IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 28, 2018, accepted July 26, 2018, date of publication August 14, 2018, date of current version September 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2864188

Path Planning of Mobile Robot Based on Hybrid Multi-Objective Bare Bones Particle Swarm Optimization With Differential Evolution

JIAN-HUA ZHANG[1], YONG ZHANG[2], AND YONG ZHOU[3]
[1]School of Mechanical and Electrical Engineering, Xuzhou Institute of Technology, Xuzhou 221018, China
[2]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China
[3]School of Computer Science Technology, China University of Mining and Technology, Xuzhou 221116, China

Corresponding author: Jian-Hua Zhang (jianhuazh@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572505 and Grant 61473299, in part by the Natural Science Fund for Colleges and Universities in Jiangsu Province under Grant 12KJD510014, and in part by the Jiangsu Six Talents Peaks Project of Province under Grant DZXX-053.

ABSTRACT An improved path planning for mobile robots is proposed based on the hybrid multi-objective bare-bones particle swarm optimization with differential evolution. The mathematical model for robot path planning is firstly devised as a tri-objective optimization with three indices, i.e., the path length, the smoothness degree of a path, and the safety degree of a path. Then, a hybrid multi-objective bare bones particle swarm optimization is developed to generate feasible paths by combining infeasible paths blocked by obstacles with feasible paths via improved mutation strategies of differential evolution. In addition, a new Pareto domination with collision constraints is developed to select the personal best position of a particle according to the definition of the collision degree of a path. Simulation results confirm the effectiveness of our algorithm.

INDEX TERMS Robot, path planning, particle swarm optimization, differential evolution.

I. INTRODUCTION

Since mobile robot can be widely used in industry manufacturing, agriculture production, medical service and so on, there has been an increasing research interests in the field of robotics over the past few decades. The path planning of mobile robot is one of the most important issues of autonomous navigation, which includes four basic parts, perception, localization, path planning and motion control [1]. In general, the problem of robot path planning [2] can be considered as searching a collision free optimal or suboptimal trajectory according to some performance indices between the starting position and destination in a working environment with obstacles, which can be formulated as a typical constrained optimization problem [3].

In general, the existing robot path planning methods can be classified into two categories [1]–[5], i.e., classical algorithms and heuristic approaches. The main classical algorithms include cell decomposition, visibility graph, artificial potential field and sampling-based methods. Cell decomposition divides the free workspace of a robot into a series of cells, and an adjacency graph planner is adopted to find a collision free path. The visibility graph draws a mutually visible connection edge between each pairs of obstacle vertices and the start and target points in the plane if the connecting vertices of them can directly see each other, which can find the optimal shortest path in the graph for robots. The artificial potential field is a widely used method for single or multiple robots due to its simplicity and effectiveness. It builds a comprehensive potential field to generate an optimal path, which includes a repulsive potential field to repel the robot away from obstacles and an attractive potential field to pull the robot toward the target. The sampling-based methods mainly including probabilistic roadmaps method (PRM) and rapidly-exploring random tree (RRT) have received an increased interest due to its capability of providing fast solutions in complex workspace. However, the classical approaches may suffer some following disadvantages, e.g., more computation in large workspace, trapping at some local minima and unsuitability for dynamic or uncertain environments.

Heuristic approaches for robot path planning include neural network [6], [7], genetic algorithms(GA) [8]–[10], ant colony optimization (ACO) [11]–[13], Firefly

algorithm [14], [15], Bacterial foraging optimization [16]–[18], particle swarm optimization [2], [3], [19]–[25], etc. These path planning methods can produce a robot path based on various optimization algorithms inspired from organism behaviors and different fitness function and constraints in a workspace.

Bare bones particle swarm optimization (BBPSO) [26] is a population-based heuristic search algorithm derived from basic particle swarm optimization (PSO). It omits the velocity updating section and only samples and explores the searching space to generate particle positions by a Gaussian distribution with the mean and deviation of personal best position and global best position. The advantages of parameter-free make it more easier and more efficient than basic particle swarm optimization. Now, particle swarm optimization and bare bones particle swarm optimization have been applied to different scenarios of robot path planning. However, these methods still suffer from certain limitations, such as the stagnation of the population, premature convergence of the population, and so on.

This paper mainly presents an improved hybrid multi-objective bare bones particle swarm optimization (HMOBBPSO) for the problem of robot path planning. Firstly, the path planning is mathematically formulated as a constrained multi-objective optimization problem with three indices, i.e. the path length, the smoothness degree of a path and the safety degree of a path. Then, an improved multi-objective bare bones particle swarm optimization algorithm of solving the above model is developed. In this algorithm, a new Pareto domination with collision constraints is developed to select the personal best position of a particle according to the definition of the collision degree of a path. An improved mutation strategy of differential evolution is designed to improve the feasibility of an infeasible path blocked by obstacles and constrained by a heading angle of a physical robot. Finally, simulation results confirm the effectiveness of our algorithm.

The contributions of this paper are as follows: (1) we consider the problem of mobile robot path planning and formalize it as a multi-objective optimization problem with constraints; (2) we present a multi-objective BBPSO algorithm with a new Pareto domination and an improved mutation operation to settle the above problem effectively; (3) we verify the effectiveness of our method by a series of simulations.

The remainder of this paper is arranged as follows. The related works on robot path planning are demonstrated briefly in Section 2. The formulation model of mobile robot path planning is investigated in Section 3. Our multi-objective BBPSO for robot path planning is involved and explained in Section 4. The simulations of the proposed method are conducted in section 5. Finally, Section 6 concludes and summarizes this paper.

II. RELATED WORK

The classic robot path planning may be prone to trap into some local minima and needs more computation time in large workspace. The heuristic algorithms have been proposed to overcome the disadvantages of the classic methods in robot fields.

Artificial neural networks have been used to realize path planning for mobile robots in various environments [6], [7]. In [6], Q-learning and a neural network planner are adopted to solve obstacle avoidance and trajectory planner for mobile robot. The robot and obstacles in the environment are firstly modelled by virtual reality modelling language. The collision detection and trajectory generation are then implemented by Pos-net neural network, whose inputs and outputs are the current position of the robot, the sample time and the matrix of Q-value, and 3 element vectors containing the position of the robots and sample time separately. In [7], two artificial neural networks, radial basis function neural network and novel wavelet neural network, are put forwarded to generate the collision free path for four-wheeled non-holonomic differential driven mobile robot. The inputs and outputs of the neural networks are the minimum distances between the robot and obstacles detected by the left, right and front ultrasonic sensors of the robot, and the driving angle separately.

Genetic algorithm is a heuristic optimization based on natural selection, crossover and mutation operations applied to the robot path planning domain. In [8], a modified genetic algorithm with diversity measure is presented to optimize a Bezier curve-based path. A binary chromosome is used to encode the control point and a validation operation is also demonstrated to valid whether the path is to be selected as a solution or not. In [9], two-Kinect cameras fixed on the roof of the workroom are used to model the environments of the robots. And genetic algorithm with random crossover and eight different mutations to the robot path is adopted to find the shortest, the least error and deviation paths. Meanwhile adaptive fuzzy-logic controller is designed to keep track of a mobile robot on the desired path.

Ant colony optimization is a kind of global natural evolutionary algorithm inspired by the foraging behaviors of the real ant colony. Liu *et al.* [11] proposed a global path planning of a mobile robot in a grid environment model based on ant colony algorithm combined with artificial potential field and geometric local optimization. The potential field force from the obstacles and the destination is used to guide for the path pheromone diffusions. Then the generated paths are smoothed and polished by the geometric local to eliminate the cross paths, the circular paths and the saw tooth paths. In [12], a new heuristic formulation is proposed to update the selection probability of the next node based on the distances among the current gird, the optional grids and the destination grid. And the pheromone update is also renewed according to the current gird, the optional grids and the grids the robot passed by but did not stay in between the current grid and the optional grids.

Bacterial foraging optimization with chemotaxis, reproduction, elimination and dispersal operations was proposed by Prof. Passino to settle various field problems in 2002 [27]. In [16], the chemotaxis operation of Bacterial foraging

optimization is employed to randomly generate and distribute some virtual particles around a robot. And the best particle is added to the desired path of the robot according to the distance error and the cost function error with respect to the obstacles and the goal in the workspace. In [17], mobile robot path planning based on bacterial chemotaxis is put forward by analogizing the headway and steering operations of the robot and swim and tumble actions of bacterial. Furthermore, the random tumble is altered as the intelligent direction selection according to the weighted fitness function of target attraction and obstacle repelling from the sensors mounted on the robot.

Particle swarm optimization is another popular heuristic algorithm, which has been applied to many different applications especially in mobile robotics. At present, robot path planning based on PSO can be fallen into one of two categories, those with single objective fitness function and those with multiple objective fitness functions. In [19], aiming at the workspace of mobile robots with uncertain danger sources, a constrained bi-objective particle swarm optimization is designed to tackle the problem. After modeling the map of the environment by dividing the workspace into a series of equal segments, two performance criterions, i.e., the length of path, the interval risk degree of path, are defined to formulate the path planning problem as a multi-objective optimization with constraints. And a mechanism of a feasible archive and an infeasible archive and a random sampling method of selecting the global best are incorporated into the particle swarm optimization to improve the diversity of population and balance the exploration and exploitation of the swarm.

In [20], a random-disturbance self-adaptive particle swarm optimization with a weighted sum fitness function of the path length and the smoothness of a path is presented to obtain a global path for a mobile robot in the workspace. To avoid trapping in the local optimum, a perturbed global best updating mechanism is added to adjust the social component of a particle based on the normal distribution with the mean and standard deviation of the particles. A fine-tuning mechanism to three control parameters, i.e., the inertial weight, the cognitive and social acceleration coefficients, is developed to well trade off the exploration and exploitation capabilities of a particle.

In [21], a global path planning with three-layer structures is proposed to obtain a feasible and safe optimal path. The triangular decomposition method is firstly used to quickly establish a geometric free space of the robot. Then the Dijkstra algorithm is applied to find a collision-free path according to the graph constructed from the triangular decomposition. Afterwards a constrained multi-objective particle swarm optimization with the indices of the path length and the path smoothness is introduced to optimize the path points from Dijkstra algorithm.

In [22], a smooth global path planning for mobile robots with kinematic constraints is presented based on modified particle swarm optimization combined with the $\eta 3$-splines. The adaptive random fluctuations indicated by four evolutionary states are incorporated into particle swarm optimization to adjust and balance the cognitive and social components according to the maximal and minimal mean distances of the particles. Then a double-layer path planning approach is formed to generate a feasible linear and smooth path with the single objective fitness function of the path length and the constraints of the continuous curvature as well as the continuous curvature derivative.

In [23], a path planning for mobile robot is proposed based on a hybridization of the social essence of particle swarm optimization and the motion mechanism of gravitational search algorithm(GSA). A hybrid velocity formulation is derived from the random combination of PSO velocity component and GSA acceleration component. Afterwards the overall fitness function is defined by weighted sum of four single objective function, i.e., the path length, the repulsive function from the obstacles, the prediction function of the dynamic obstacles and the smoothness of the path, to generate multiple feasible paths for each robot. Das *et al.* [3] also proposes a similar hybrid approach of particle swarm optimization and differential evolutionary algorithm for robot path planning. A difference vector calculated by the particle and its two neighbors is inserted into the velocity updating equation of PSO instead of the local best component to produce some additional exploration capability. Moreover, a random mutation operation will shift the particle to a new position to escape the local minima and keep swarm moving when a particle does not change its position in a predefined number of iteration in the search space.

Su and Wang [24] also proposes a global path planning with three-layer structures i.e., the MAKLINK graph, Dijkstra algorithm and random coding particle swarm optimization, for a mobile robot in a two-dimensional environment. The MAKLINK graph theory is firstly adopted to establish the free space model of a mobile robot. Then the Dijkstra algorithm is used to search for a sub-optimal path according to the weight undirected graph with the adjacency matrix. Afterwards a random coding particle swarm optimization with the crossover operator of genetic algorithm is developed to optimize the paths from Dijkstra algorithm.

In [25], an adaptive Gaussian parameter updating rules are imposed to the inertia weight and the two acceleration parameters to avoid trapping into the local optimum and balance the exploration and exploitation in the search space. And the improved particle swarm optimization is used to plan a feasible path with the weighted sum fitness function combined by three objective functions, i.e., the path length, the danger degree of a path and the smoothness of a path.

However, most of the above-mentioned methods on robot path planning usually have been formulated as a single objective optimization problem, or a weighted sum one formulated by multiplying multiple objective functions with different weights. But the latter is essentially a single objective optimization method and has some inevitable problems, specifically, for its lack of detecting non-convex regions to miss

some Pareto solutions. Moreover the correct weighting and priori knowledge may not always be clear upfront, which makes the method be just as costly as the multi-objective formulation. In this paper, an improved multi-objective bare bones particle swarm optimization is proposed to solve the path planning of the mobile robot. We formulate the path planning as a constrained tri-objective optimization problem with the indices of the path length, the smoothness degree of a path and the safety degree of a path. And a new Pareto domination and an improved mutation operation are incorporated into the particle swarm optimization to deal with the above problem.

III. MATHMATICAL MODEL OF ROBOT PATH PLANNING

This section focuses on the modeling of the robot workspace, path encoding and the formulation of the robot path planning in a two-dimensional workplace with obstacles. And a tri-objective optimization model of robot path planning is developed based on the environment model. Furthermore, a hybrid multi-objective BBPSO algorithm with a new Pareto domination and an improved mutation operation is proposed to search for multiple robot paths simultaneously.

A. WORKSPACE MODELING

To accomplish the effective path planning, the environment map of the workspace should be firstly converted and stored into computational representation to make the robot understand and see the surrounding environments. The common description methods of the environment map include cell and grid decomposition, Maklink graph, Voronoi diagram, line-equivalent-division method, etc. And this paper adopts the method in [2] and [19] to model the robot workspace due to its simplicity and insensitivity to the shapes of obstacles.

In the global coordinate system O-XY in Fig. 1, the polygons entities, denoted as n, are some obstacles, S and T represent the start and the target points of a robot, respectively. A local coordinate system O$'$-$X'Y'$ is established by setting S as the origin, the line between S and T as X' axis and the line perpendicular to X' axis as Y' axis. The transformation equation between the global coordinate system and the local coordinate system is as following:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_S \\ y_S \end{bmatrix} \tag{1}$$

where φ is the intersection angle from the X axis to the line ST, x_S and y_S are the coordinates of start point S in the global coordinate system O-XY, x' and y' are the coordinates of the point (x, y) in the local coordinate system O$'$-$X'Y'$.

The segment ST is evenly split up into $m+1$ intervals by m points in X' axis. Then a series of vertical lines crossing the m points in sequence will be formulated as shown in Fig. 1. The robot path, denoted as $PH = S, p_1, p_2, \cdots p_m, T$, can be uniquely set up by randomly sampling on these vertical lines of $l_1, l_2, \cdots l_m$. Furthermore, denote the start and the target points as p_0 and p_{m+1}, respectively, then a robot path can be represented as $PH = p_0, p_1, p_2, \cdots p_m, p_{m+1}$.

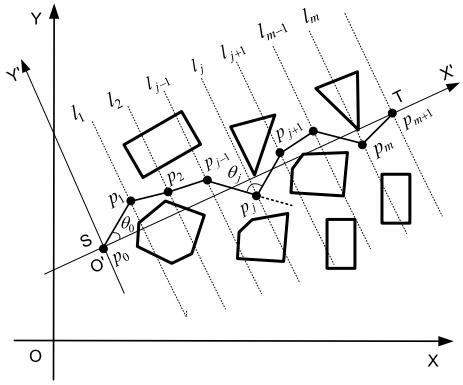

FIGURE 1. Model of robot workspace.

To reach the target safely from the start point, the robot must not collide with the obstacles in the workspace. For a robot path, we calculate its collision times with obstacles in the workspace to assess its feasible or infeasible performances. If the segment p_jp_{j+1} of a robot path collides with some obstacle k, its attribute value, denoted as cv_{jk}, $j = 0, 1, \cdots m$, $k = 1, \cdots n$, is set to be one; otherwise, the value is zero. Then the total collision degree of a path PH is formulated as follows:

$$cv(PH) = \sum_{j=0}^{m} \sum_{k=1}^{n} cv_{jk} \tag{2}$$

B. MATHEMATICAL MODEL OF ROBOT PATH PLANNING

The path planning of a robot is to find an optimal or sub-optimal collision-free path from the start point to the target point according to some performances. In this study, three performance criteria, i.e. the length of a path, the smoothness degree of a path and the safety of a path, are considered. The shorter a robot path is, the less the travel time and the energy consumption are. The farther a robot path from the obstacles is, the safer the robot path is.

According to the workspace model in subsection III(A), the length of a robot path, denoted as $f_L(PH)$, can be mathematically devised as follows:

$$f_L(PH) = \sum_{j=0}^{m} d(p_j, p_{j+1}) \tag{3}$$

where $d(\cdot, \cdot)$ is the Euclidean distance of two points p_j and p_{j+1}. In the local coordinate system O$'$-$X'Y'$, the length of a path can be approximately calculated as:

$$\begin{aligned} f_L(PH) &= \sum_{j=0}^{m} \sqrt{(x'_{j+1} - x'_j)^2 + (y'_{j+1} - y'_j)^2} \\ &= \sum_{j=0}^{m} \sqrt{(\frac{d(S,T)}{m+1})^2 + (y'_{j+1} - y'_j)^2} \end{aligned} \tag{4}$$

When a robot wanders through the environment, we wish the path can be as smooth as possible, which can in turn

decrease the energy consumption of a robot. The smoothness degree of a path, denoted as $f_S(PH)$, is formulated as follows:

$$f_S(PH) = \max_{j=1,2,\cdots m} (180^\circ - \theta_j, \theta_0) \tag{5}$$

where θ_j is the intersection angle from line $p_{j-1}p_j$ to line p_jp_{j+1}, and θ_0 is the intersection angle between the segment p_0p_1 and the X' axis assuming that the initial head orientation of a robot is directly oriented towards the target point. The maximum deviation angle is adopted to describe the smoothness rather than the summation of the deviation angles because the latter cannot be a good representation to the smoothness [28].

When a robot approaches the goal from the start point in the environment, the robot cannot collide with the obstacles in the workspace and should reach the destination safely. To estimate how insecurely the robot moves close to the obstacles, the safety degree of a path, denoted as $f_O(PH)$, is formulated as follows:

$$f_O(PH) = \exp(-\alpha \cdot d(PH, O_k)) \tag{6}$$

where α is an adjustable parameter, $d(PH, O_k) = \min_{j=0,1,\cdots m} \min_{k=1,2,\cdots n} d(p_jp_{j+1}, O_k)$ is the minimum distance of the segment p_jp_{j+1} from an obstacle O_k. The exponent function is aimed to convert the maximum optimization to minimum one for coinciding with the other two indices.

Synthesizing the formulas (4), (5) and (6), the mathematical optimization model of the robot path planning can be formulated as the following tri-objective optimization with a constraint:

$$\begin{aligned} &\min f(PH) = (f_L(PH), f_S(PH), f_O(PH)) \\ &s.t.\ cv(PH) = 0 \end{aligned} \tag{7}$$

where the constraint $cv(PH) = 0$ shows avoiding collision with the obstacles in the workspace on the path of a robot.

When solving the above model, there is a conflict among the three different objectives. The length, smoothness and safety degree of a path are dependent in a way that a solution may be optimal for a sub-objective, whereas be sub-optimal for another sub-objective, which makes the algorithm difficult to get the best solutions for all objectives at the same time. A short robot path usually implies the generated paths is relative smooth, but also relative dangerous and close to the obstacles in the workspace. In other words, the length of a robot path $f_L(PH)$ is smaller, the smoothness degree of the path $f_S(PH)$ is smaller too. However, the robot may get closer to the obstacles in the workspace, and the safety degree of the path gets worse and worse. So it is necessary to find a Pareto set to keep balance among all conflicting objectives.

IV. ROBOT PATH PLANNING BASED ON HYBRID MULTI-OBJECTIVE BARE BONES PARTICLE SWARM OPTIMIZATION WITH DIFFERENTIAL EVOLUTION

In the following subsection, a hybrid multi-objective bare-bones particle swarm optimization with differential evolution (HMOBBPSO) is introduced to solve the above problem of robot path planning. Aiming at the infeasible paths blocked with obstacles, an improved mutation operation of differential evolution is designed to combine the infeasible path with feasible path to improve the feasibility of a path. To evaluate the quality of a particle in a constrained multi-objective optimization problem, a new constrained Pareto domination relation with variable objectives is developed to select personal best position of a particle and update the infeasible solutions archive.

A. PARTICLE ENCODING

As a robot path can be represented as $PH = S, p_1, p_2, \cdots p_m, T$ in subsection III(A), the coordinate values of the robot path are uniquely determined by the Y'-axis values of those lines under the premise that the X'-axis values of the vertical lines of $l_1, l_2, \cdots l_m$ have been configured beforehand according to the size of the obstacles in a given environment. Then an m-dimensional vector $y'_1, y'_2 \cdots y'_m$ is taken as the decision variables of the path planning issues, denoted as

$$X_i = (y'_1, y'_2 \cdots y'_m) = (x_1, x_2 \cdots x_m) \tag{8}$$

B. SELECTING PERSONAL BEST POSITION

Due to the characteristics of multi-objective optimization with constraints, a new constrained Pareto domination with variable objectives is defined to update personal best positions and the archives of the solutions based on the definition of the existing Pareto domination relationship.

Denote $X_i(t)$ and $XP_i(t)$ as the position and its personal best position of a particle in the t-th iteration, the constrained Pareto domination with variable objectives is arranged to renew the locally optimal position of a particle on the basis of the definitions of aforementioned three performance indices and the collision degree of a path.

1) If these particles are both feasible solutions, i.e., $cv(X_i(t+1)) = 0$ and $cv(XP_i(t)) = 0$, three objective functions, i.e., the path length, the smoothness degree of a path and the safety degree of a path, are used to construct tri-objective Pareto domination to update the personal best position of a particle. In this scenario, the offspring $X_i(t+1)$ is chosen as its new locally optimal particle if $f(X_i(t+1)) \prec_F f(XP_i(t))$ is satisfied, where $\prec_F$ represents Pareto domination for feasible solutions. If $X_i(t+1)$ and $XP_i(t)$ do not dominate each other, its new locally optimal particle is randomly selected from $X_i(t+1)$ and $XP_i(t)$.

2) If these particles are both infeasible solutions, i.e., $cv(X_i(t+1)) \neq 0$ and $cv(XP_i(t)) \neq 0$, four objective functions, i.e., the path length, the smoothness degree of a path, the safety degree of a path and the collision degree of a path, are used to construct quad-objective Pareto domination to update the personal best position of a particle. In this scenario, the offspring $X_i(t+1)$ is chosen as its new locally optimal particle if $f(X_i(t+1)) \prec_{IF} f(XP_i(t))$ is satisfied, where $\prec_{IF}$ represents Pareto domination for infeasible solutions. If $X_i(t+1)$ and $XP_i(t)$ do not dominate each other, its new

locally optimal particle is randomly selected from $X_i(t+1)$ and $XP_i(t)$.

3) If one particle is feasible solution and another particle is infeasible solution, i.e., $cv(X_i(t+1)) = 0$ and $cv(XP_i(t)) \neq 0$, $X_i(t+1)$ is chosen as its personal best position. Otherwise, keep the personal best position unchanged.

C. UPDATING ARCHIVES AND SELECTING GLOBAL BEST POSITION

The archives are used to select the global best position of a particle during the evolution progress. Here the methods in [2] and [19] are employed to get non-dominated solutions for updating the feasible archive with a capacity of N_a and the infeasible archive with a capacity of N'_a.

The global best position of a particle, denoted as $XG_i(t)$, is the best solution found by the neighbors of a particle. Similarly, we adopt our previous methods in [2] and [19] to update the global best position of a particle from the feasible solutions archive and/or infeasible solutions archive on basis of the conception of crowding distance.

D. IMPROVED MUTATION OPERATIONS TO INFEASIBLE SOLUTIONS BASED ON DIFFERENTIAL EVOLUTION

Differential evolution is a simple and powerful stochastic evolutionary algorithm [29]–[31], which has been shown to be efficient and effective performances in solving numerous real-world problems. Generally speaking, differential evolution utilizes three operations, i.e., mutation operation, crossover operation and selection operation, to search for the optimal solutions at each generation.

For the infeasible particle X_i, i.e., the paths blocked by obstacles in the environment, an improved mutation operation with the probability $p_m(i)$ is formulated in (9) to accelerate the search progress based on DE/rand/1, DE/rand/2 and DE/current-to-best/1.

$$p_m(i) = \frac{\exp(cv(X_i(t))) - \exp(-cv(X_i(t)))}{\exp(cv(X_i(t))) + \exp(-cv(X_i(t)))} \tag{9}$$

If a random number *rand* is smaller than or equal to $p_m(i)$, we randomly extract two feasible solutions from the feasible archive, denoted as X_{r1} and X_{r2}, as the difference vector. Then a new solution X'_i is generated by combining the differential vector to an infeasible solution X_i in the population. The improved updating equation of DE/rand/1 is as following:

$$X'_i = X_i + F \cdot (X_{r1} - X_{r2}) \tag{10}$$

where F is a user-defined scaling factor to control the magnitude of difference vector, $r1$ and $r2$ are distinct integers randomly selected from $[1, N_a]$.

Similarly, we randomly extract four feasible solutions from the feasible archive, and then the updating equation of DE/rand/2 is as following:

$$X'_i = X_i + F \cdot (X_{r1} - X_{r2}) + F \cdot (X_{r3} - X_{r4}) \tag{11}$$

where F is a user-defined scaling factor to control the magnitude of difference vector, $r1$, $r2$, $r3$ and $r4$ are distinct integers randomly selected from $[1, N_a]$.

Furthermore, we randomly extract two feasible solutions from the feasible archive and one feasible solution with maximum crowding distance from the feasible archive, then the updating equation of DE/current-to-best/1 is as following:

$$X'_i = X_i + F \cdot (XG_i - X_i) + F \cdot (X_{r1} - X_{r2}) \tag{12}$$

Fig. 2 demonstrates the pseudo code of the improved mutation operation to infeasible solutions based on DE/rand/1. If a random number *rand* is smaller than or equal to $p_m(i)$ at each iteration, the infeasible paths are mutated to increase the feasibility of a path according to the improved mutation operations of differential evolution. The larger the collision degree of a path is, the larger the mutation probability of an infeasible path is.

Function Mutation Operation
Input: Infeasible solutions **Output:** the solutions after mutated

```
FOR i = 1 to N
    IF cv(X_i(t)) > 0
        Calculate the mutation probability by equation (9)
        IF random < p_m(i)
            Select randomly two feasible solutions from
              feasible archive
            FOR j = 1 to m
                X'_ij = X_ij + F•(X_r1j − X_r2j)
            ENDFOR
        ENDIF
    ENDIF
ENDFOR
```

FIGURE 2. The pseudo code of mutation operation.

E. STEPS OF ALGORITHMS

The steps of the improved path planning for mobile robots based on hybrid multi-objective bare-bones particle swarm optimization with differential evolution are described as follows.

Step 1 Model the workspace of robot and set relative parameters, including the division number m, the population size N, the maximum iteration $t_{\max}$, the parameter α, the sizes of the feasible and the infeasible archives N_a and N'_a.

Step 2 Generate initialization population randomly and set the current position as its personal best position $XP_i(t)$ for each particle. Select the global best position of each particle according to the method in subsection IV(C).

Step 3 Update the position of a particle according to the following equations [32]:

$$\begin{cases} V_{i,j} = \begin{cases} N\left(\dfrac{XP_{i,j}+XG_{ij}}{2}, \left|XP_{i,j}-XG_{ij}\right|\right), & \text{if } U(0,1) < 0.5 \\ XP_{i,j}, & \text{otherwise} \end{cases} \\ X_{i,j}(t+1) = V_{i,j}(t+1) \end{cases} \tag{13}$$

where $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, m$, $N(\cdot)$ is a Gaussian distribution with mean $(XP_{i,j} + XG_{ij})/2$ and standard deviation $\left|XP_{i,j} - XG_{ij}\right|$, and $U(0,1)$ is a random variable with uniform distribution in [0, 1].

Step 4 Update the feasible and the infeasible archives by the method in subsection IV(C).

Step 5 Perform the mutation operation to infeasible particles based on differential evolution proposed in subsection IV(D).

Step 6 Update the personal best positions and global best positions for each particle according to the methods in subsection IV(B) and IV(C) separately.

Step 7 If the termination condition is satisfied, stop the algorithm and output the optimal results. Otherwise, go to Step3.

V. SIMULATION EXPERIMENTS AND ANALYSIS

To validate the proposed algorithm in solving path planning, the simulation experiments are conducted by MATLAB software on a Core processor computer. The performances of the proposed algorithm are compared with those of multi-objective particle swarm optimization with constriction factor (MOCPSO)[33], multi-objective particle swarm optimization with linearly decreased inertia weight (MOLPSO) [34], multi-objective particle swarm optimization with time-vary acceleration coefficients (MOLTPSO) [35] under two simulation scenarios in the 100×100 workspace. The parameters of the proposed method HMOBBPSO are set as following, $N = 50$, $N_a = 10$, $N'_a = 10$, $\alpha = 1$, $t_{\max} = 200$. The simulation parameters for other algorithms are extracted from the corresponding literatures and shown in Table 1. All algorithms are independently run for 30 times to collect the statistical results. We adopt the C metric, denoted as $C(\cdot, \cdot)$, the spacing metric, denoted as $SP(\cdot)$, and the hyper-volume metric, denoted as $S(M, r)$ to analysis the optimization results [36].

TABLE 1. Parameters of other algorithms.

Algorithm	Inertia Weight	Cognitive Parameters
CMOPSO	$\omega = 0.729$	$c_1 = c_2 = 1.49$
LMOPSO	$\omega_{\max} = 0.9, \omega_{\min} = 0.4$	$c_1 = c_2 = 2$
LTMOPSO	$\omega_{\max} = 0.9,$ $\omega_{\min} = 0.4$	$c_{1i} = c_{2f} = 2.5$ $c_{1f} = c_{2i} = 0.5$

A. EXPERIMENT 1

For the first scenario shown in Fig. 3 to Fig. 5, there are twelve convex obstacles in the workspace. The coordinates of the start and the target points of robot are S (10.311, 68.857) and T (90.593,29.846), respectively. The line between the start and the target points is equally divided into nine segments. The scaling factor F of differential evolution is set as $F = 0.5$.

Tables 2 to 4 illustrate the hyper-volume metric and the spacing metric corresponding to the maximal value of $S(M, r)$ for this environment; while Fig. 3 to Fig. 5 demonstrates the optimal robot paths with the maximal values of $S(M, r)$ from different methods. Table 5 shows the C metric of different methods.

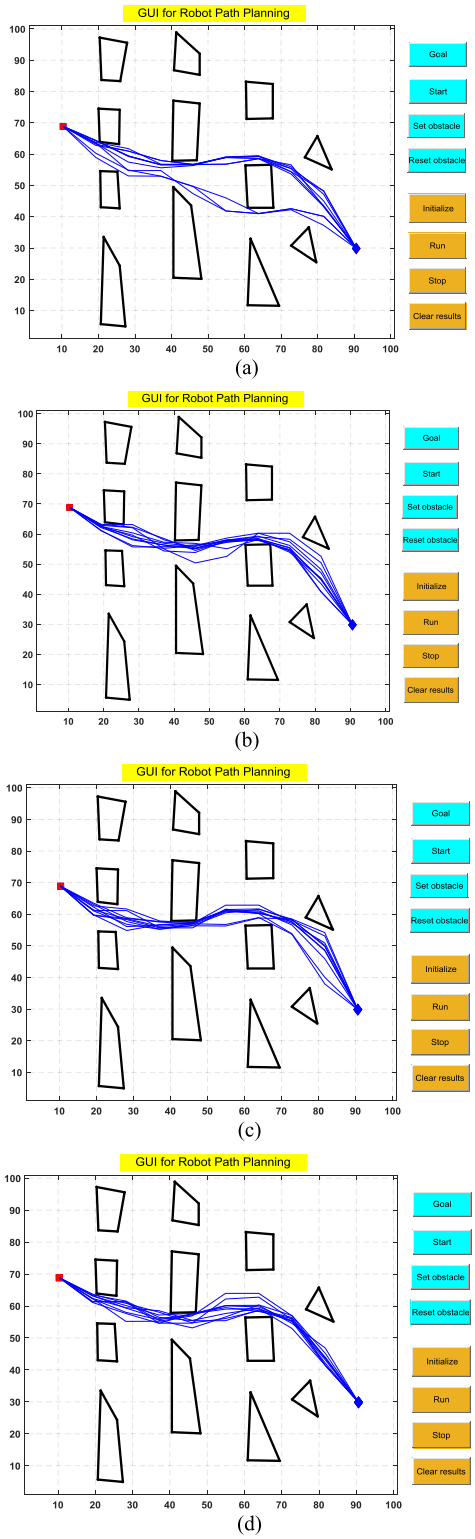

FIGURE 3. Optimal robot paths of different algorithms with DE/rand/1: (a) HMOBBPSO. (b) MOLPSO. (c) MOLTPSO. (d) MOCPSO.

It can be seen from Tables 2 to 4 that the average hyper volume values of the proposed HMOBBPSO method are

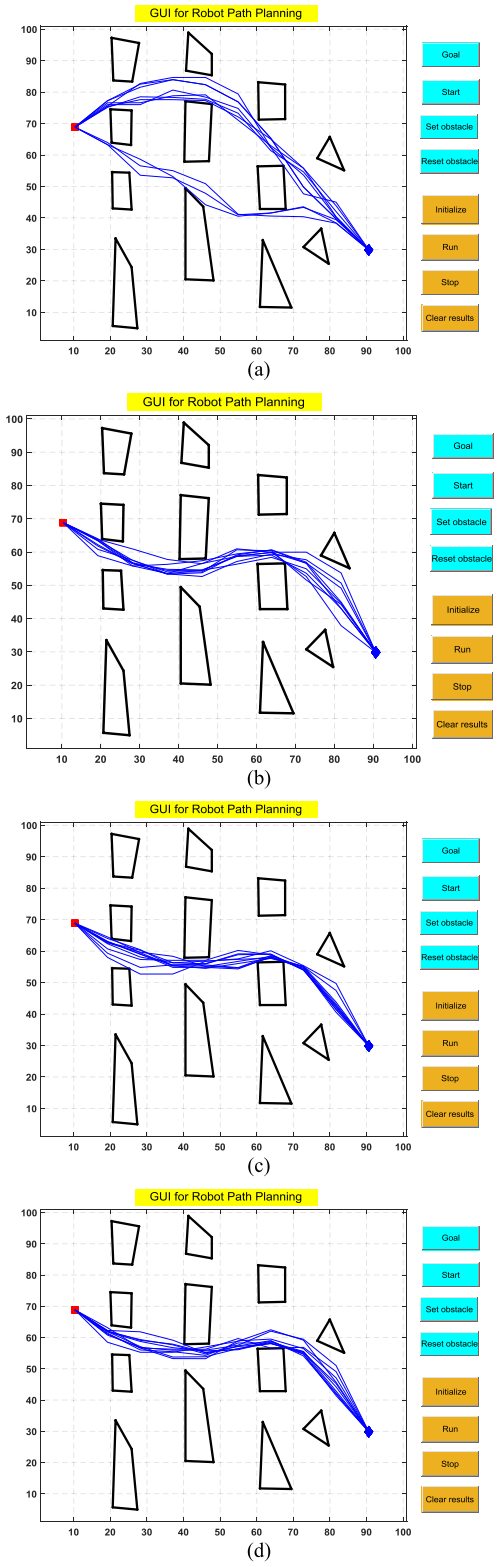

FIGURE 4. Optimal robot paths of different algorithms with DE/rand/2: (a) HMOBBPSO. (b) MOLPSO. (c) MOLTPSO. (d) MOCPSO.

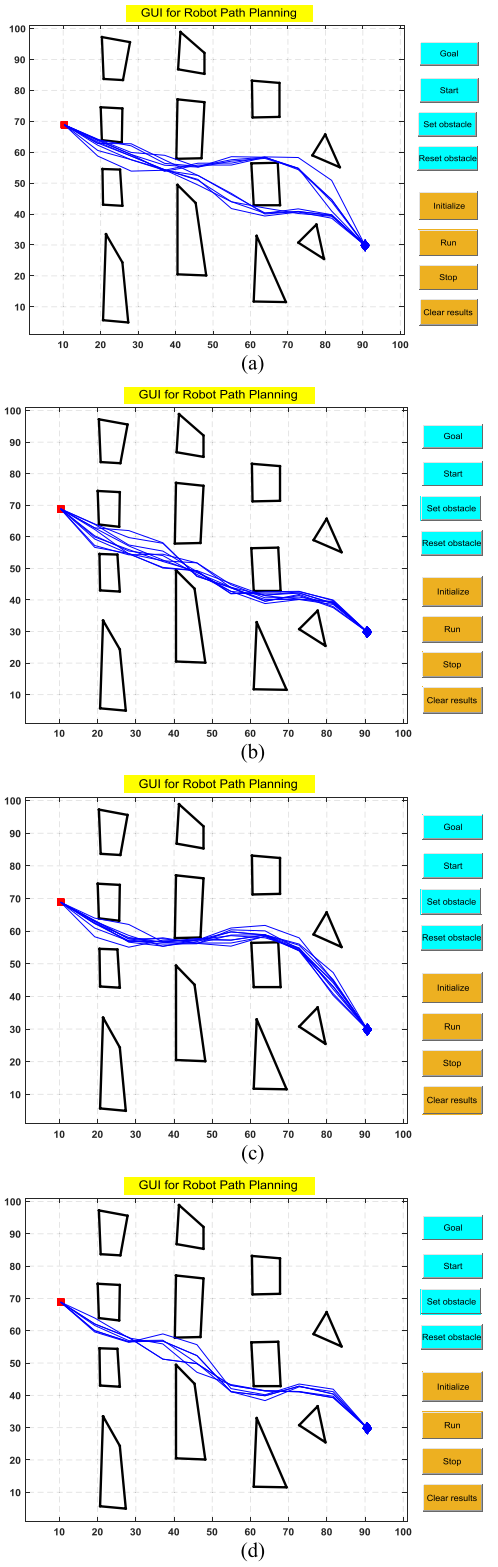

FIGURE 5. Optimal robot paths of different algorithms with DE/current-to-best/1: (a) HMOBBPSO. (b) MOLPSO. (c) MOLTPSO. (d) MOCPSO.

overall better than those of other methods. The average hyper volume values of MOLTPSO method with three kinds of mutation operations are close to each other, which can be concluded from the Fig. 3(c), Fig. 4(c) and Fig. 5(c) in some extent. The MOCPSO method gains the worst average hyper volume values when the mutation operations are applied

TABLE 2. Statistical results of hyper-volume metric and spacing metric based on DE/rand/1.

Algorithm	Maximum $S(M,r)$	Minimum $S(M,r)$	Average $S(M,r)$	$SP(\cdot)$
HMOBBPSO	227.3	11.927	75.306	0.2876
MOLPSO	207.16	8.4409	58.963	0.0908
MOLTPSO	203.82	18.099	76.963	0.2085
MOCPSO	167.19	13.915	61.183	0.1159

TABLE 3. Statistical results of hyper-volume metric and spacing metric based on DE/rand/2.

Algorithm	Maximum $S(M,r)$	Minimum $S(M,r)$	Average $S(M,r)$	$SP(\cdot)$
HMOBBPSO	417.59	12.213	83.288	0.1926
MOLPSO	197.21	24.988	72.019	0.1229
MOLTPSO	214.38	10.289	71.057	0.1622
MOCPSO	187.79	0.0876	63.018	0.1573

TABLE 4. Statistical results of hyper-volume metric and spacing metric based on DE/current-to-best/1.

Algorithm	Maximum $S(M,r)$	Minimum $S(M,r)$	Average $S(M,r)$	$SP(\cdot)$
HMOBBPSO	256.25	1.5728	76.511	0.0845
MOLPSO	212.54	1.0662	73.412	0.1560
MOLTPSO	178.78	16.353	76.272	0.1557
MOCPSO	214.16	0.71278	61.786	0.3572

into the algorithm. The proposed HMOBBPSO method with DE/current-to-best/1 takes the best spacing metric 0.0845, which can be observed from the relative smooth and safe paths in the Fig. 5(a).

For different algorithms with DE/rand/1 in Table 2, MOLTPSO algorithm ranks the first place with the maximal average hyper volume value 76.963, but the average hyper volume value of HMOBBPSO algorithm is just adjacent to the value of MOLTPSO algorithm and the maximal hyper volume value of HMOBBPSO is superior to those of other three methods. With respect to the spacing metric, MOLPSO algorithm obtains the first rank with the value 0.0908, and the MOCPSO algorithm gains the second rank with the value 0.1159.

For different algorithms with DE/rand/2 in Table 3, HMOBBPSO algorithm achieves the best results with the maximal hyper volume value 417.59 and the maximal average hyper volume value 83.288. The MOLPSO and MOLTPSO algorithms rank the second and the third places with the similar values, 72.019 and 71.057. About the spacing metric, the four algorithms do not show very large gap and the MOLPSO algorithm gives the best value 0.1229.

For different algorithms with DE/current-to-best/1 in Table 4, HMOBBPSO algorithm acquires the best hyper volume values, 256.25 and 76.511, and the best spacing metric 0.0845. The spacing metric value 0.0845 of the HMOBBPSO algorithm is far superior to those of other three algorithms, which shows the distribution of the proposed algorithm is better than that of other methods.

The values of C metric in Table 5 show the solutions of HMOBBPSO algorithm overlap most solutions of other three algorithms. The values of C metric in columns 2 and 3 display that the HMOBBPSO algorithm can obtain more better non-inferior solutions with DE/rand/1 and DE/rand/2 mutation operations than other three algorithms with the same mutation operations. The C metric values of $C(\text{HMOBBPSO}, \text{MOLPSO}) = 1$, $C(\text{HMOBBPSO}, \text{MOLTPSO}) = 0.8$ and $C(\text{HMOBBPSO}, \text{MOCPSO}) = 1$ show that the coverage and the smoothness of HMOBBPSO is far superior to that of other three algorithms, which can be validated by the maximal hyper volume values 417.59, 197.21, 214.38 and 187.79 in table 3.

TABLE 5. C metric of different algorithms.

Algorithm	DE/rand/1	DE/rand/2	DE/current-to-best/1
HMOBBPSO	0.5	1	0.6000
MOLPSO	0	0	0
HMOBBPSO	0.7	0.8	0.2000
MOLTPSO	0.1	0	0.1000
HMOBBPSO	0.6	1	0.5714
MOCPSO	0.2	0	0.1000

It can be observed from Fig. 3 to Fig. 5 that all the algorithms are efficient in finding multiple non-dominated robot paths, whereas the HMOBBPSO algorithm can search more feasible robot paths than other three algorithms. The Fig. 3(a), Fig. 4(a) and Fig. 5(a) are all produce multiple different paths, whereas the paths in Fig. 4(a) take a devious routes and Fig. 3(a) and Fig. 5(a) generate similar robot paths, which can be seen from the maximal hyper volume values 417.59 in Table 3, 256.25 in Table 4 and 227.3 in Table 2. The rest corresponding robot paths in Fig. 3(b-d), Fig. 4(b-d) and Fig. 5(b-d) gains similar optimized results, which can be validated by the corresponding maximal hyper volume values in Tables 2 to 4.

B. EXPERIMENT 2

For the second scenario shown in Fig. 6 to Fig. 8, there are also twelve convex obstacles including two traps in the workspace. The coordinates of the start and the target points of robot are S (10.155, 59.883) and T (90.437,59.883), respectively. The line between the start and the target points is equally divided into nine segments. The scaling factor F of differential evolution is set as $F = 0.7$.

Tables 6 to 8 illustrate the hyper-volume metric and the spacing metric corresponding to the maximal value of $S(M, r)$ for this environment; while Fig. 6 to Fig. 8 demonstrates the optimal robot paths with the maximal values of $S(M, r)$ from different methods. Table 9 shows the C metric of different methods.

TABLE 6. Statistical results of hyper-volume metric and spacing metric based on DE/rand/1.

Algorithm	Maximum $S(M,r)$	Minimum $S(M,r)$	Average $S(M,r)$	$SP(\cdot)$
HMOBBPSO	304.57	10.037	121.28	0.0868
MOLPSO	374.20	14.005	105.52	0.1242
MOLTPSO	267.07	14.543	102.36	0.1104
MOCPSO	371.37	5.7505	114.43	0.1898

TABLE 7. Statistical results of hyper-volume metric and spacing metric based on DE/rand/2.

Algorithm	Maximum $S(M,r)$	Minimum $S(M,r)$	Average $S(M,r)$	$SP(\cdot)$
HMOBBPSO	505.64	1.7656	148.44	0.1343
MOLPSO	254.68	7.2734	116.45	0.1551
MOLTPSO	377.47	19.919	137.72	0.0889
MOCPSO	456.30	30.346	140.34	0.2102

TABLE 8. Statistical results of hyper-volume metric and spacing metric based on DE/current-to-best/1.

Algorithm	Maximum $S(M,r)$	Minimum $S(M,r)$	Average $S(M,r)$	$SP(\cdot)$
HMOBBPSO	432.03	9.9002	123.69	0.0993
MOLPSO	378.29	12.504	105.81	0.1419
MOLTPSO	302.64	7.299	116.15	0.2326
MOCPSO	397.86	1.0671	104.37	0.1989

TABLE 9. C metric of different algorithms.

Algorithm	DE/rand/1	DE/rand/2	DE/current-to-best/1
HMOBBPSO	0.2	0.3	0.5
MOLPSO	0.2	0.2	0.1
HMOBBPSO	0.5	0.2	0.6
MOLTPSO	0.1	0.2	0
HMOBBPSO	0.4	0.1	0.6
MOCPSO	0.2	0.3	0.3

It can be seen from Tables 6 to 8 that the average hyper volume values of the proposed HMOBBPSO method are overall better than those of other methods. The proposed HMOBBPSO method with DE/rand/1 takes the best spacing metric 0.0868, which can be seen from the relative smooth and even paths in the Fig. 6(a). The different routes in Fig. 7(a) reveal that the proposed HMOBBPSO with DE/rand/2 produces the largest hyper volume values 505.64 in Table 7. The average hyper volume values of MOLPSO method with three kinds of mutation operations are approximate to each other, and there are the similar robot paths in the Fig. 6(b), Fig. 7(b) and Fig. 8(b).

For different algorithms with DE/rand/1 in Table 6, HMOBBPSO algorithm ranks the first place with the maximal average hyper volume value 121.28, and MOLPSO and MOLTPSO algorithms obtain the similar values, 105.52 and 102.36. MOLPSO and MOCPSO algorithms also gain

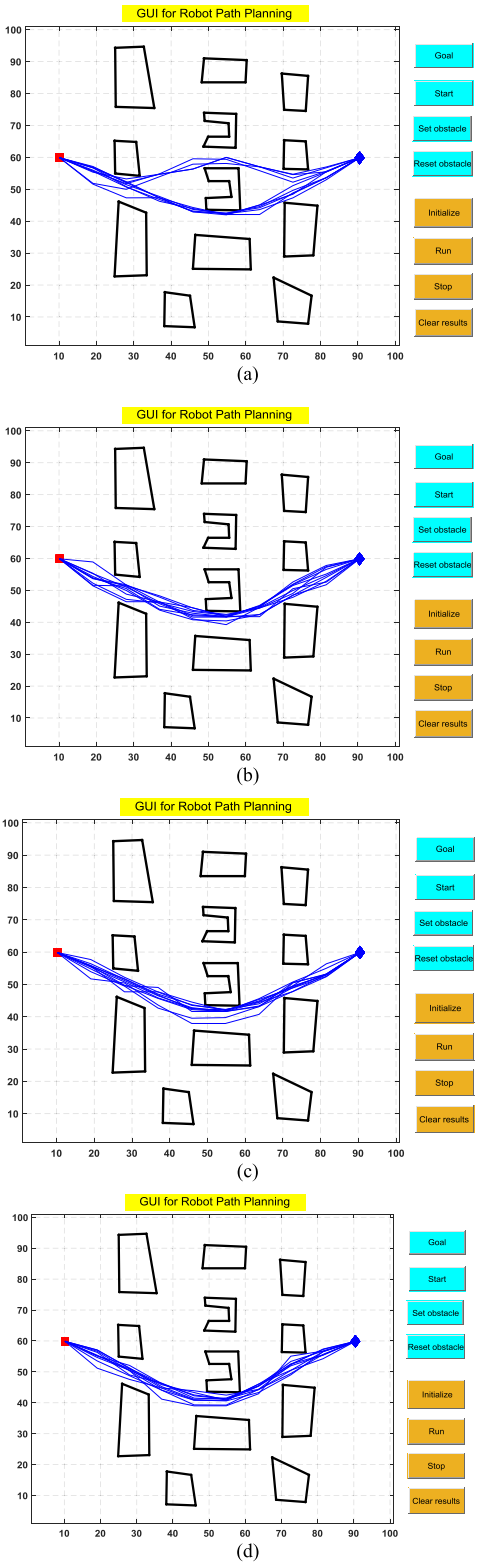

FIGURE 6. Optimal robot paths of different algorithms with DE/rand/1: (a) HMOBBPSO. (b) MOLPSO. (c) MOLTPSO. (d) MOCPSO.

the similar values, 374.20 and 371.37, for the maximal hyper volume metric. With respect to the spacing metric, HMOBBPSO and MOLTPSO algorithms obtain the first and

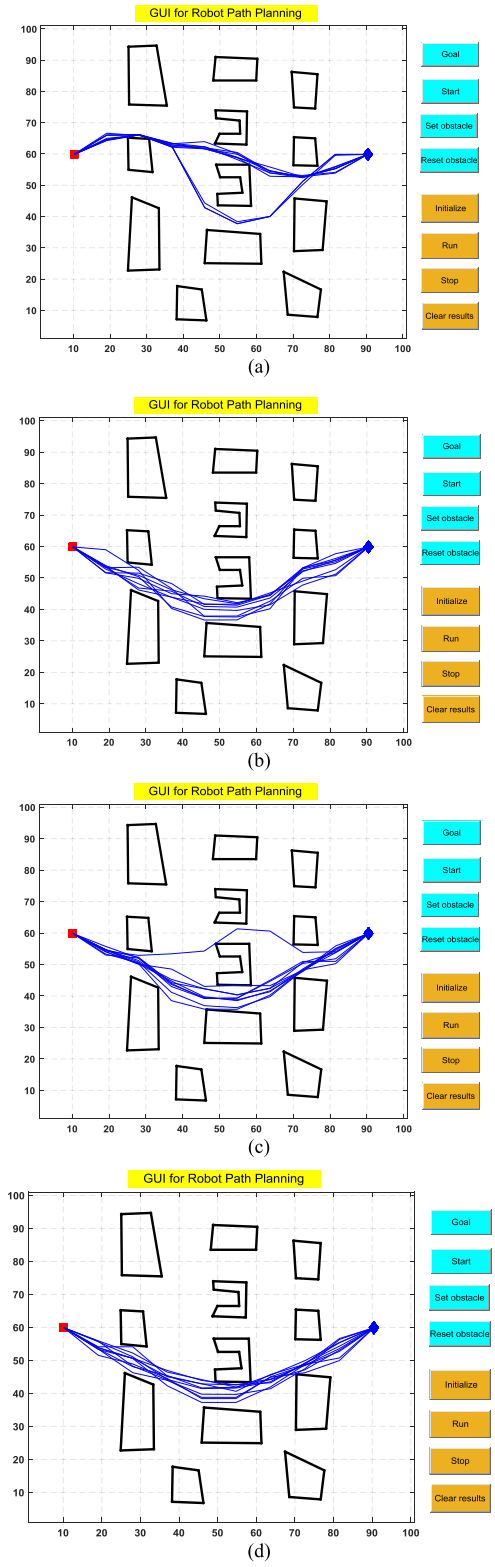

FIGURE 7. Optimal robot paths of different algorithms with DE/rand/2: (a) HMOBBPSO. (b) MOLPSO. (c) MOLTPSO. (d) MOCPSO.

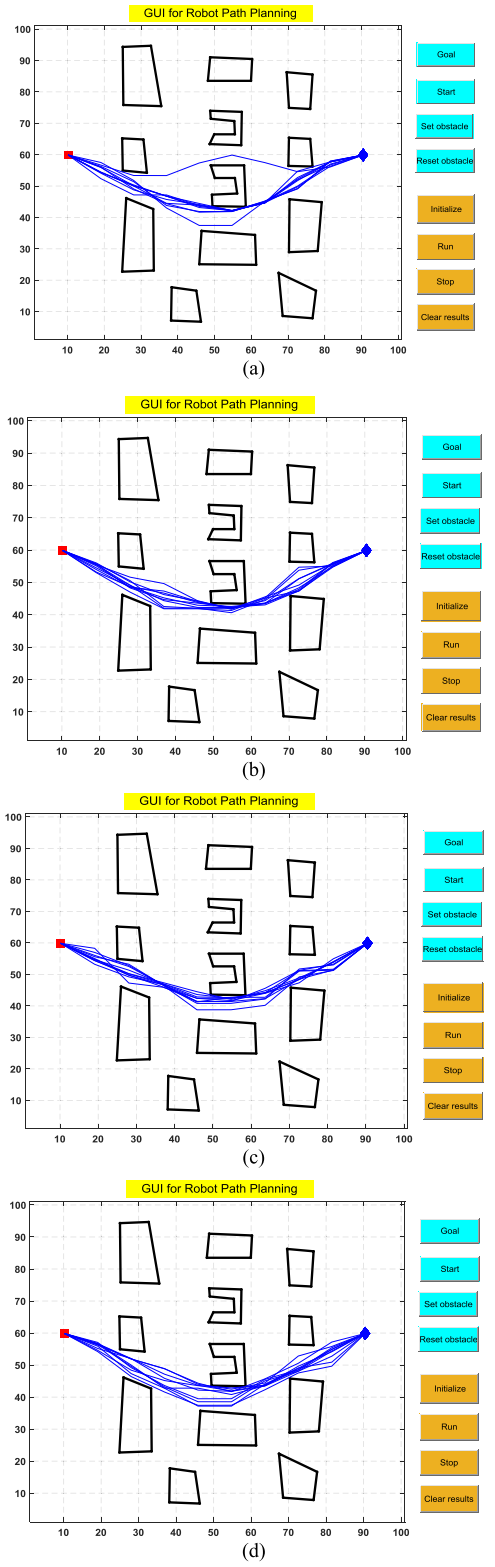

FIGURE 8. Optimal robot paths of different algorithms with DE/current-to-best/1: (a) HMOBBPSO. (b) MOLPSO. (c) MOLTPSO. (d) MOCPSO.

the second rank with the adjacent values, 0.0868 and 0.1104. The MOLPSO algorithm gains the third rank with the value 0.1242.

For different algorithms with DE/rand/2 in Table 7, HMOBBPSO algorithm achieves the best results with the maximal hyper volume value 505.64 and the maximal

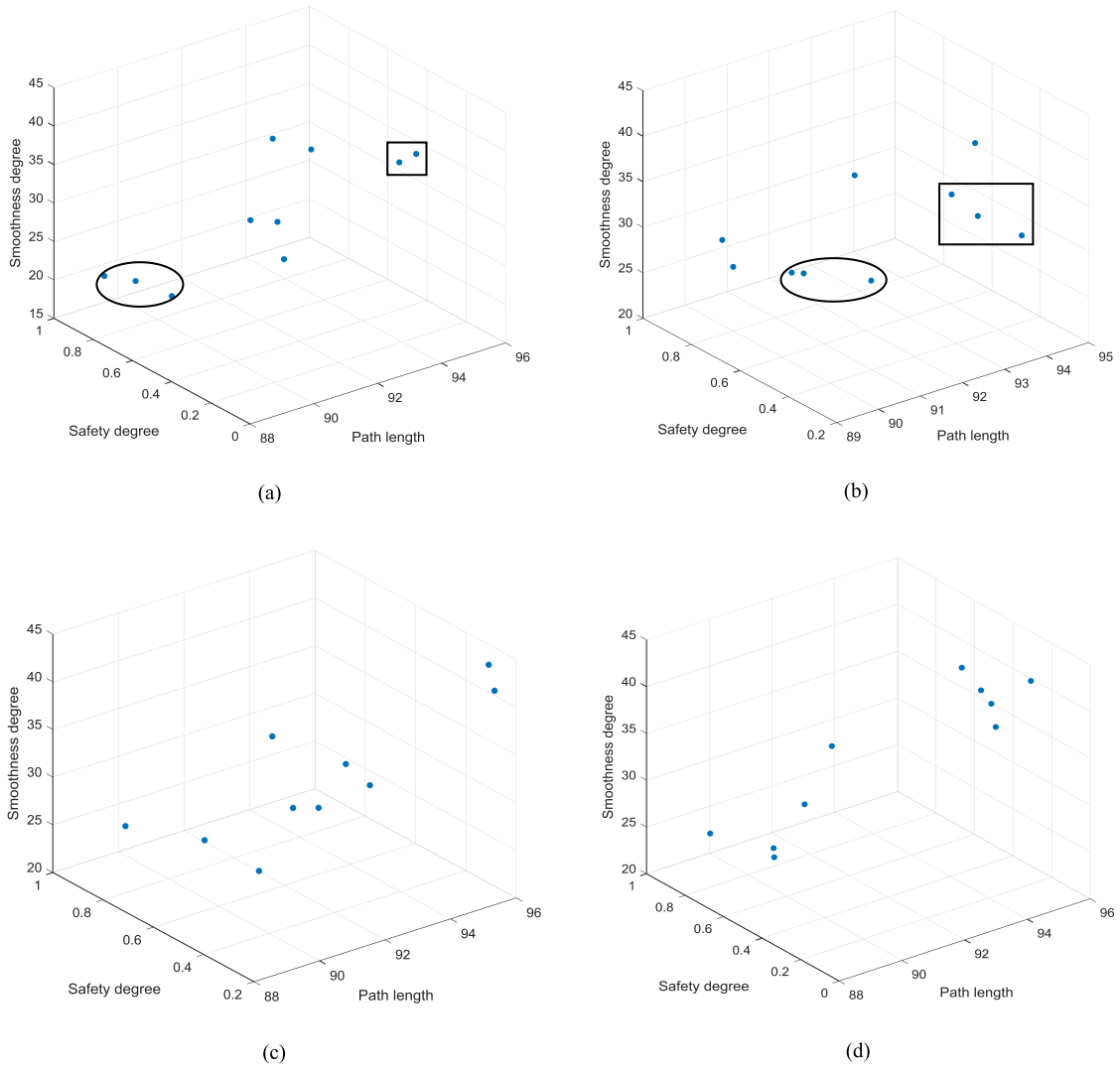

FIGURE 9. Pareto solutions of different algorithms with DE/rand/1: (a) HMOBBPSO. (b) MOLPSO. (c) MOLTPSO. (d) MOCPSO.

average hyper volume value 148.44. The MOCPSO and MOLTPSO algorithms rank the second and third places with the similar values, 140.34 and 137.72. About the spacing metric, The MOLTPSO algorithm gains the best value 0.0889, and the HMOBBPSO algorithm ranks the second place with the value 0.1343, which can be validated from the Fig. 7(a) and Fig. 7(c).

For different algorithms with DE/current-to-best/1 in Table 8, the HMOBBPSO and MOLTPSO algorithms acquire the first and second rank on average hyper volume metric with the values of 123.69 and 116.15. The HMOBBPSO method obtains the best maximal hyper volume value 432.03. With respect to the spacing metric, HMOBBPSO algorithm obtains the first rank with the minimal value 0.0993, which shows the distribution of the proposed algorithm is better than that of other methods. The MOLPSO algorithm gains the second rank with the value 0.1419.

The values of $C(\text{HMOBBPSO}, \text{MOLPSO}) = 0.5$, $C(\text{HMOBBPSO}, \text{MOLTPSO}) = 0.6$ and $C(\text{HMOBBPSO}, \text{MOCPSO}) = 0.6$ with DE/current-to-best/1 operation in Table 9 mean that the coverage of HMOBBPSO is superior to that of other three algorithms when DE/current-to-best/1 is incorporated into the robot path planning, which can be seen from the maximal hyper volume values 432.03, 378.29, 302.64 and 397.86 in Table 8. The values of C metric in column 1 in Table 9 show the solutions of HMOBBPSO algorithm are overall better than those of other three algorithms when DE/rand/1 is incorporated into the robot path planning. The values of C metric in row 2 show the HMOBBPSO algorithm can obtain better non-inferior solutions with different differential evolution operations than MOLPSO algorithm.

It can be observed from Fig. 6 to Fig. 8 that all the algorithms can skip over the trap obstacles and are efficient in finding multiple non-dominated robot paths, whereas the HMOBBPSO algorithm can seek more feasible paths than other three algorithms. The Fig. 6(a), Fig. 7(a) and Fig. 8(a) are all generate two piles of robot path, whereas Fig. 7(a) takes a devious routes and Fig. 6(a) and Fig. 8(a) gives

similar robot paths, which is coincide with the maximal hyper volume values 304.57 in table 6, 505.64 in Table 7 and 432.03 in Table 8. In particular, Fig. 7(c) also gains two piles of robot path, but there is only one path to be selected for the robot, which can be validated by the maximal hyper volume value 377.47 and the spacing metric 0.0889 in Table 7.

To demonstrate clearly the conflicts and variations among different considered objectives, Fig. 9 demonstrates the conflicting Pareto solutions from feasible archive taking different algorithms with DE/rand/1 as an example. It can be seen from Fig. 9(a) that the path length of the solutions in the ellipse is shorter than those in the rectangle, and the path smoothness degree of the solutions in the ellipse is also better than those in the rectangle, whereas the path safety degree of the solutions in the ellipse is worse than those in the rectangle. Similarly, it can be seen from Fig. 9(b) that the path length of the solutions in the rectangle is longer than those in the ellipse, and the path smoothness degree of the solutions in the rectangle is worse than those in the ellipse, whereas the path safety degree of the solutions in the rectangle is superior to those in the ellipse. The distribution of the solutions in Fig. 9(a) is better than the one of the solutions in Fig. 9(b), 9(c) and 9(d), which can be validated from the spacing metric values in Table 6.

VI. CONCLUSIONS

In this paper, a hybrid multi-objective bare bones particle swarm optimization combined with differential evolution is presented to solve the path planning of mobile robot. Since there are three optimization indices in the proposed algorithm, i.e., the path length, and the smoothness degree and the safety degree of a path, the path planning of mobile robot is consequently converted into a multi-objective optimization problem with constraints. To evaluate the fitness of a particle and select the personal best position of a particle, a new Pareto domination with collision constraints is developed according to the definition of the collision degree of a path. To improve the feasibility of an infeasible path, differential evolution is adopted to promote the mutation of infeasible paths blocked by the obstacles with the difference vector extracted from the feasible archive. Finally, the proposed algorithm is tested by two simulation experiments, and the simulation results reveal that the proposed algorithm outperforms other algorithms in terms of the three metrics.

However, there are still other issues to be studied further. Firstly, the division number of the segment between the start point the target point is fixed in our method. How to design a strategy to adaptively adjust the division number is worth to study further. Secondly, if there are some dynamic obstacles in the workspace, how to develop our algorithm to avoid the dynamic characteristics is also another topic in the future. Finally, we will further compare our methods with other classic path planning methods under some complex environments, even more in the three-dimension scenarios.

REFERENCES

[1] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Robot. Auton. Syst.*, vol. 86, pp. 13–28, Dec. 2016.
[2] D.-W. Gong, J.-H. Zhang, and Y. Zhang, "Multi-objective particle swarm optimization for robot path planning in environment with danger sources," *J. Comput.*, vol. 6, no. 8, pp. 1554–1561, 2011.
[3] P. K. Das, H. S. Behera, S. Das, H. K. Tripathy, B. K. Panigrahi, and S. K. Pradhan, "A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment," *Neurocomputing*, vol. 207, pp. 735–753, Sep. 2016.
[4] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
[5] E. G. Tsardoulias, A. Iliakopoulou, A. Kargakos, and L. Petrou, "A review of global path planning methods for occupancy grid maps regardless of obstacle density," *J. Intell. Robot. Syst.*, vol. 84, nos. 1–4, pp. 829–858, 2016.
[6] M. Duguleana and G. Mogan, "Neural networks based reinforcement learning for mobile robots obstacle avoidance," *Expert Syst. Appl.*, vol. 62, pp. 104–115, Nov. 2016.
[7] S. Ghosh, P. P. Kumar, and D. R. Parhi, "Performance comparison of novel WNN approach with RBFNN in navigation of autonomous mobile robotic agent," *Serbian J. Electr. Eng.*, vol. 13, no. 2, pp. 239–263, 2016.
[8] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," *J. Comput. Sci.*, vol. 25, pp. 339–350, Mar. 2018.
[9] A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, and B. Bouzouia, "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control," *Robot. Auton. Syst.*, vol. 89, pp. 95–109, Mar. 2017.
[10] X. Zhang, Y. Zhao, N. Deng, and K. Guo, "Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 3, p. 91, 2016.
[11] J. Liu, J. Yang, H. Liu, X. Tian, and M. Gao, "An improved ant colony algorithm for robot path planning," *Soft Comput.*, vol. 21, no. 19, pp. 5829–5839, 2017.
[12] M.-R. Zeng, L. Xi, and A.-M. Xiao, "The free step length ant colony algorithm in mobile robot path planning," *Adv. Robot.*, vol. 30, no. 23, pp. 1509–1514, 2016.
[13] C.-T. Yen and M.-F. Cheng, "A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance," *Microsyst. Technol.*, vol. 24, no. 1, pp. 125–135, 2016.
[14] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," *Soft Comput.*, vol. 21, no. 4, pp. 949–964, 2017.
[15] C. Liu, Y. Zhao, F. Gao, and L. Liu, "Three-dimensional path planning method for autonomous underwater vehicle based on modified firefly algorithm," *Math. Problems Eng.*, vol. 2015, Dec. 2015, Art. no. 561394.
[16] M. A. Hossain and I. Ferdous, "Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique," *Robot. Auton. Syst.*, vol. 64, pp. 137–141, Feb. 2015.
[17] X.-C. Pu, H.-Q. Zhao, and Y. Zhang, "Mobile robot path planning research based on bacterial chemotaxis," *CAAI Trans. Intell. Syst.*, vol. 9, no. 1, pp. 69–75, 2014. [Online]. Available: http://tis.hrbeu.edu.cn/oa/DArticle.aspx?type=view&id=20140110
[18] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5177–5191, 2015.
[19] Y. Zhang, D.-W. Gong, and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, Mar. 2013.
[20] B. Tang, Z. Zhanxia, and J. Luo, "A convergence-guaranteed particle swarm optimization method for mobile robot global path planning," *Assem. Automat.*, vol. 37, no. 1, pp. 114–129, 2017.
[21] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization," *Appl. Soft Comput.*, vol. 59, pp. 68–76, Oct. 2017.
[22] B. Song, Z. Wang, L. Zou, L. Xu, and F. E. Alsaadi, "A new approach to smooth global path planning of mobile robots with kinematic constraints," *Int. J. Mach. Learn. Cybern.*, 2017, doi: 10.1007/s13042-017-0703-7.

[23] P. K. Das, H. S. Behera, and B. K. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning," *Swarm Evol. Comput.*, vol. 28, pp. 14–28, Jun. 2016.
[24] K. Su, Y. Wang, and X. Hu, "Robot path planning based on random coding particle swarm optimization," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 4, pp. 58–64, 2015.
[25] N. Setyawan, R. E. A. Kadir, and A. Jazidie, "Adaptive Gaussian parameter particle swarm optimization and its implementation in mobile robot path planning," in *Proc. Int. Seminar Intell. Technol. Appl. (ISITIA)*, Surabaya, Indonesia, Aug. 2017, pp. 238–243.
[26] H. Liu, G. Ding, and B. Wang, "Bare-bones particle swarm optimization with disruption operator," *Appl. Math. Comput.*, vol. 238, pp. 106–122, Jul. 2014.
[27] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Mar. 2002.
[28] M. Davoodi, F. Panahi, A. Mohades, and S. N. Hashemi, "Clear and smooth path planning," *Appl. Soft Comput.*, vol. 32, pp. 568–579, Jul. 2015.
[29] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Inf. Sci.*, vol. 423, pp. 172–186, Jan. 2018.
[30] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016.
[31] A. P. Piotrowski, "Review of differential evolution population size," *Swarm Evol. Comput.*, vol. 32, pp. 1–24, Feb. 2017.
[32] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, "Feature selection algorithm based on bare bones particle swarm optimization," *Neurocomputing*, vol. 148, pp. 150–157, Jan. 2015.
[33] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.
[34] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput., IEEE World Congr. Comput. Intell.*, May 1998, pp. 69–73.
[35] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, Jun. 2004.
[36] N. Riquelme, C. Von Lücken, and B. Baran, "Performance metrics in multi-objective optimization," in *Proc. Latin Amer. Comput. Conf. (CLEI)*, Arequipa, Peru, Oct. 2015, pp. 1–11.

JIAN-HUA ZHANG received the B.S. degree in electrical engineering and automation, and the M.S. and Ph.D. degrees in control theory and control engineering from the China University of Mining and Technology in 2003, 2006, and 2014, respectively. He is currently an Associate Professor with School of Mechanical and Electrical Engineering, Xuzhou Institute of Technology. His research interests include robot path planning and odor source localization.

YONG ZHANG received the M.S. and Ph.D. degrees in control theory and control engineering from the China University of Mining and Technology in 2006 and 2009, respectively. He is currently a Professor with the School of Information and Control Engineering, China University of Mining and Technology. His research interests include intelligence optimization, pattern recognition, and odor source localization.

YONG ZHOU received the M.S. and Ph.D. degrees in control theory and control engineering from the China University of Mining and Technology in 2003 and 2006, respectively. He is currently a Professor with the School of Computer Science and Technology, China University of Mining and Technology. His research interests include machine learning, intelligence optimization, and data mining.