convergence speed of the algorithm.

The rest of this paper is organized as follows. Section 2 introduces the kinetic model of a home service robot, environment modeling, path encoding, and mathematical model of the optimization objective required for PP. Section 3 presents the algorithmic framework of the IMO-ABC algorithm and the detailed evolutionary strategy. Section 4 describes the simulation environments, performance metrics, effectiveness analysis of the algorithm, and comparative analysis with NSGA-II and MO-PSO. Finally, Section 5 presents the concluding remarks.

2. Path planning

This section presents the kinetic model of a home service robot, environment modeling, path encoding, and the two objectives for optimization.

2.1. Kinetic model of a home service robot

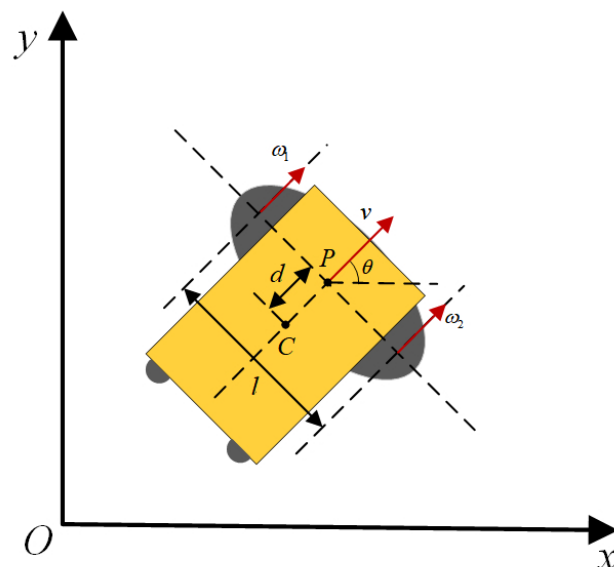


Figure 1. Kinetic model of a home service robot.

Figure 1 shows the kinetic model of a home service robot [42]. The two drive wheels are distributed in the front. The two universal wheels are at the back. Taking P point as the motion reference point and the linear and angular velocities of the robot as v and w , we obtain:

$$\begin{pmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (2.1)$$

Supposing the centroid of the robot C is similar to the geometric center of the box body, the distance from the particle to point P is d . The centroid coordinates are as follows.

$$\begin{cases} x_C = x_P - d \cos \theta \\ y_C = y_P - d \sin \theta \end{cases} \quad (2.2)$$

Taking the derivative of Eq (2.2) and combining Eq (2.1), the equation of the centroid motion of the robot is obtained as

$$\begin{pmatrix} \dot{x}_C \\ \dot{y}_C \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & d \sin \theta \\ \sin \theta & -d \cos \theta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (2.3)$$

The robot is driven by two driving wheels. The direct control is that the speeds of the two driving motor wheels are w_1 and w_2 . Supposing the distance between the two driving wheels is l , the radius is represented as R , and the following formula is obtained.

$$\begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(w_1 R + w_2 R) \\ \frac{1}{2}(w_1 R - w_2 R) \end{pmatrix} \quad (2.4)$$

2.2. Environmental model

Building an environment model that is easy to understand and compute is important in the PP process of mobile robots. In this work, a two-dimensional (2D) map was rasterized with grids containing all the map information. Each grid has a fixed value that indicates the current environmental conditions of the grid. We used occupancy to indicate the degree of the presence of grid obstacles. The occupancy values of the current grid are represented by 0 and 1 if no obstacle is present in the grid and if a grid is fully occupied by obstacles, respectively. However, if an obstacle partially occupies a grid, then its occupancy in the current grid is represented by a value between 0 and 1. The mobile robot occupies a certain physical space; hence, we appropriately extended the obstacles in the map to more reasonably calculate the path safety through the grid. Figure 2(a) shows the rasterized map, where the black grids represent the obstacles. Figure 2(b) presents the map after two obstacle layers had been extended, with the lighter color indicating reduced danger. The obstacles were expanded by three layers, each with 0.7, 0.4, and 0.1 values.

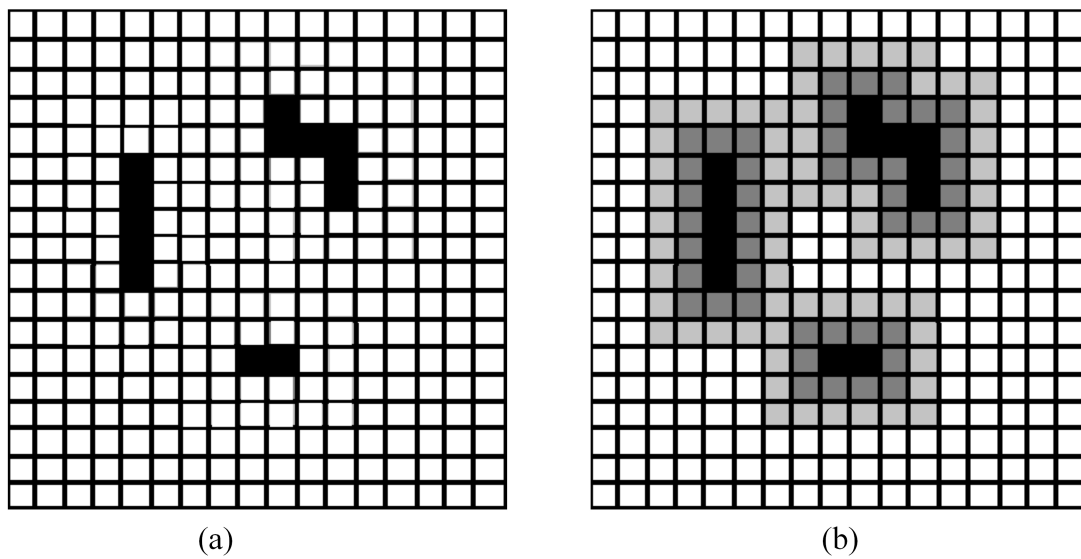


Figure 2. Environment model.

2.3. Path coding

The path coding is a path representation. This section represents the path by several 2D points and sequential line segments connecting the points. The first point represents the start point of the path, while the last point depicts its target point. The number of other points between is a variable. Two consecutive points corresponded to a line segment of the path. Table 1 is a tabular description of the path coding.

Table 1. Path encoding.

Points	Start point	...	Middle point	...	Target point
Coordinate	$P(x_s, y_s)$...	$P(x_i, y_i)$...	$P(x_t, y_t)$

2.4. Path length objective

Choosing the shortest path is the best option when a robot moves from a starting point to a target point. The path length objective is to obtain the shortest possible path. Expressed mathematically, it is the sum of the lengths of each line segment of a path. Therefore, the line segment formed by any two points is calculated based on the Euclidean distance, where $P = [P_0, P_1, \dots, P_i, P_{i+1}, \dots, P_n, P_{n+1}]$ represents the path P . Let $S = P_0$ represent the start point and $T = P_{n+1}$ be the target point. Equations (2.5) and (2.6) calculate path length objective.

$$d(P_i, P_{i+1}) = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad (2.5)$$

$$Length(P) = \sum_{i=0}^n d(P_i, P_{i+1}) \quad (2.6)$$

where $P_i = (x_i, y_i)$ and $P_{i+1} = (x_{i+1}, y_{i+1})$ are the two consecutive points in the path, $d(P_i, P_{i+1})$ is the segment distance in the path; $Length(P)$ denotes the total path length obtained by summing up all line segments; and n denotes the number of points in the path.

2.5. Path safety objective

Path safety objective means that the robot should avoid touching or moving away from the obstacles in the environment as it moves through space. It is calculated by adding the occupancy (Subsection 2.2) of the stepped cells of the grid by the robot along the path. In this work, the robot was considered to have the same size as the grid representing the environment. Equation (2.7) calculates the path safety objective as follows:

$$Safety(P) = \sum_{i=1}^{n-1} SCells_i \quad (2.7)$$

where $SCells_i$ is the sum of the grid occupancy that the robot travels through the i th segment of the path, and n represents the number of points in the path.

3. IMO-ABC algorithm

This section presents the proposed IMO-ABC algorithm used to solve the multi-objective PP problem. First, the framework of the IMO-ABC algorithm is described. Then, the hybrid initialization strategy and path optimization operators are presented. Subsequently, the variable neighborhood local search strategy and the global search strategy are given, respectively.

3.1. The framework of IMO-ABC algorithm

Algorithm 1 presents the framework of IMO-ABC algorithm. Compared with the original ABC algorithm, the detailed improvements are described as follows. In the population initialization stage, the IMO-ABC algorithm employs a new hybrid initialization strategy (cf. Subsection 3.2) instead of random initialization to generate the initialization population. To enhance the quality of the initial solutions, the path-shortening operator (cf. Subsection 3.3) is implemented to remove excess polylines from the feasible path. In the employed bee stage, individuals in the population are implemented by the path-crossing operator (cf. Subsection 3.4) and the variable neighborhood local search strategy (cf. Subsection 3.5) sequentially which aims to generate new individuals. Individuals with better objective values are retained based on Pareto dominance and then are thrown to the next stage. It can be noted that the variable neighborhood local search strategy proposed in this study changes the searching neighborhoods of path nodes according to the surrounding environment, which can effectively enhance the local search ability of the algorithm. In the onlooker bee stage, excellent individuals are implemented by the variable neighborhood local search once again to improve the convergence speed of the algorithm. In the scout bee stage, the original ABC algorithm directly removes individuals with long-time evolutionary stagnation, and then generate random solutions to maintain the scale of the population. Similar but different implementation in this study is that global search strategy (cf. Subsection 3.6) provides an effective mechanism for generating new individuals. For the new individuals, one group of individuals is generated by inheriting the excellent individuals in the current Pareto set, and another

group of individuals is generated randomly. Compared with the original ABC algorithm, this mechanism can not only enhance the global search ability of the algorithm, but also significantly accelerate the convergence speed of the algorithm.

Algorithm 1 IMO-ABC

```

1: Generate an initial population of size  $P_s$  by the hybrid initialization strategy (cf. Subsection 3.2)
2: Perform the path-shortening operator for each initial solution to delete redundant points (cf. Subsection 3.3)
3: Employed bee stage
4: for each solution  $i$  do
5:   Perform the path-crossing operator (cf. Subsection 3.4)
6:   Generate a neighborhood solution  $j$  of  $i$  by using the variable neighborhood local search strategy (cf. Subsection 3.5)
7:   if  $j$  dominates  $i$  then
8:     Replace the current solution with  $j$ 
9:   else
10:    Update the number of non-updating iterations about  $i$ 
11:   end if
12: end for
13: Onlooker bee stage
14: for each solution  $i$  do
15:   Randomly select a non-repeating solution  $k$  from population
16:   Select the better solution between  $i$  and  $k$  as the current solution  $i$ 
17:   Perform steps 6–12 lists in the employed bee stage
18: end for
19: Scout bee stage
20: for each solution  $i$  do
21:   if the number of non-updating iterations of  $i$  reaches  $L_m$  then
22:     Perform the global search strategy (cf. Subsection 3.6)
23:   end if
24: end for

```

3.2. Hybrid initialization strategy

In general, initial individuals with good characteristics are beneficial to accelerate the convergence speed of the algorithm. In this study, we present a hybrid initialization strategy to generate the initial population, which implements based on the RRT algorithm and the space segmentation method of Reference [28]. Feasible paths generated by the RRT algorithm have a wide range of randomness. The RRT algorithm explores the configuration space through a tree structure from the start point and incrementally extends the nodes to find a collision-free path until the branches of the tree close to the target point. The steps of the RRT algorithm are described as follows. First, generate a random point in space, traverse the tree nodes from the tree containing the start point, and find the nearest tree node to the random point. Next, a new node is formed by extending a fixed step from the nearest tree

node toward the random point; a random point is regenerated if an obstacle exists between the new and nearest nodes, otherwise, a new node is added to the tree. Finally, the abovementioned expansion process is repeated until the distance between the new generated node and the target point is less than the step length. A path connecting the start point to the target point is found.

Figure 3 shows the space segmentation method used to generate initial paths. The black entities are obstacles in the configuration space. S and T represent the start and target points of the robot, respectively. Connect S and T and divide a number of equal-length vertical lines l_1, l_2, \dots, l_n . Randomly choose points P_1, P_2, \dots, P_n on each vertical line and sequentially connect these points to form a new path. This method has a certain directionality from the start to the target point. Hence, it is advantageous for finding these initial solutions with better objectives.

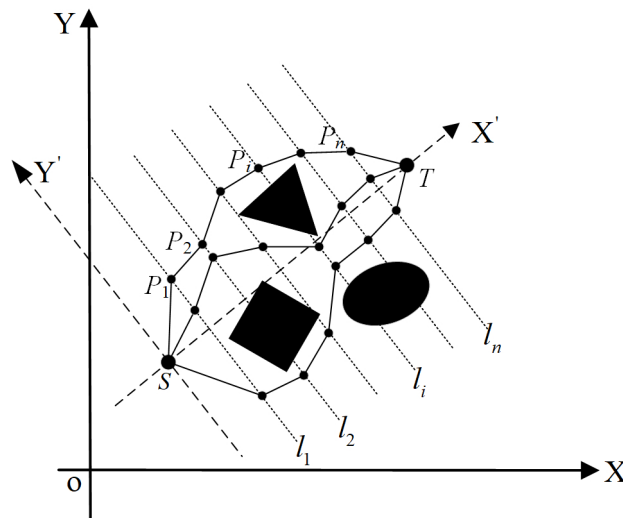


Figure 3. The space segmentation method used to generate initial paths.

The hybrid initialization strategy is implemented as follows:

Step 1: the RRT algorithm and the space segmentation method are used to generate two solutions.

Step 2: the fitness of the initial solutions generated by these two algorithms is evaluated. The solutions are sorted according to the fitness and deposited in the set $Hc = \{h_1, h_2\}$ where $h_i = \{1, 2\}$ denotes the i th initialization method. The fitness of each solution is deposited in $Fc = \{f_1, f_2\}$. Equation (3.1) shows the fitness calculation method:

$$f = \sqrt{(O_{CL} - O_{IL})^2 + (O_{CS} - O_{IS})^2} \quad (3.1)$$

where O_{CL} and O_{CS} represent path length and safety objective values, respectively, after the normalization of the two solutions generated in Step 1, and O_{IL} and O_{IS} are those, respectively, after the normalization of the ideal reference point. The ideal reference point is optimal in each dimension. The ideal value of path length should be the Euclidean distance from the start point to the target point. Path safety is expressed herein by the degree of obstacle possession in the grid. Therefore, the absence of obstacles in the grid indicates a safer path. With this, the ideal safest path value is 0.

Step 3: define a variable AT_i that indicates the number of times each initialization method h_i is used in population initialization. Equation (3.2) is the AT_i calculation method, where P_s is the number of

initial solutions desired to be generated.

$$AT_i = P_s \times \frac{f_i}{\sum_{j=1}^2 f_j} \quad (3.2)$$

Step 4: AT_i solutions are generated with h_i until the population size reaches P_s .

3.3. Path-shortening operator

The number of points of each solution in the population is variable; thus, the purpose of executing the path-shortening operator is to reduce redundant points and facilitate solution evolution. The shortening operator is implemented as follows. Sequentially traverse each path in the population and randomly select two nonadjacent points from the path. These two points are directly connected if they satisfy collision detection. The intermediate points and the line segments between these two points are deleted. The new path is saved. If collision detection is not satisfied, the path is saved, and the next path is continuously traversed until the end of the traversal. Figure 4 illustrates the process of the path-shortening operator.

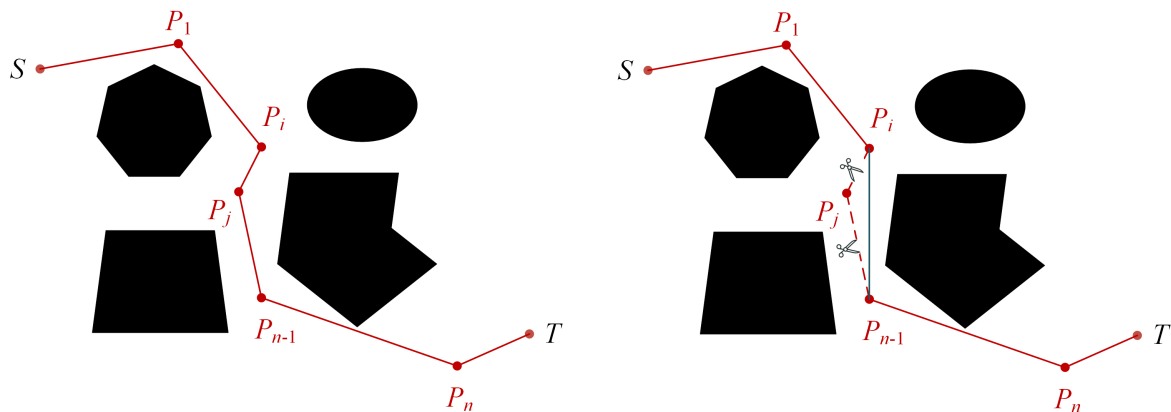


Figure 4. Path-shortening operator.

3.4. Path-crossing operator

Select the two closest points, except for the starting and target points, in the two paths to implement the path-crossing operator and reduce the collision detection risk. The path-crossing operator procedure is performed as follows. Two adjacent paths are sequentially selected from the population. A point is randomly selected from the first path. Then, points on the second path are sequentially traversed to find the closest point to the point selected by the first path for a crossover to form new solutions. If collision detection is not satisfied between the two points, the next closest point from the second path is selected until it is satisfied. If all points on the second path do not satisfy collision detection with the point selected on the first path, another point is randomly selected from the first path, and the abovementioned operations continue. Figure 5 illustrates the process of the path-crossing operator.

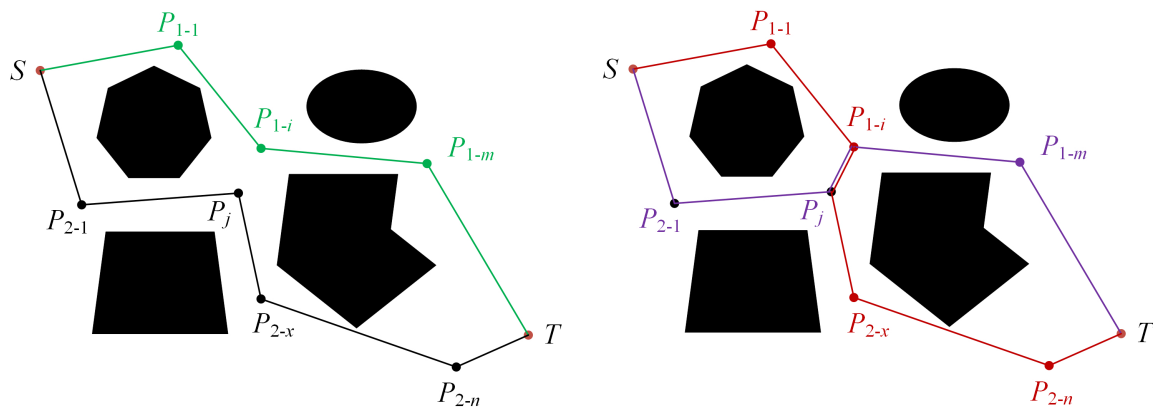


Figure 5. Path-crossing operator.

3.5. Variable neighborhood local search strategy

The employed bee and onlooker bee stages used a new variable neighborhood local search strategy for the path points to enhance the algorithm's exploitation capability. The variable neighborhood local search strategy is implemented as follows:

Step 1: determine the proportion of points on the path that participating in the variable neighborhood local search. In this study, this proportion K_L is a variable, which is calculated based on the number of iterations and numerical interval. Equation (3.3) calculates K_L as follows:

$$K_L = K_{\min} + (K_{\max} - K_{\min}) * \frac{I_c}{I_{\max}} \quad (3.3)$$

where K_{\min} and K_{\max} are the upper and lower bounds of the numerical interval that directly affects the variable neighborhood local search probability K_L . At least K_{\min} of the points in a path are selected for local search. At most, K_{\max} are selected. I_c is the current number of iterations. I_{\max} is the maximum number of iterations.

Step 2: the path points that require local search are selected with probability K_L . The nearest distance d_x to the obstacle is calculated for each point. The d_x is used as the radius of the neighborhood of this point. New points are generated in this neighbor range.

Step 3: if the new point satisfies collision detection with the adjacent points of the original point, it is connected with the adjacent points of the original point to form a new path; otherwise, it will be regenerated in the neighbor range.

Step 4: each new solution performs the path-shortening operator with a certain probability. Note that the probability size has no effect on the population evolution. The number of points in each path is not fixed; thus, redundant points must be deleted.

Figure 6 shows the evolution process by using the variable neighborhood local search strategy.

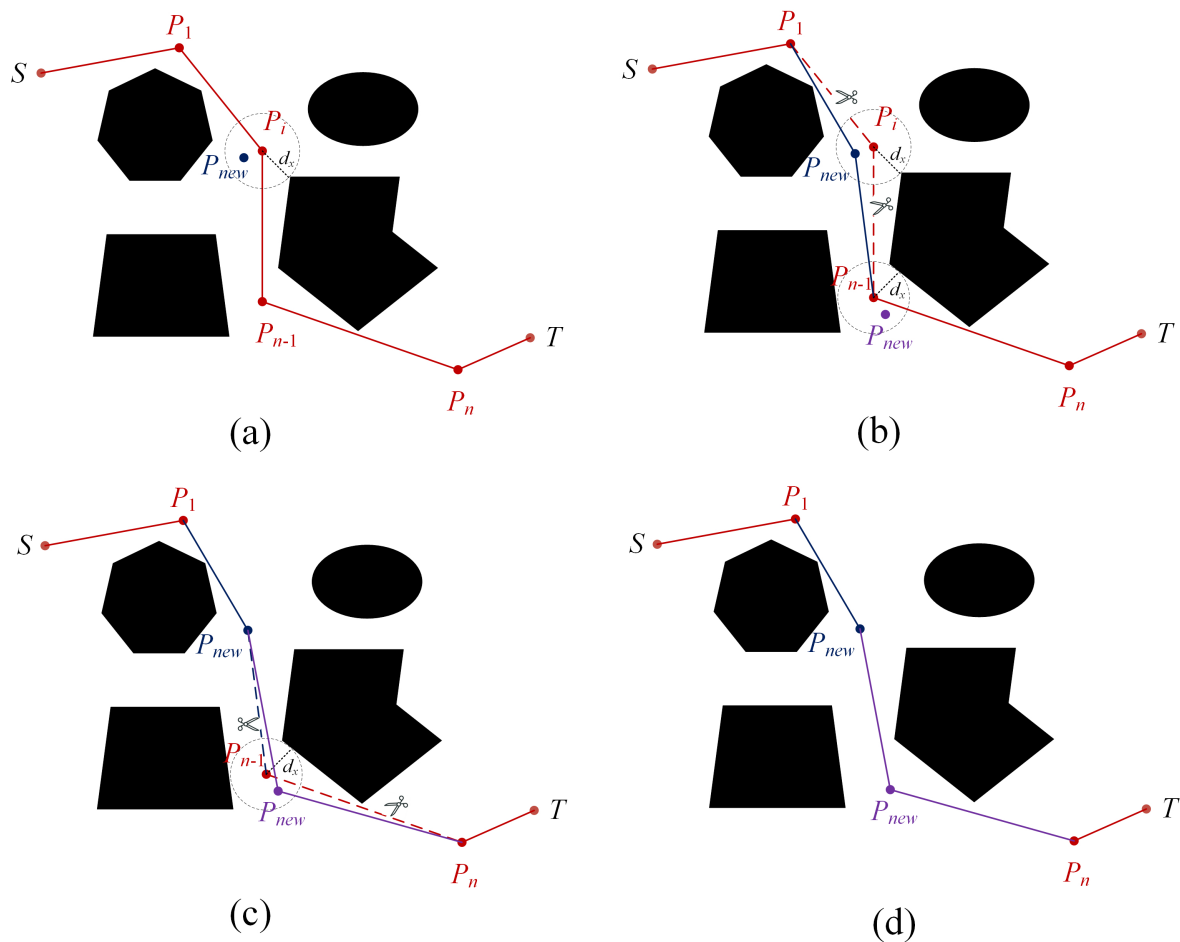


Figure 6. Variable neighborhood local search strategy.

3.6. Global search strategy

In the traditional ABC algorithm, the scout bee will be allowed to randomly generate the solution if a certain individual that has not improved after reaching L_m exists. However, useful information about the current solution will be lost. We developed a new global search strategy embedded in the scout bee stage, which can effectively utilize the information collected in the previous iterations. The strategy is presented in detail in the following steps. All solutions that are not updated after the L_m iterations are eliminated. Accordingly, there is 50% probability of generating a new solution using the hybrid initialization strategy and 50% probability of generating a new solution using Eq (3.4).

$$\eta_{new}^{I_c} = p_{l_{max}}^{I_c} \otimes p_{s_{max}}^{I_c} \quad (3.4)$$

where operator \otimes is the path-crossing operator in Subsection 3.4; $p_{l_{max}}^{I_c}$ is the solution with the largest path length objective value in the Pareto solution set at the current iteration of I_c ; $p_{s_{max}}^{I_c}$ is the solution with the largest path safety objective value in the Pareto solution set at the current iteration of I_c ; and $\eta_{new}^{I_c}$ is the new solution. A better solution is selected based on the Pareto dominance relationship.

4. Simulation results and discussion

4.1. Experimental configuration

Four characteristic maps shown in Figure 7 were used for the simulation test. Map1 shows a structured environment with low-density obstacles. Map2 depicts a scenario with trapped obstacles. Map3 scenario is symmetrical and contains many obstacles. Lastly, Map4 gives a real environment map that cites from Reference [31]. Table 2 shows the coordinate distributions of the starting and target points in the tested maps.

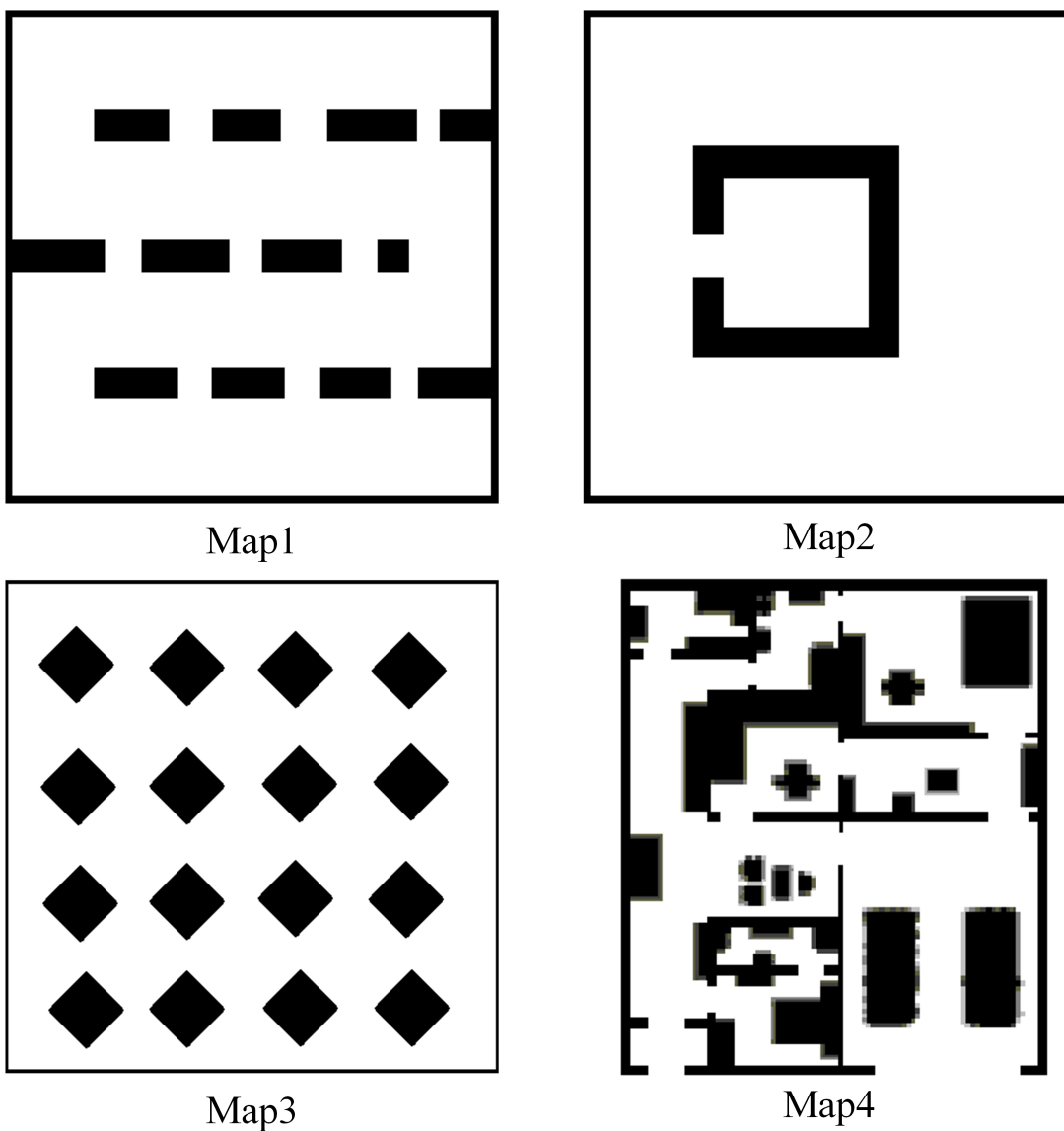


Figure 7. Four maps used for simulation test.

Table 2. Initialization of the tested maps.

Maps	Start point	Target point
Map1	(20, 250)	(480, 250)
Map2	(250, 400)	(250, 250)
Map3	(480, 20)	(20, 480)
Map4	(430,80)	(125,340)

4.2. Performance metrics

The obtained Pareto solution set was analyzed using the performance metrics hypervolume (Hv) and set coverage (SC) to verify the performance of the proposed algorithm. The Hv metric represents the volume covered by the solutions of the Pareto solution set in the objective function space. The concept related to the Hv metric is as follows: in a d -dimensional solution space, $A = \{a_1, a_2, a_3, \dots, a_n\}$ is the set of non-dominated solutions and $r = \{r_1, r_2, r_3, \dots, r_d\}$ denotes the reference points. The hypercube for each element a_i in set A is calculated as $h(a_i) = [a_{i1}, r_1] \times [a_{i2}, r_2] \times \dots \times [a_{id}, r_d]$. The hypervolume of A can be obtained using the union of $|A|$ hypercubes, the repeatedly covered hypercubes are counted once. Equation (4.1) calculates the hypervolume. Equation (4.2) computes the SC metric.

$$Hv(A, r) = Leb\left(\bigcup_{i=1}^{|A|} h(a_i) | a_i \in A\right) \quad (4.1)$$

$$SC(A, B) = \frac{|\{b \in B; \exists a \in A : a < b\}|}{|B|} \quad (4.2)$$

where B is a set of non-dominated solutions $\{b_i\}$ covered by the non-dominated solutions $\{a_i\}$ in A . $SC(A, B) = 0$ means that none of the solutions in B is dominated by the solutions in A . By contrast, $SC(A, B) = 1$ shows that all solutions in A dominate the solutions in B . The dominance relationship between solution sets A and B is asymmetric; hence, $SC(A, B)$ and $SC(B, A)$ are simultaneously computed.

Finally, the algorithm metrics are analyzed by calculating the reference points in each map. An ideal reference point and a reference point at the nadir are chosen. The nadir point represents the worst in each dimension. Table 3 lists the reference points in each map.

Table 3. Test maps reference points (path length, path safety).

Maps	Ideal point	Nadir point
Map1	(460.00, 0)	(1038.43, 42.8)
Map2	(150.00, 0)	(982.49, 126.3)
Map3	(650.54, 0)	(2271.54, 64.3)
Map4	(400.78, 0)	(1308.09, 41.8)

4.3. Experimental parameter analysis

In general, parameters significantly affect the algorithm performance. The experimental parameters include the population size P_s [23–25, 29, 43], maximum number of non-updating iterations L_m [44],

and variable neighborhood local search variables, K_{min} and K_{max} . The population size P_s and the maximum number of non-updating iterations L_m should be appropriately adjusted. Excessively large P_s and L_m could occupy too much computational space and affect the running speed, while excessively small P_s and L_m could result in low-quality solutions beyond our expectations. K_{min} and K_{max} are closely related; thus, they were set as an interval variable $[K_{min}, K_{max}]$. Fewer intermediate points exist in each path at the late stage of the experimental evolution. To more precisely perform a local search for the solution, the existence of one point of each path is basically guaranteed to execute the variable neighborhood local search strategy. Therefore, the $[K_{min}, K_{max}]$ value should be set moderately. We refer to the parameter settings of other multi-objective PP problems and set an appropriate level classification for each parameter. Table 4 shows the suitable parameter levels, the [1.0, 1.0] indicates that 100% of all intermediate points in the path executes the variable neighborhood local search strategy, while [0.1, 0.5] indicates that as the number of iterations increases, a minimum of 10% to a maximum of 50% of the intermediate points in the path are selected to execute the variable neighborhood local search strategy. We used the Taguchi method [45] to find the best parameter combination and achieve optimal algorithm performance. Table 5 lists the orthogonal array $L_{16}(4^3)$ to determine the parameter combination. Each combination is run 30 times independently using the IMO-ABC algorithm to obtain statistical results. Figure 8 shows the factor-level trends of these two parameters. Based on the result analysis, we set the appropriate parameter values as $P_s = 100$, $L_m = 15$, and $[K_{min}, K_{max}] = [0.1, 0.5]$.

Table 4. Parameter levels and tested values.

Parameters	Parameter levels			
	1	2	3	4
P_s	50	100	150	200
L_m	5	10	15	20
$[K_{min}, K_{max}]$	[1.0, 1.0]	[0.1, 0.9]	[0.6, 0.9]	[0.1, 0.5]

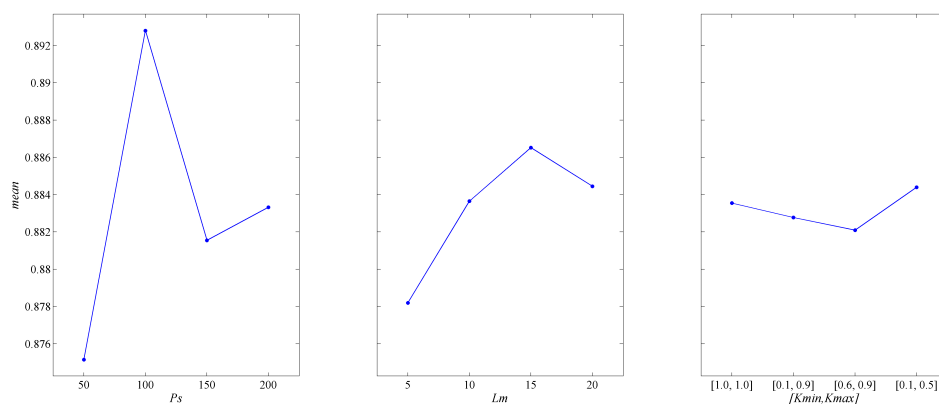


Figure 8. Factor-level trend.

Table 5. Orthogonal array and the average Hv metric values.

Experiment number	Parameters			Average Hv values
	P_s	L_m	$[K_{min}, K_{max}]$	
1	1	1	1	0.8741
2	1	2	2	0.8749
3	1	3	3	0.8752
4	1	4	4	0.8764
5	2	1	2	0.8836
6	2	2	1	0.8940
7	2	3	4	0.9027
8	2	4	3	0.8909
9	3	1	3	0.8787
10	3	2	4	0.8821
11	3	3	1	0.8805
12	3	4	2	0.8849
13	4	1	4	0.8764
14	4	2	3	0.8836
15	4	3	2	0.8877
16	4	4	1	0.8856

4.4. Performance of each part of the proposed algorithm

4.4.1. Effectiveness of the hybrid initialization strategy

We analyze the performance of the hybrid initialization strategy by implementing two different types of IMO-ABC algorithm: one that includes all the components discussed in Section 3 (IMO-ABC algorithm) and another that embeds all components, except for the hybrid initialization strategy (IMO-ABC-NH). We apply the relative percentage increase (RPI) as a performance measure [46]. The results obtained by the two comparison algorithms are collected in 30 independent runs. Equation (4.3) calculates the RPI as follows:

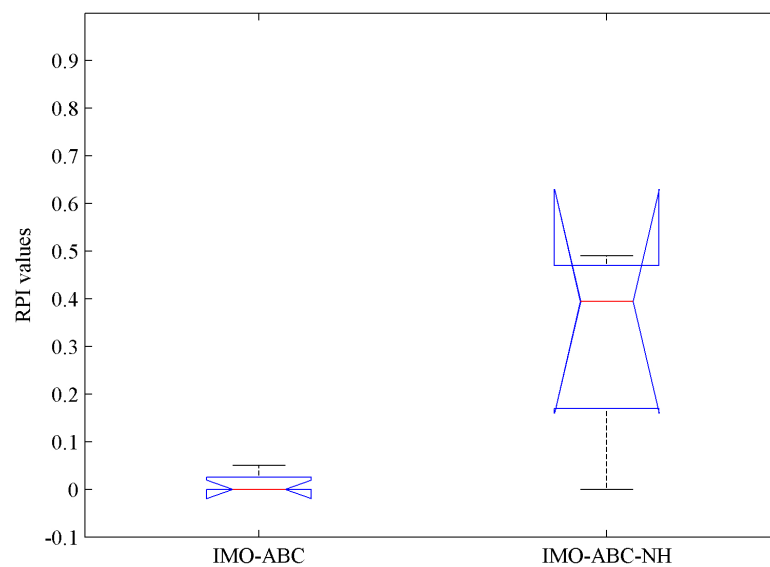
$$\text{RPI}(Hv) = \frac{Hv_b - Hv_c}{Hv_b} \times 100 \quad (4.3)$$

where Hv_b denotes the best Hv value obtained by all comparison algorithms and Hv_c is the best Hv value obtained by the specified algorithm. Table 6 presents the experimental results of both algorithms. The first column shows the different map scenarios. The second column gives the best average Hv values for the two algorithms run independently for 30 times. The last two columns calculate the RPI values of both algorithms.

We performed a multifactor analysis of variance (ANOVA) [47] to evaluate whether or not the differences between the two algorithms were significant. Figure 9 shows the ANOVA results of the two algorithms, showing that the hybrid initialization strategy improved the performance of the IMO-ABC algorithm. The hybrid initialization strategy produced an initial population with a better distribution, while the high-quality solution facilitated a rapid population evolution.

Table 6. Comparison results of the optimal solution sets for IMO-ABC and IMO-ABC-NH.

Maps	Best H_v value	Algorithms		RPI	
		IMO-ABC	IMO-ABC-NH	IMO-ABC	IMO-ABC-NH
Map1	0.9045	0.9045	0.9014	0.00	0.34
Map2	0.9027	0.9027	0.8978	0.00	0.49
Map3	0.9124	0.9119	0.9124	0.05	0.00
Map4	0.9029	0.9029	0.8988	0.00	0.45
Mean	0.9056	0.9055	0.9026	0.01	0.32

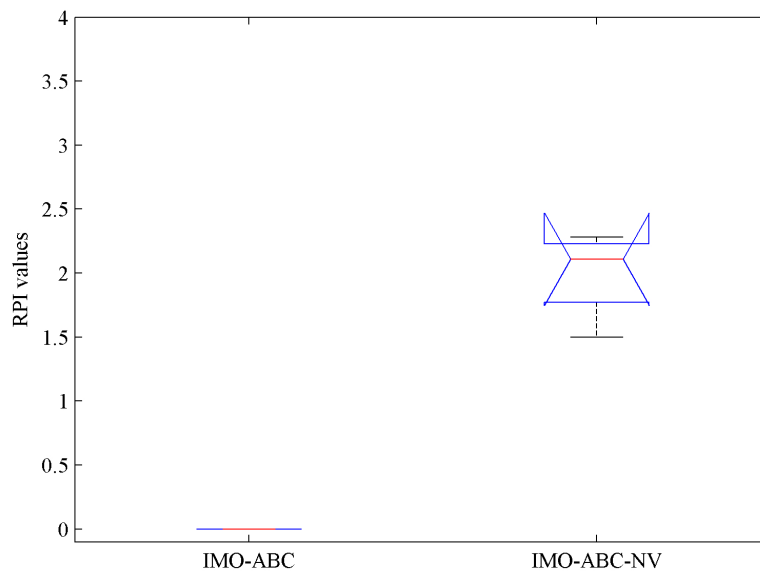
**Figure 9.** ANOVA of the proposed hybrid initialization strategy.

4.4.2. Effectiveness of the variable neighborhood local search strategy

We analyze the performance of the variable neighborhood local search strategy by implementing two different types of IMO-ABC algorithms. We implement two different types of IMO-ABC algorithms: one that includes all the components discussed in Section 3 (referred to as the IMO-ABC algorithm) and one that embeds all the components except the variable neighborhood local search strategy (referred to as IMO-ABC-NV). Table 7 presents the experimental results of both algorithms, and Figure 10 shows their ANOVA results. The proposed variable neighborhood local search strategy can improve the local search capability of the algorithm. The algorithm can adaptively change the search intensity in the evolution using the dynamic probability K_L and the variable search neighborhood, allowing for a detailed exploitation process in the early stages of evolution or an extensive exploration in the later stages.

Table 7. Comparison results of the optimal solution sets for IMO-ABC and IMO-ABC-NV.

Maps	Best H_v value	Algorithms		RPI	
		IMO-ABC	IMO-ABC-NV	IMO-ABC	IMO-ABC-NV
Map1	0.9037	0.9037	0.8901	0.00	1.50
Map2	0.9032	0.9032	0.8848	0.00	2.04
Map3	0.9116	0.9116	0.8917	0.00	2.18
Map4	0.9032	0.9032	0.8826	0.00	2.28
Mean	0.9054	0.9054	0.8873	0.00	2.00

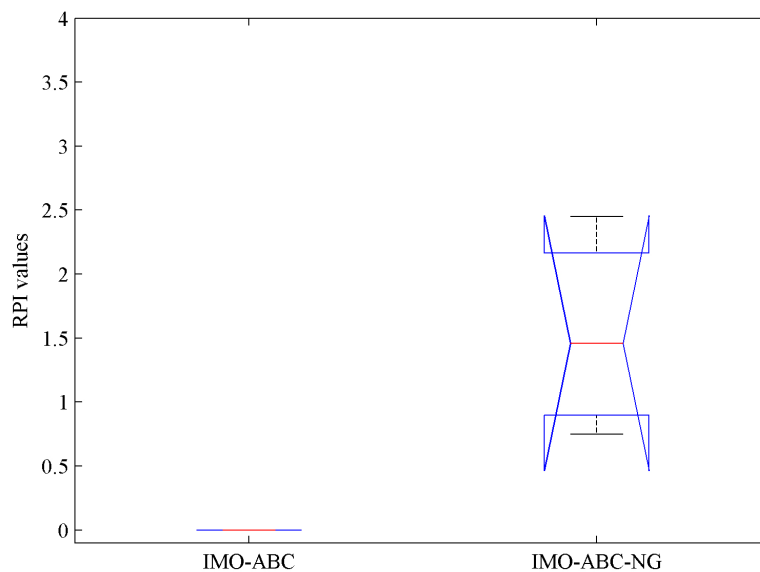
**Figure 10.** ANOVA of the proposed variable neighborhood local search strategy.

4.4.3. Effectiveness of the global search strategy

We analyze the performance of the new global search strategy by implementing two different types of IMO-ABC algorithms: the IMO-ABC and IMO-ABC-NG algorithms. In the IMO-ABC-NG algorithm, the scout bee randomly generates solutions as a global search process. Table 8 presents the experimental results of both algorithms, while Figure 11 shows their ANOVA results. The proposed new global search strategy obtains better results than the random method, utilizing the useful information of solutions in the current Pareto solution set to generate a new solution. In summary, the new global search strategy can improve exploration ability.

Table 8. Comparison results of the optimal solution sets for IMO-ABC and IMO-ABC-NG.

Maps	Best H_v value	Algorithms		RPI	
		IMO-ABC	IMO-ABC-NG	IMO-ABC	IMO-ABC-NG
Map1	0.9035	0.9035	0.8865	0.00	1.88
Map2	0.9028	0.9028	0.8960	0.00	0.75
Map3	0.9097	0.9097	0.8874	0.00	2.45
Map4	0.9041	0.9041	0.8947	0.00	1.04
Mean	0.9050	0.9050	0.8912	0.00	1.53

**Figure 11.** ANOVA of the proposed global search strategy.

4.5. Algorithm comparison results

We validate the performance of the proposed algorithm by comparing it with those of several other well-known algorithms. All results are the average values of 30 independent runs of the three algorithms, with 100 iterations each. Table 9 presents the mean and standard deviation values of the knee point (solution) of NSGA-II [24], MO-PSO [28] and IMO-ABC in different maps. The knee point refers to the solution with the smallest distance normalized to the ideal point in the obtained set of non-dominated solutions. All algorithms are characterized by randomness in PP. The data in front of “ \pm ” denote the average objective value. The data after “ \pm ” show the standard deviation. The data in bold emphases are optimum. The running time is the result of each algorithm after 100 iterations. The IMO-ABC algorithm outperforms the two multi-objective optimization algorithms (i.e., NSGA-II and MO-PSO) in terms of path length, path safety, and running time. The standard deviation values of all objectives in the four maps are the smallest, directly reflecting the better robustness of the proposed

algorithm.

Table 9. Mean and standard deviation values of the knee point of different algorithms.

Maps	Algorithm	Path length	Path safety	Running time(s)
Map1	NSGA-II	487.75 ± 5.78	10.34 ± 3.47	134.14 ± 5.86
	IMO-ABC	484.78 ± 1.45	13.35 ± 1.78	91.72 ± 1.24
	MO-PSO	486.24 ± 3.59	13.29 ± 4.25	116.85 ± 2.49
Map2	NSGA-II	570.23 ± 3.46	28.96 ± 5.37	108.62 ± 3.34
	IMO-ABC	568.34 ± 0.12	26.88 ± 2.53	83.75 ± 2.23
	MO-PSO	572.85 ± 2.16	25.69 ± 7.93	138.71 ± 3.97
Map3	NSGA-II	663.71 ± 1.05	5.09 ± 2.63	93.76 ± 2.46
	IMO-ABC	659.28 ± 0.12	5.23 ± 2.12	78.38 ± 2.25
	MO-PSO	660.85 ± 0.32	5.86 ± 3.25	102.27 ± 4.61
Map4	NSGA-II	513.76 ± 1.07	2.56 ± 1.53	83.54 ± 2.78
	IMO-ABC	510.27 ± 0.13	5.34 ± 0.91	67.15 ± 1.98
	MO-PSO	520.17 ± 2.52	1.89 ± 1.24	97.02 ± 2.53

We also compare the Hv and SC parameter metrics of the three algorithms. Figure 12 shows the Hv values of the three algorithms in the four maps. The average Hv values of the proposed algorithm are better than the other two algorithms, indicating that the Pareto optimal solution set obtained by IMO-ABC covers a greater area of the solution space. The proposed algorithm converges well and reflects its good robustness from the side. In conclusion, the proposed IMO-ABC is always better than NSGA-II and MO-PSO in solving the PP problem. Table 10 shows the average SC results achieved by the compared algorithms, which indicate that the proposed IMO-ABC covers a higher number of NSGA-II and MO-PSO solutions. Similarly, the proposed algorithm is a better choice than NSGA-II and MO-PSO in solving the PP problem.

Figure 13 shows the approximate Pareto solution sets obtained by the IMO-ABC, NSGA-II and MO-PSO algorithms in different maps. The red points depict the approximate Pareto solution sets obtained by the present algorithm. The green and black points are the approximate Pareto solution sets of NSGA-II and MO-PSO, respectively. The red points have better distribution and are closer to the ideal reference point than the other points. The yellow ideal reference points are not exact because the coordinate range is limited. Table 3 presents detailed data. The figures relatively intuitively show that the proposed IMO-ABC algorithm has advantages over the other two algorithms in solving the PP problem. Figure 14 illustrates the knee point (solution) of the proposed algorithm. The knee point is a solution with a minimum distance from the ideal point. Equation (3.1) gives the distance calculation method. The pink square represents the start point. The red pentagram represents the target point.

We conducted a detailed comparison of the IMO-ABC, NSGA-II, and MO-PSO algorithms to demonstrate the convergence performance of the present algorithm. Figure 15 depicts the convergence curves in the Hv metric for the four scenarios. The red line denotes the convergence curve of the proposed algorithm. The blue line is the convergence curve of the NSGA-II algorithm. Lastly, the green line represents the convergence curve of the MO-PSO algorithm. The IMO-ABC algorithm provides the best initial results for the Hv values compared to the other two algorithms, indicating the

effectiveness of the hybrid initialization strategy used in the population initialization stage to generate a high-quality initial population. In addition, it can converge to the optimal value faster, proving that the variable neighborhood local search strategy can find better solutions faster, and that the global search strategy can uncover more valuable solutions and avoid the population from falling into a local optimum. Based on this analysis, the IMO-ABC algorithm is proven to better balance the global search and the local exploitation of the population.

Table 10. Average SC results achieved by different algorithms.

Maps	SC			
	[IMO-ABC, NSGAI]	[NSGAI, IMO-ABC]	[IMO-ABC, MO-PSO]	[MO-PSO, IMO-ABC]
Map1	0.8182	0.0015	0.6012	0.0023
Map2	0.6667	0.0421	0.8571	0.0754
Map3	0.4424	0.0337	0.7500	0.0264
Map4	0.3333	0.0833	0.9064	0.0561
Average	0.5652	0.0402	0.7787	0.0401

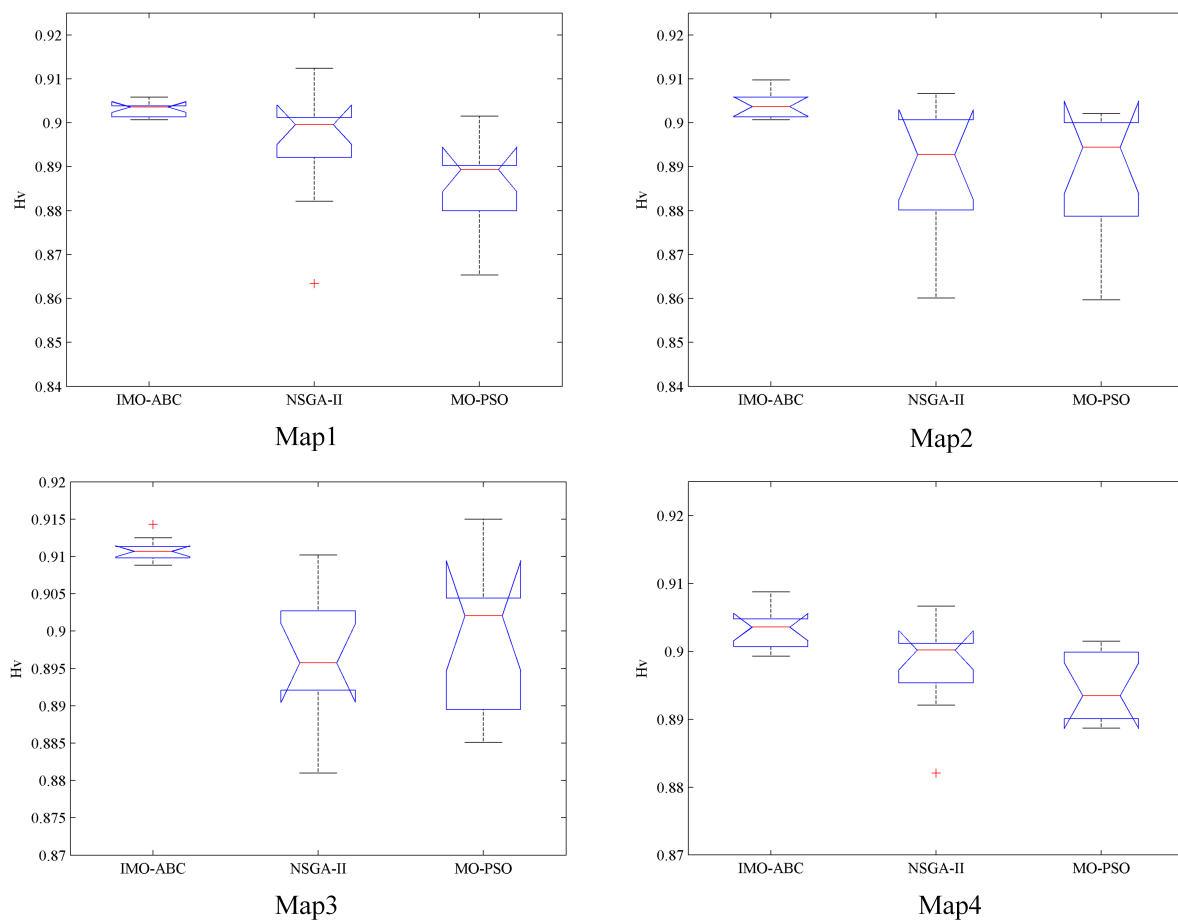


Figure 12. Distribution of Hv values of the three algorithms in four tested maps.

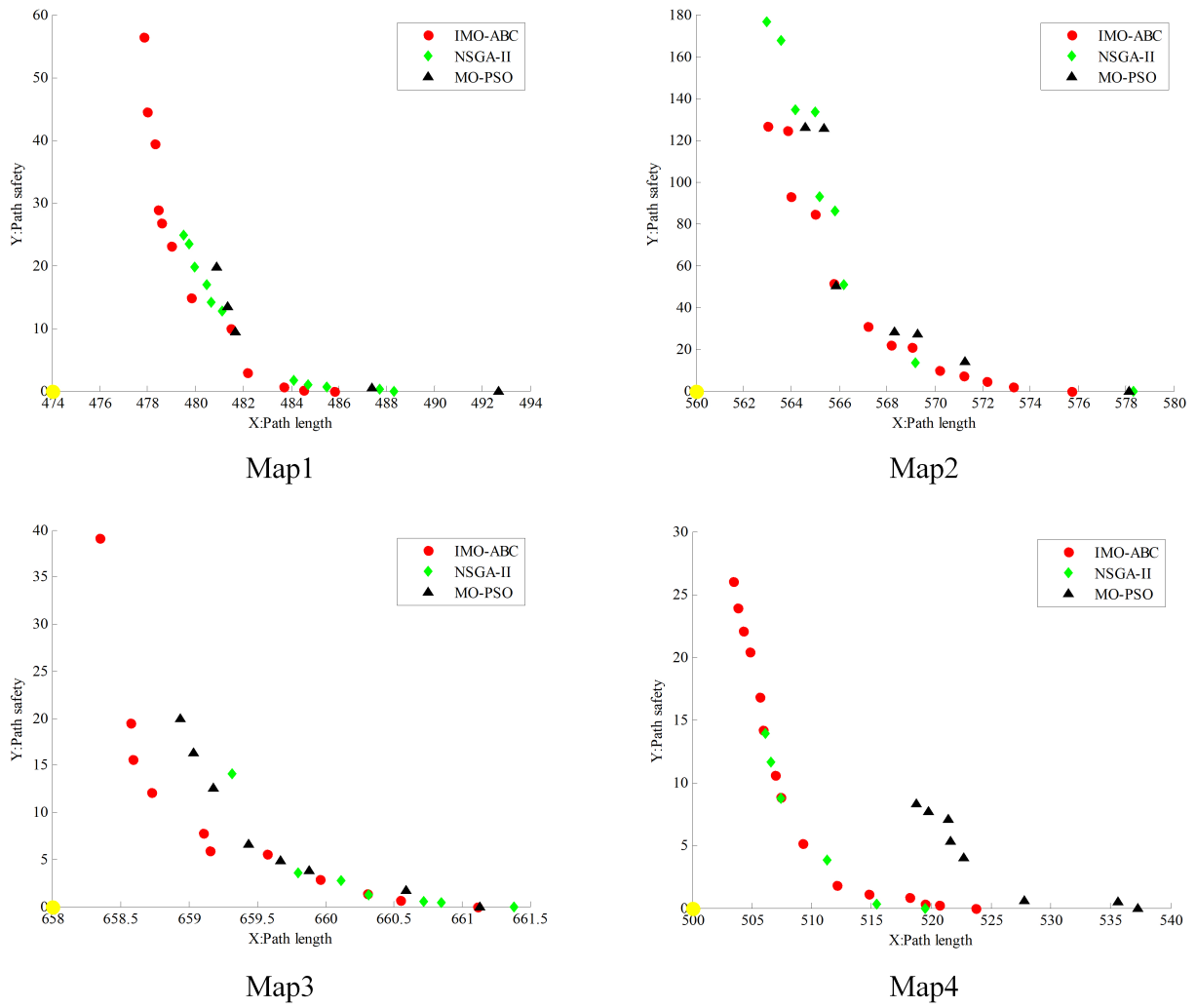
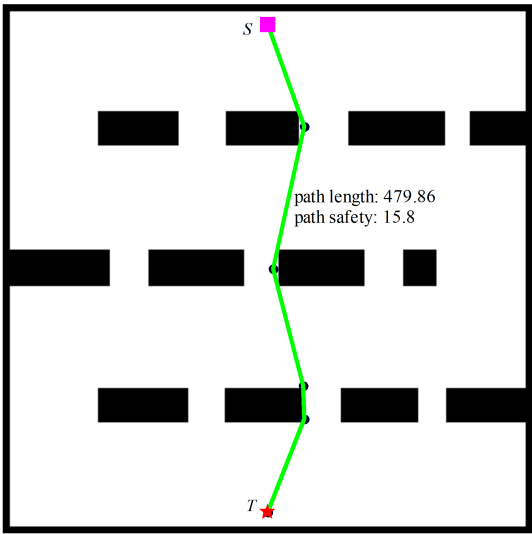
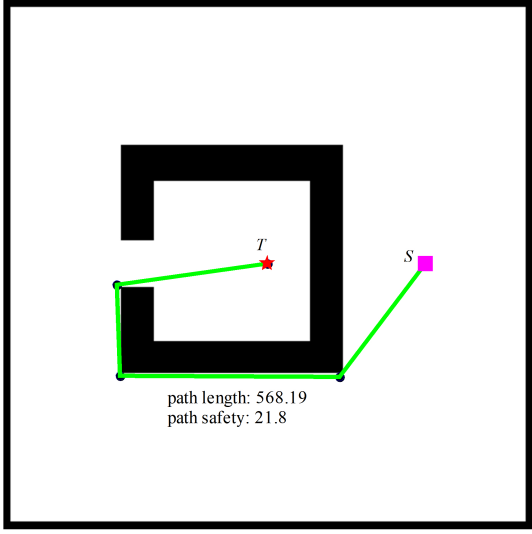


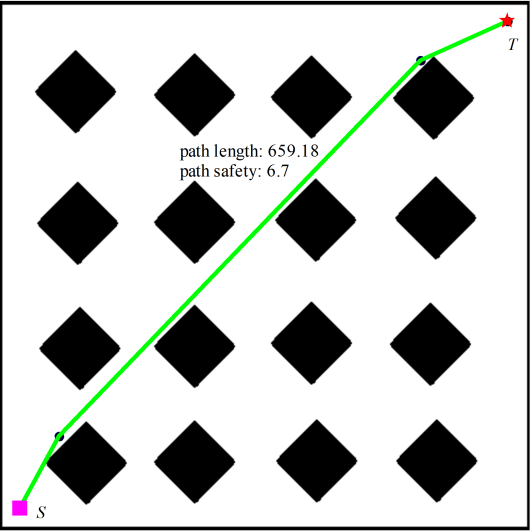
Figure 13. Approximate Pareto fronts obtained from four tested maps.



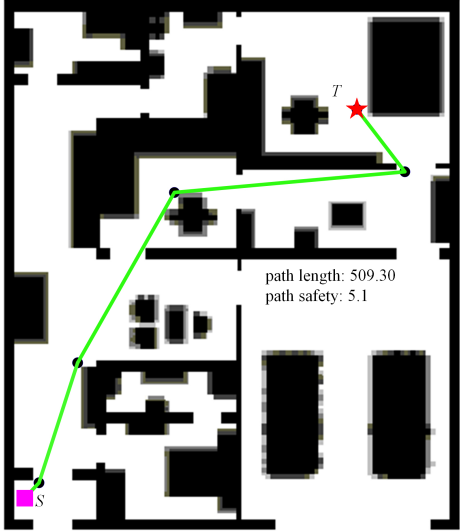
Map1



Map2



Map3



Map4

Figure 14. Graphical solution achieved in four tested maps.

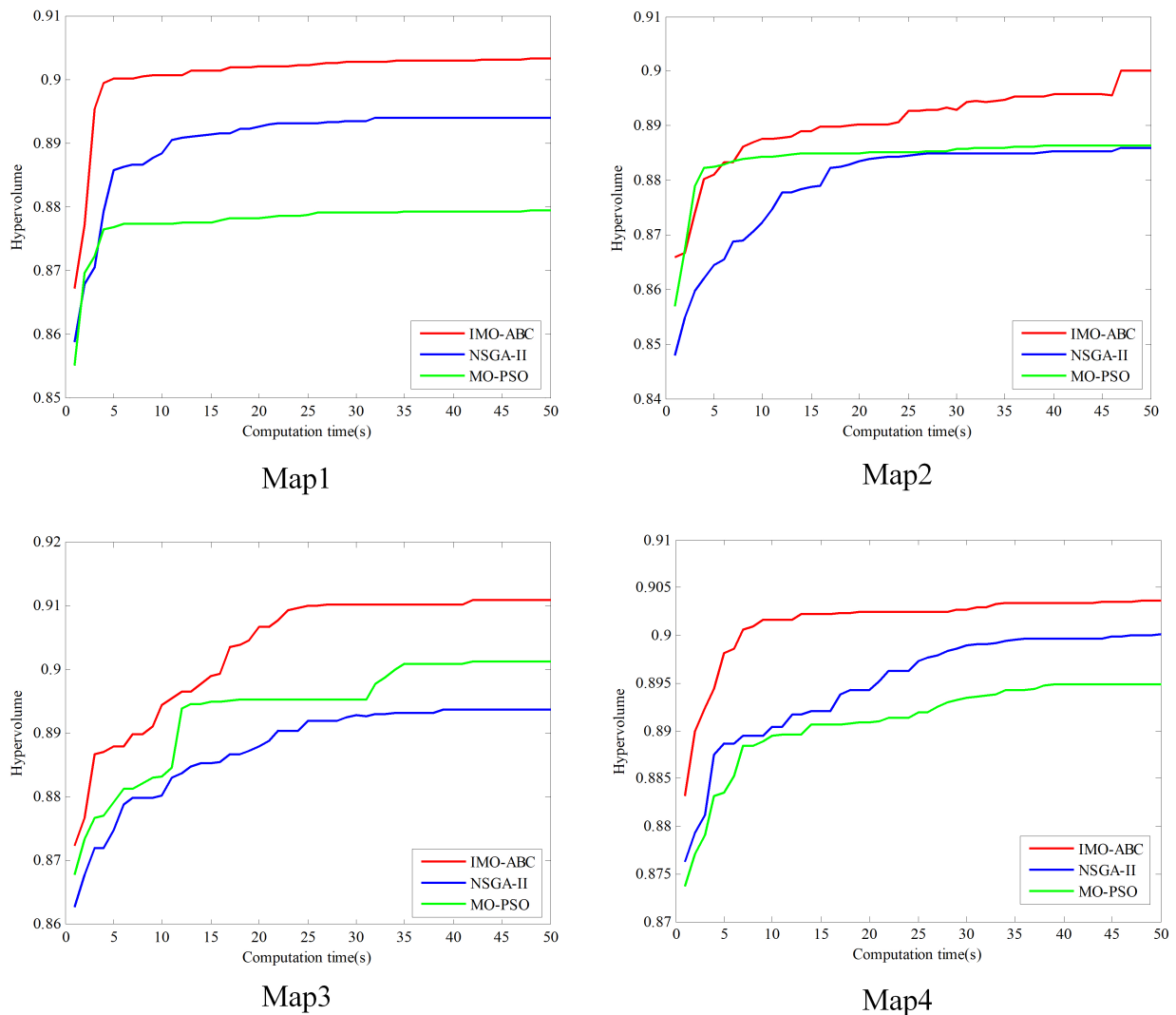


Figure 15. Comparison of convergence curves in Hv metric obtained by the proposed IMO-ABC, NSGA-II and MO-PSO algorithms.

5. Concluding remarks

In this work, we proposed an improved ABC algorithm to solve the multi-objective PP problem of mobile robots. Two objectives, i.e., path length and path safety, are optimized simultaneously. First, a hybrid initialization strategy employs the RRT algorithm and the space segmentation method to generate the initial population. The RRT algorithm was stochastic, and the solutions it produced basically covered the entire initial population, while the space segmentation method facilitated finding the initial solution with better objective characteristics. The proposed hybrid initialization strategy ensures the diversity of the initial population and the quality of individuals. Then, we adopted the variable neighborhood local search and global search strategies to improve the algorithm exploitation and exploration during the population evolution. The variable neighborhood local search strategy changed

searching neighborhoods of path nodes according to the closest distance between the current node and the obstacle in each iteration. The global search strategy utilizes the information of edge solutions in the current Pareto front to generate new solutions to replace the solutions with long-time evolutionary stagnation. Thus, this strategy benefited the population to get rid of the local optimum. In the simulation, we used four representative maps, including a real environment to verify the effectiveness of the proposed algorithm. Taguchi method was employed for parametric analysis to find suitable parameter combinations of the IMO-ABC algorithm. In addition, the effectiveness of each proposed strategy was examined one by one. Compared with the well-known NSGA-II and MO-PSO algorithms, the solution obtained by the proposed algorithm has outstanding advantages in the Hv and SC metrics. Results also indicated that the proposed IMO-ABC algorithm has faster convergence and better robustness than the other two algorithms. Therefore, the proposed algorithm could be an alternative method to solve the multi-objective PP problem for later decision-makers.

This study focused on finding as many feasible paths as possible to provide multiple options for a mobile robot or a decision-maker. The robot or the decision-maker could choose the best path that is more conducive to completing the service task according to the current application scenario. Once a feasible path is selected, the path smoothing module can further smooth the discounted path segments formed by the point set to facilitate path tracking for the service robot. Besides, the static environment was considered in this work, indeed, the PP problem in dynamic environments has been a hot topic in recent years. Therefore, in subsequent studies, the future work mainly focuses on two aspects: 1) design a path smoothing module to produce a path that is easy for the robot to follow; 2) apply the multi-objective optimization methods to solve the considered PP problems under dynamic environments.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (52205529, 61803192), the Natural Science Foundation of Shandong Province (ZR2021QE195, ZR2021MD090, ZR2018BD008, and ZR2016FL13), and the Special Fund Plan for Local Science and Technology Development Lead by Central Authority, and the Guangyue Youth Scholar Innovation Talent Program support received from Liaocheng University (LCUGYTD2022-03).

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. K. Sharma, R. Doriya, Path planning for robots: An elucidating draft, *Int. J. Intell. Robot.*, **4** (2020), 1–14. <https://doi.org/10.1007/s41315-020-00129-0>
2. H. Zhao, J. Liu, H. Chen, J. Chen, Y. Li, J. Xu, et al., Intelligent diagnosis using continuous wavelet transform and gauss convolutional deep belief network, *IEEE T. Reliab.*, (2022), 1–11. <https://doi.org/10.1109/TR.2022.3180273>

3. S. Liu, G. Tian, Y. Zhang, P. Duan, Scene recognition mechanism for service robot adapting various families: A CNN-based approach using multi-type cameras, *IEEE T. Multimedia*, 2021. <https://doi.org/10.1109/TMM.2021.3080076>
4. M. Zhang, G. Tian, Y. Zhang, P. Duan, Service skill improvement for home robots: Autonomous generation of action sequence based on reinforcement learning, *Knowl.-based Syst.*, **212** (2021). <https://doi.org/10.1016/j.knosys.2020.106605>
5. Y. Zhang, G. Tian, X. Shao, S. Liu, M. Zhang, P. Duan, Building metric-topological map to efficient object search for mobile robot, *IEEE Trans. Ind. Electron.*, **69** (2022), 7070–7087. <https://doi.org/10.1109/TIE.2021.3095812>
6. M. Basavanna, M. Shivakumar, An overview of path planning and obstacle avoidance algorithms in mobile robots, *IJERT*, **8** (2019). <https://doi.org/10.17577/IJERTV8IS120252>
7. W. Deng, L. Zhang, X. Zhou, Y. Zhou, Y. Sun, W. Zhu, et al., Multi-strategy particle swarm and ant colony hybrid optimization for airport taxiway planning problem, *Inf. Sci.*, **612** (2022), 576–593. <https://doi.org/10.1016/j.ins.2022.08.115>
8. A. A. Ravankar, A. Ravankar, T. Emaru, Y. Kobayashi, HPPRM: Hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots, *IEEE Access*, **8** (2020), 221743–221766. <https://doi.org/10.1109/ACCESS.2020.3043333>
9. F. Islam, J. Nasir, U. Malik, Y. Ayaz, O. Hasan, RRT-Smart: Rapid convergence implementation of RRT towards optimal solution, in *2012 IEEE International Conference on Mechatronics and Automation*, (2020), 1651–1656. <https://doi.org/10.1109/ICMA.2012.6284384>
10. J. Suh, J. Gong, S. Oh, Fast damping-based cost-aware path planning with non-myopic extensions using cross entropy, *IEEE Trans. Robot.*, **33** (2017), 1313–1326. <https://doi.org/10.1109/TRO.2017.2738664>
11. R. Szczepanski, A. Bereit, T. Tarczewski, Efficient local path planning algorithm using artificial potential field supported by augmented reality, *Energies*, **14** (2021), 1–14. <https://doi.org/10.3390/en14206642>
12. H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu, T. Liu, The EBS-A* algorithm: An improved A* algorithm for path planning, *Plos One*, **17** (2022), 1–27. <https://doi.org/10.1371/journal.pone.0263841>
13. Y. Singh, S. Sharma, R. Sutton, D. Hatton, Towards use of dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field, *TransNav Int. J. Marine Navigat. Safety Sea Transp.*, **12** (2018), 125–131. <https://doi.org/10.12716/1001.12.01.14>
14. M. T. Jensen, Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms, *IEEE Trans. Evolut. Comput.*, **7** (2003), 503–515. <https://doi.org/10.1109/TEVC.2003.817234>
15. C. Lamini, S. Benhlima, A. Elbekri, Genetic algorithm based approach for autonomous mobile robot path planning, *Procedia Comput. Sci.*, **127** (2018), 180–189. <https://doi.org/10.1016/j.procs.2018.01.113>

16. K. Hao, J. Zhao, B. Wang, Y. Liu, C. Wang, The application of an adaptive genetic algorithm based on collision detection in path planning of mobile robots, *Comput. Intell. Neurosci.*, **2021** (2021), 1–20. <https://doi.org/10.1155/2021/5536574>
17. Y. Liu, X. Zhang, X. Guan, D. Delahaye, Potential odor intensity grid based UAV path planning algorithm with particle swarm optimization approach, *Math. Probl. Eng.*, **2016** (2016), 1–20. <https://doi.org/10.1155/2016/7802798>
18. J. Li, X. Chen, P. Duan, J. Mou, KMOEA: A knowledge-based multi-objective algorithm for distributed hybrid flow shop in a prefabricated system, *IEEE Trans. Ind. Inf.*, **18** (2022), 5318–5329.
19. Z. Xie, Y. Jia, C. Zhang, X. Shao, D. Li, Blocking flow shop scheduling problem based on migrating birds optimization, *Comput. Int. Manu. Syst.*, **21** (2015), 2099–2107. <https://doi.org/10.13196/j.cims.2015.08.015>
20. W. Deng, J. Xu, X. Gao, H. Zhao, An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems, in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **52** (2022), 1578–1587. <https://doi.org/10.1109/TSMC.2020.3030792>
21. Y. Song, X. Cai, X. Zhou, B. Zhang, H. Chen, Y. Li, et al., Dynamic hybrid mechanism-based differential evolution algorithm and its application, *Expert Syst. Appl.*, **213** (2022). <https://doi.org/10.1016/j.eswa.2022.118834>
22. W. Deng, H. Ni, Y. Liu, H. Chen, H. Zhao, An adaptive differential evolution algorithm based on belief space and generalized opposition-based learning for resource allocation, *Appl. Soft. Comput.*, **127** (2022). <https://doi.org/10.1016/j.asoc.2022.109419>
23. E. Masehian, D. Sedighzadeh, A multi-objective PSO-based algorithm for robot path planning, in *2010 IEEE International Conference on Industrial Technology*, (2010), 465–470. <https://doi.org/10.1109/ICIT.2010.5472755>
24. M. Davoodi, F. Panahi, A. Mohades, S. N. Hashemi, Multi-objective path planning in discrete space, *Appl. Soft Comput.*, **13** (2013), 709–720. <https://doi.org/10.1016/j.asoc.2012.07.023>
25. P. Duan, J. Li, H. Sang, Y. Han, Q. Sun, A developed firefly algorithm for multi-objective path planning optimization problem, in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, (2018), 1393–1397. <https://doi.org/10.1109/CYBER.2018.8688342>
26. F. Guo, H. Wang, Y. Tian, Multi-objective path planning for unrestricted mobile, in *2009 IEEE International Conference on Automation and Logistics*, (2009), 1046–1051. <https://doi.org/10.1109/ICAL.2009.5262574>
27. D. Gong, J. Zhang, Z. Yong, Multi-objective particle swarm optimization for robot path planning in environment with danger sources, *J. Comput.*, **6** (2011), 1554–1561. <https://doi.org/10.4304/jcp.6.8.1554-1561>
28. Y. Zhang, D. Gong, J. Zhang, Robot path planning in uncertain environment using multi-objective particle swarm optimization, *Neurocomputing*, **103** (2013), 172–185. <https://doi.org/10.1016/j.neucom.2012.09.019>

29. T. T. Mac, C. Copot, D. T. Tran, R. D. Keyser, A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization, *Appl. Soft Comput.*, **59** (2017), 68–76. <https://doi.org/10.1016/j.asoc.2017.05.012>
30. A. Hidalgo-Paniagua, M. Vega-Rodríguez, J. Ferruz, N. Pavón, Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach, *Soft Comput.*, **21** (2017), 1–16. <https://doi.org/10.1007/s00500-015-1825-z>
31. A. Hidalgo-Paniagua, M. Vega-Rodríguez, J. Ferruz, N. Pavón, MOSFLA-MRPP: Multi-objective shuffled frog-leaping algorithm applied to mobile robot path planning, *Eng. Appl. Artif. Intell.*, **44** (2015), 123–136. <https://doi.org/10.1016/j.engappai.2015.05.011>
32. D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, *Appl. Math. Comput.*, **214** (2009), 108–132. <https://doi.org/10.1016/j.amc.2009.03.090>
33. J. Li, M. Song, L. Wang, P. Duan, Y. Han, H. Sang, et al., Hybrid artificial bee colony algorithm for a parallel batching distributed flow-shop problem with deteriorating jobs, *IEEE Trans. Cybern.*, **50** (2020), 2425–2439. <https://doi.org/10.1109/TCYB.2019.2943606>
34. J. Li, Y. Han, A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in cloud computing system, *Cluster Comput.*, **23** (2020), 2483–2499. <https://doi.org/10.1007/s10586-019-03022-z>
35. L. Meng, C. Zhang, Y. Ren, B. Zhang, C. Lv, Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem, *Comput. Ind. Eng.*, **142** (2020). <https://doi.org/10.1016/j.cie.2020.106347>
36. L. Meng, K. Gao, Y. Ren, B. Zhang, H. Sang, C. Zhang, Novel MILP and CP models for distributed hybrid flowshop scheduling problem with sequence-dependent setup times, *Swarm Evol. Comput.*, **71** (2022). <https://doi.org/10.1016/j.swevo.2022.101058>
37. P. G. Asteris, M. Nikoo, Artificial bee colony-based neural network for the prediction of the fundamental period of infilled frame structures, *Neural Comput. Appl.*, **31** (2019), 4837–4847. <https://doi.org/10.1007/s00521-018-03965-1>
38. P. Duan, T. Wang, M. Cui, H. Sang, Q. Sun, Multi-person pose estimation based on a deep convolutional neural network, *J. Visual Commun. Image Rep.*, **62** (2019), 245–252. <https://doi.org/10.1016/j.jvcir.2019.05.010>
39. Z. Wang, R. Song, P. Duan, X. Li, EFNet: Enhancement fusion network for semantic segmentation, *Pattern Recogn.*, **118** (2021). <https://doi.org/10.1016/j.patcog.2021.108023>
40. M. H. Saffari, M. J. Mahjoob, Bee colony algorithm for real-Time optimal path planning of mobile robots, in *2009 Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control*, (2009), 1–14. <https://doi.org/10.1109/ICSCCW.2009.5379462>
41. P. Bhattacharjee, P. Rakshit, I. Goswami, A. Konar, A. Nagar, Multi-robot path-planning using artificial bee colony optimization algorithm, in *2011 Third World Congress on Nature and Biologically Inspired Computing*, (2011), 219–224. <https://doi.org/10.1109/NaBIC.2011.6089601>
42. D. Cui, X. Gao, W. Guo, Mechanism design and motion ability analysis for wheel/track mobile robot, *Adv. Mech. Eng.*, **8** (2016), 1–13. <https://doi.org/10.1177/1687814016679763>

43. J. Hao, J. Li, Y. Du, M. Song, P. Duan, Y. Zhang, Solving distributed hybrid flowshop scheduling problems by a hybrid brain storm optimization algorithm, *IEEE Access*, **7** (2019), 66879–66894. <https://doi.org/10.1109/ACCESS.2019.2917273>
44. J. Li, S. Bai, P. Duan, H. Sang, Y. Han, Z. Zheng, An improved artificial bee colony algorithm for addressing distributed flow shop with distance coefficient in a prefabricated system, *Int. J. Prod. Res.*, **57** (2019), 6922–6942. <https://doi.org/10.1080/00207543.2019.1571687>
45. Y. Du, J. Li, C. Li, P. Duan, A reinforcement learning approach for flexible job shop scheduling problem with crane transportation and setup times, *IEEE Trans. Neural Networks Learn. Syst.*, (2022). <https://doi.org/10.1109/TNNLS.2022.3208942>
46. J. Li, Y. Du, K. Gao, P. Duan, D. Gong, Q. Pan, et al., Hybrid iterated greedy algorithm for a crane transportation flexible job shop problem, *IEEE Trans. Autom. Sci. Eng.*, **19** (2022), 2153–2170. <https://doi.org/10.1109/TASE.2021.3062979>
47. Y. Du, J. Li, X. Chen, P. Duan, Q. Pan, Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem, *IEEE Trans. Emerg. Topics Comput. Intell.*, 2022. <https://doi.org/10.1109/TETCI.2022.3145706>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)