

Received July 10, 2021, accepted July 15, 2021, date of publication July 20, 2021, date of current version July 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098706

Mobile Robot Dynamic Path Planning Based on Self-Adaptive Harmony Search Algorithm and Morphin Algorithm

YONGBIN QUAN¹, HAIBIN OUYANG¹, CHUNLIANG ZHANG¹, STEVEN LI², AND LI-QUN GAO³

¹School of Mechanical and Electric Engineering, Guangzhou University, Guangzhou 510006, China

²Graduate School of Business and Law, RMIT University, Melbourne, VIC 3000, Australia

³College of Information Science and Engineering, Northeastern University, Shenyang 110004, China

Corresponding authors: Haibin Ouyang (oyhb1987@163.com) and Chunliang Zhang (nhzcl@163.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 61806058, in part by the Natural Science Foundation of Guangdong Province under Grant 2018A030310063, and in part by the Guangzhou Science and Technology Plan Project under Grant 201804010299.

ABSTRACT As a vital part of autonomous navigation of mobile robot, path planning is a hot research direction which aims at searching a shortest collision-free path from the starting position to the goal position in a complex environment. In this paper, a method for global dynamic path planning is designed based on improved self-adaptive harmony search algorithm (ISAHS) and Morphin algorithm. Firstly, to improve the quality of new solution vector, a neighbors and optimal learning strategy is introduced. Secondly, two key parameters are adjusted adaptively and a probability disturbance strategy is designed for renewing harmony memory, and then an improved self-adaptive harmony search algorithm is proposed to obtain an initial optimal path in the static environment. Finally, the Morphin algorithm is introduced to avoid the moving obstacles in real time. Simulation results indicate that the proposed method performs well in planning an initial static optimal path and it can avoid all preset moving obstacles effectively.

INDEX TERMS Dynamic path planning, improved self-adaptive harmony search algorithm, morphin algorithm, mobile robot.

I. INTRODUCTION

Mobile robot is one of the most intelligent devices which can replace humans to do repetitive and dangerous jobs [1], [2]. Path planning, an important part of autonomous navigation, is aimed at searching a shortest collision-free path from the starting position to the goal position in a complex environment. There are four kinds of path planning, i.e., global path planning, local path planning, static path planning and dynamic path planning [3], [4].

In the past few decades, scholars have studied path planning deeply and proposed a series of classical path planning methods, such as artificial potential field method (APF) [5], grid method [6], simulated annealing algorithm (SA) [7], A* algorithm [8], rapidly extended random tree algorithm (RRT) [9], Morphin algorithm [10], etc. APF believes that robot will move according to a resultant force formed by

the attraction of goal position and the repulsion of obstacles. APF's principle is simple, but it is not easy to escape the local trap in an environment with concave edge obstacles. [11], Chen *et al.* solved the APF's problem mentioned above by changing the calculation method of repulsion force and adding virtual target points. SA simulates solid's cooling process, and introduces probability disturbance strategy to jump out of local trap to obtain a global optimum. [7], Xiao *et al.* used a simulated annealing algorithm to find the near-optimized paths, which can achieve balanced task assignment for UAV formations. [12], Zhuge *et al.* have overcome the inflexible trajectory of traditional Morphin algorithm and obtain a smooth path by constructing a multi-layer Morphin search tree. In recent years, some new intelligence optimization algorithms have performed excellently in solving path planning problem such as genetic algorithm (GA) [13], [14], particle swarm optimization algorithm (PSO) [15], differential evolutionary algorithm (DE) [16] and firework algorithm (FWA) [17]. [18], Li and Chou

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li¹.

designed a self-adaptive learning particle swarm optimization algorithm (SLPSO) which selects the most appropriate search strategy adaptively in different stages and limits particle velocity and position according to the boundary violation strategy. DE algorithm is very similar to GA algorithm, which includes heredity, crossover and mutation. [19], Fan and Zhang planned an optimal collision-free path by amending the infeasible solution generated by DE's crossover operation.

Harmony search algorithm (HS) is an intelligent optimization algorithm, which is proposed by Geem *et al.* [27]. HS algorithm based on memory consideration, pitch adjusting and stochastic selection operation to explore new and better solution in the search process. Since the proposed of HS algorithm, many improvements and applications of HS algorithm have been presented in many engineering fields. HS algorithm attracts much attention because of its unique optimizing means and extensive application prospects [28], [29]. HS algorithm performs better than the other swarm intelligent optimization algorithms in many fields, but it also suffers from some drawbacks potentially and its optimization efficiency is restricted. For example, the basic HS can be easily trapped and lead to a local optimum solution. It is quite sensitive to the value of a few key control parameters like other meta-heuristic algorithms. In order to overcome this weakness, some variants of HS are proposed and applied to many optimization problems [20]. To enhance the global search capability, the best harmony solution has been employed in the HS algorithm pitch adjustment step. Inspired by swarm intelligent, a novel global harmony search algorithm (NGHS) based on position updating and genetic mutation was proposed by Zou *et al.* [24]. NGHS algorithm has high optimization efficiency and strong convergence performance. It is used to solve task assignment problem [31], 0–1 knapsack problem [32] and reliability problem [33]. Ouyang *et al.* offer a strategy for the adjustment of BW and design a modified harmony search algorithm (MHS) [30]. In 2017, an improved HS algorithm called LHS have been developed to improve solution precision and enhance the ability of escaping local optima [34]. Ouyang *et al.* proposed an amended harmony search algorithm with perturbation strategy for large-scale system reliability problems in 2018 [35]. In recent years, HS has been gradually applied to robot path planning due to its simple concept, fewer parameters, and easier implement. [21], Combined with the Pythagorean velocity curve PH, a modified harmony search algorithm (MHS) with crossover mutation strategy is designed by Wu and Yi to solve UAV path planning problem in the urban and mountainous environment. [22], Kundu and Parhi introduced a dynamically adaptive harmony search algorithm (DAHS) which has improved the search capability and random generalization ability by adjusting important parameters adaptively and designing a new search scheme. Finally, they planned a collision-free path for underwater robot in a three-dimensional scene. [23], an improved evolutionary algorithm based on GA and HS is designed

to solve the path planning problem of multi-UAV through different evolutionary factors.

Harmony search algorithm has unique search mechanism, simple principle and strong expansibility, and a better optimization result often can be obtained after improvement. Therefore, in this paper, the harmony search algorithm is introduced to solve the static path planning problem. Moreover, compared with APF method, the Morphin algorithm does not have the disadvantage of falling into local traps because it selects the most appropriate path among multiple predicted arc paths, so this article will use it to avoid the moving obstacles in real time.

The main contributions of this paper are as follows:

(i) a global dynamic path planning method based on improved self-adaptive harmony search algorithm (ISAHS) and Morphin algorithm under dynamic environment is proposed. Firstly, the path length and the path collision degree are considered as two optimization objectives, and then an evaluation function is built.

(ii) an improved self-adaptive harmony search algorithm was designed to get an initial optimal path in the static environment by introducing the neighbors and optimal learning strategy, self-adaptive strategy and probability disturbance updating strategy.

(iii) a global dynamic path planning method based on the initial static optimal path and Morphin algorithm is proposed to avoid the moving obstacles in real time. Our simulation results reveal that the proposed algorithms perform better.

The remainder of the paper is organized as follows. The path planning problem and related algorithms are explained in Section 2. Section 3 introduces an improved self-adaptive harmony search algorithm (ISAHS) in detail. Many simulation experiments to investigate the effectiveness of the proposed algorithms are presented in Section 4. Section 5 concludes this paper.

II. PATH PLANNING PROBLEM AND RELATED ALGORITHMS

This section introduces a mathematical model and a corresponding evaluation function for path planning problem, and then explains the implementation process of harmony search algorithm and the basic idea of Morphin algorithm.

A. PATH PLANNING PROBLEM

1) MATHEMATICAL MODEL

A reasonable mathematical model is helpful to plan a shortest collision-free path. This paper establishes a mathematical model for path planning based on the existing method presented in [18].

Fig. 1 is a static environment for mobile robot which includes four obstacles (O), starting position (S) and goal position (G). The progress of establishing mathematical model is divided into three steps. Firstly, a relative coordinate system x - y is built when the line between the starting position and the goal position is considered as a horizontal axis.

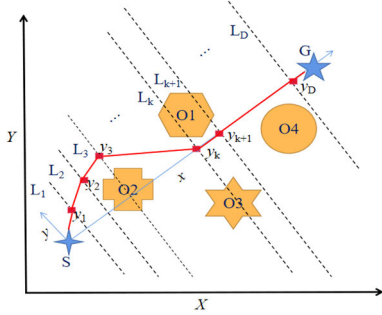


FIGURE 1. The static environment for mobile robot.

Secondly, divide the line into $D+1$ equal part with D points and then draw dotted lines perpendicular to x axis. Finally, the points on the dotted lines are considered as key points and a path will be expressed by the coordinates of key points on y axis as

$$Path = [y_S, y_1, y_2, \dots, y_k, \dots, y_D, y_G] \quad (1)$$

Sometimes, the path in x - y coordinate system needs to be transformed to X - Y coordinate system. The transformation relationship is defined as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} X_S \\ Y_S \end{bmatrix} \quad (2)$$

where, X_S and Y_S are the coordinates of starting point in X - Y coordinate system; θ is a vector of angular differences between X and x axis.

2) EVALUATION FUNCTION

The path evaluation criteria include low energy consumption, no collision, smooth, etc., among which no collision is the minimum requirement for path planning. In this paper, low energy consumption and no collision are taken as two optimization objectives.

The first target is low energy consumption, which can be measured by the path length. Naturally, the path length is hoped to be as small as possible. Assuming that the starting position is p_0 and the goal position is p_{D+1} , the path length (L) is calculated as

$$L(Path) = \sum_{i=0}^D \|p_{i+1} - p_i\| \quad (3)$$

where, $\|p_{i+1} - p_i\|$ is the distance from point p_i to p_{i+1} ; D is the number of key points.

The second target is no collision, which can be measured by the path collision risk degree. The path collision risk degree (C) is defined as

$$C(Path) = \sum_{i=0}^D \sum_{j=1}^H S(l_i, O_j) \cdot \|p_{D+1} - p_0\| \quad (4)$$

$$S(l_i, O_j) = \begin{cases} 1, & \text{collision} \\ 0, & \text{no collision} \end{cases} \quad (5)$$

where, H is the number of obstacles; $S(l_i, O_j)$ will value at 1 to add a penalty if the i^{th} segment of path crashes with the j^{th} obstacle.

In fact, it is difficult to meet the above two targets simultaneously. For example, a shorter path may increase the collision risk degree, and a fewer collision risk degree may lead to a longer path. This paper combines these two goals as

$$f(Path) = v_1 \cdot L(Path) + v_2 \cdot C(Path) \quad (6)$$

where, f is the evaluation function for path; v_1 and v_2 are weight factors, which are set to 0.9 and 0.1 respectively in Part IV.B

B. HARMONY SEARCH ALGORITHM

Harmony search algorithm is an optimization algorithm imitating the music improvising process, which was explained in detail in [27]. In this paper, a brief overview of HS algorithm is as follows:

Step 1 Initialize related parameters.

The parameters include variable dimension (D), variable range $[y^L, y^U]$, harmony memory size (HMS), harmony memory considering rate ($HMCR$), pitch adjustment rate (PAR), pitch adjusting bandwidth (BW), maximum iterations (T).

Step 2 Initialize harmony memory.

The harmony memory (HM) is composed of many solution vectors with D dimensions which are generated from the variable range $[y^L, y^U]$ randomly. HM is defined as

$$HM = \begin{bmatrix} y_1^1 & y_2^1 & \dots & y_D^1 & f(y^1) \\ y_1^2 & y_2^2 & \dots & y_D^2 & f(y^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ y_1^{HMS} & y_2^{HMS} & \dots & y_D^{HMS} & f(y^{HMS}) \end{bmatrix} \quad (7)$$

Step 3 Improvise a new solution vector for HS.

Improvising a new solution vector is the most important step to realize harmony search algorithm, which is usually related to $HMCR$, PAR , and BW . The detailed pseudo-code for improvising a new solution vector is given in Algorithm 1 below.

Step 4 Renew harmony memory.

The worst solution vector in harmony memory will be replaced by the new solution vector whose quality is better.

Step 5 End search or not.

Continue to execute step 3 and 4 when the current iterations are less than the maximum iterations.

C. MORPHIN ALGORITHM

Morphin algorithm is a local dynamic obstacle avoidance algorithm, which is often applied in dynamic path planning. The main idea of Morphin algorithm is creating a set of predicted arc paths according to the different steering angles of mobile robot, and then the most suitable path will be chosen. Morphin algorithm mainly includes two aspects, which are the generation of predicted arc paths and the selection of predicted arc paths.

Algorithm 1 Harmony Search Algorithm

Input: the problem to be solved
Output: the best solution vector of the problem to be solved

Step 1: initialize related parameters, including *HMCR*, *PAR*, *BW* etc.;

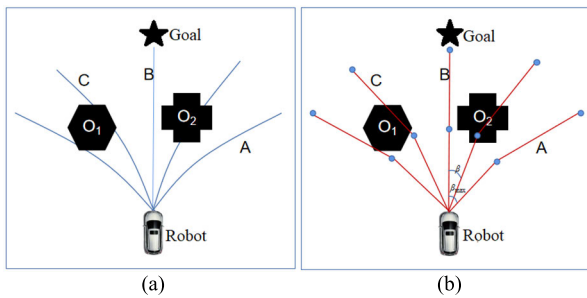
Step 2: initialize harmony memory *HM*;

Step 3: improvise a new solution vector;

```

for  $k = 1$  to  $T$  do
  for  $i = 1$  to  $D$  do
    if  $\text{rand}(1) < \text{HMCR}$  do
       $y_i^{\text{new}} = y_i^a$ ; where  $a \in 1, 2, \dots, \text{HMS}$ 
      if  $\text{rand}(1) < \text{PAR}$  do
         $y_i^{\text{new}} = y_i^a \pm \text{rand}(1) \cdot \text{BW}$ ;
      end
    else
       $y_i^{\text{new}} = y^L + \text{rand}(1) \cdot (y^U - y^L)$ ;
    end
  end
  renew HM;
end
Step 4: return the best solution in HM.

```

**FIGURE 2.** The predicted arc paths.

The generation of predicted arc paths depends on the motion equation of mobile robot. The predicted arc paths are essentially number of arc paths with different steering angles, as shown in Fig. 2(a). In the simulation experiment, the predicted arc paths are often linearized as shown in Fig. 2(b).

The selection of predicted arc paths involves a new evaluation function [38], which is defined as

$$f'(arc) = \begin{cases} \infty & \text{collision} \\ \gamma_1 \cdot (1/G) + \gamma_2 \cdot \Delta d + \gamma_3 \cdot \beta & \text{nocollision} \end{cases} \quad (8)$$

where, f' is the evaluation function of predicted arc paths; G is the minimum distance from the predicted arc path to obstacles; Δd is the distance between the end point of predicted arc path and the goal point; β is the mobile robot's steering angle; γ_1 , γ_2 , and γ_3 are the weight factors, which are 0.98, 0.02, and 0 in Part IV.C.

III. IMPROVED SELF-ADAPTIVE HARMONY SEARCH ALGORITHM

In this section, to obtain an initial optimal path in the static environment, an improved self-adaptive harmony search

algorithm (ISAHS) is designed. Moreover, the Morphing algorithm is introduced to avoid the moving obstacles in real time and then the global dynamic path planning steps are given.

A. ISAHS ALGORITHM

In this paper, an improved self-adaptive harmony search algorithm (ISAHS) is designed to obtain an initial optimal path in the static environment. Compared with traditional harmony search algorithm, ISAHS has been improved in three parts.

1) NEIGHBORS AND OPTIMAL LEARNING STRATEGY

The new solution vector of HS algorithm has three parts, i.e., harmony memory consideration, pitch adjustment and random mutation, which are controlled by *HMCR*, *PAR* and *BW*. Among them, the pitch adjustment part is to search a solution vector near the harmony memory, which belongs to the local independent adjustment. The random mutation part is to search a solution vector in the variable range $[y^L, y^U]$, which belongs to the global independent adjustment. To enhance the learning ability of ISAHS, this paper introduces a neighbors learning and optimal learning strategy which will improve the quality of improvising new solution vector. The pseudo-code is as follow:

```

for  $i = 1$  to  $D$  do
  if  $\text{rand}(1) < \text{HMCR}$  do
     $y_i^{\text{new}} = y_i^{\text{worst}} + c_1 \cdot \text{rand}(1) \cdot (y_i^{\text{best}} - y_i^{\text{worst}})$ ;
    if  $\text{rand}(1) < \text{PAR}$  do
       $y_i^{\text{new}} = y_i^{\text{worst}} + c_2 \cdot \text{rand}(1) \cdot (y_i^a - y_i^{\text{worst}})$ ;
    end
  else
     $y_i^{\text{new}} = y^L + \text{rand}(1) \cdot (y^U - y^L)$ ;
  end
end

```

where, a is a random number from $1, 2, \dots, \text{HMS}$; y_i^{worst} represents the worst solution vector in *HM*; y_i^{best} represents the best solution vector in *HM*; c_1 and c_2 are optimal learning factor and neighbors learning factor respectively, which are recommended to value at range [1, 2].

The new solution vector of ISAHS algorithm also has three parts, i.e., optimal learning, neighbors learning and random mutation. When a new solution vector is mainly learning from the optimal solution, it is conducive to the fast convergence. When a new solution vector is mainly learning from neighbors, it is beneficial to avoid falling into the local extremes early in the iteration. When a new solution vector is mainly derived from the random mutation, it is beneficial to enrich the variety of harmony memory and avoid getting into the local minimum.

2) SELF-ADAPTIVE STRATEGY

To reduce the difficulty of setting parameters and improve ISAHS's performance, a self-adaptive strategy of setting *HMCR* and *PAR* is designed in this part.

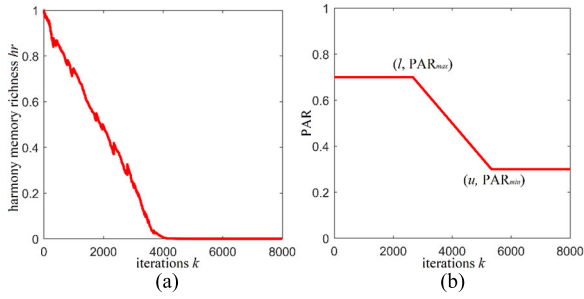


FIGURE 3. Variation of hr and PAR .

First, this paper introduced a new concept called harmony memory richness (hr), which is related to the relative closeness of evaluation function of all solution vectors in HM . The harmony memory richness (hr) is defined as

$$hr(k) = \begin{cases} \frac{\ln\{1 + [f_{avg}(k) - f_{min}(k)]\}}{\ln[1 + (f_{avg0} - f_{min0})]}, & hr < 1 \\ 1, & hr \geq 1 \end{cases} \quad (9)$$

where, k is the current iteration number; $hr(k)$ is the richness of the k^{th} harmony memory; $f_{avg}(k)$ and $f_{min}(k)$ are the average value and the minimum value of all solution vectors in the k^{th} harmony memory respectively; f_{avg0} and f_{min0} are the average value and the minimum value in the first harmony memory respectively.

With the iteration going on, the evaluation function values of all solution vectors in HM will gradually approach, which will lead to the average value closing to the minimum value. Therefore, the harmony memory richness (hr) will gradually tend to zero, which implies that hr is declining. Fig. 3(a) is a variation diagram of harmony memory richness in solving Sphere problem.

Second, a self-adaptive scheme of $HMCR$ based on harmony memory richness is designed as

$$HMCR(k) = HMCR_{max} - hr(k) \cdot (HMCR_{max} - HMCR_{min}) \quad (10)$$

where, $HMCR_{max}$ and $HMCR_{min}$ are maximum and minimum considering rate respectively, which are usually taken at range [0.9, 1] according to [27]. At the beginning of iteration, the new solution vector is mainly composed of random mutation due to the smaller $HMCR$, which is beneficial to the richness of harmony memory and global search ability. In the later period of iteration, the new solution vector is mainly from optimal learning and neighbors learning due to the higher $HMCR$, which is conducive to the fast convergence.

Finally, this paper designs a real-time variation scheme for PAR . To avoid premature convergence caused by learning too much from the optima, a larger PAR should be taken in early iteration. In the middle and late iteration, the optimal learning ability of ISAHS should be strengthened by controlling PAR decline to reduce searching time and improve searching

accuracy. Therefore, the PAR in this paper is defined as

$$PAR(k) = \begin{cases} PAR_{max}, & k < l \\ MI(k), & l \leq k \leq u \\ PAR_{min}, & k > u \end{cases} \quad (11)$$

$$MI(k) = \frac{(k - l)}{(u - l)} \cdot (PAR_{max} - PAR_{min}) \quad (12)$$

where, PAR_{max} and PAR_{min} are the maximum and minimum pitch adjustment rate respectively, which are valued at range [0, 1]; l is a boundary number distinguishing between the middle iteration and the early iteration, which is set as $T/3$ in this paper. u is also a boundary number distinguishing between the middle iteration and the late iteration, which is $2T/3$ in this paper.

The variation diagram of PAR is given in Fig. 3(b).

3) PROBABILITY DISTURBANCE UPDATING STRATEGY

To avoid the rapid convergence to local optimum, this paper accepts a new solution with a certain probability based on the simulated annealing algorithm (SA). In SA, a new solution is accepted as a current solution when the new solution is superior to the past solution, which is conducive to rapid convergence. A new solution is accepted as a current solution according to a certain probability when the new solution is poorer than the old solution, which is conducive to exclude the local extremum. Therefore, this paper updates harmony memory as follow:

Step 1 replace the worst solution vector in HM directly with a new harmony vector whose quality is better;

Step 2 replace the worst solution vector in HM with a new harmony vector according to a certain probability (p) if the new solution vector is worse. The calculation formula of probability (p) is as

$$p(k) = \begin{cases} 1, & f(y^{new}) < f(y^{worst}) \\ p_{max} \cdot \lambda^k, & f(y^{new}) \geq f(y^{worst}) \end{cases} \quad (13)$$

where, p_{max} is the maximum probability of accepting the new solution vector with worse quality, which is set as 1 in this paper; λ is an attenuation coefficient less than 1, which is given as 0.99 in this paper. In the early stages of iteration, a larger p is beneficial to exclude the local extremum, when a smaller p is conducive to the fast convergence and fine search at the end of iteration.

B. GLOBAL DYNAMIC PATH PLANNING STEPS

In this part, a global dynamic path planning method is designed based on the proposed ISAHS algorithm and Morphing algorithm. First, we search an initial optimal path in the static environment according to ISAHS and then the Morphing algorithm is used to avoid the moving obstacles in real time. The specific implementation steps are as follows:

Step 1 Search an initial optimal path in the static environment.

Firstly, build a mathematical model for path planning problem according to Section 2 above and then plan an

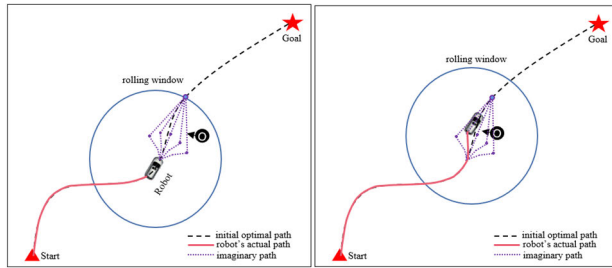


FIGURE 4. Diagram of local dynamic path planning.

initial optimal path under the static environment based on the proposed ISAHS algorithm.

Step 2 Suppose that the mobile robot will run on the initial optimal path and then introduce a rolling window.

The rolling window will follow the movement of mobile robot and roll forward. The radius of rolling window is related to the sensor's sensing distance, which is required to be greater than the minimum collision distance (d_c).

Step 3 Judge whether there are dynamic obstacles in the rolling window.

Record the position $[x_{ob}(t), y_{ob}(t)]$, speed $v_{ob}(t)$ and direction $\theta_{ob}(t)$ of dynamic obstacles and mark the mobile robot as early-warning state if there are dynamic obstacles in the rolling window. Otherwise, judge whether the mobile robot has reached the goal position. If so, end path planning; If not, return to Step 2.

Step 4 Judge whether the local obstacle avoidance is required.

Mark the mobile robot as obstacle-avoidance state when the early-warning state robot is predicted to crash with the dynamic obstacles. Otherwise, judge whether the mobile robot has reached the goal position. If so, end path planning; If not, return to Step 2. The mobile robot will be predicted to crash with dynamic obstacles if the distance between themselves is less than the minimum collision distance (d_c) according to an assumption that the mobile robot and dynamic obstacles will continue to move forward at a constant speed and direction.

Step 5 Local dynamic path planning in the rolling window based on Morphin algorithm.

In this step, the Morphin algorithm given in Section 2 is used for local dynamic obstacle avoidance. The predicted arc path obtained by Morphin algorithm will consider as an imaginary path where the mobile robot will move. The local dynamic path planning diagram is given in Fig. 4.

Step 6 End search or not.

If the mobile robot does not reach the goal position, return to Step 2. Conversely, end the program.

IV. SIMULATION RESULTS AND ANALYSIS

To verify the performance of ISAHS algorithm and the effectiveness of global dynamic path planning method, a series of simulation experiments have been down in this part. The simulation software is MATLAB R2016a, and

the computer is configured with CPU 2.20Ghz and RAM 4.00GB. In addition, all problems to be solved in this paper are the minimum optimization problems

A. BENCHMARK FUCTION TEST

To evaluate the performance of ISAHS algorithm, there are 10 standard functions (Sphere, Rosenbrock, Rastrigin, Griewank, Ackley, Rotated hyper, Zakharov, Schwefel's problem 2.22, Schwefel's problem 2.21 and Schwefel's problem 1.2), which will be used to compare ISAHS, NGHS [24], HS, PSO, DE(rand/1/bin), RMDE [25], and SRDE [26].

The variable range of Ackley function is $[-32, 32]$, and the variable range of other 9 functions is $[-100, 100]$. The variable dimension (D) of all standard functions is equal to 10. The population size (M) of all tested algorithms is set as 60. In each iteration, HS algorithms only update an old solution (serial search), while PSO and DE algorithms update the whole population (parallel search). Therefore, the maximum iterations (T) of PSO and DE algorithms are set as 1500, when the maximum iterations of HS algorithms are 9×10^4 (1500×60).

Other key parameters are set as follows: the parameters of ISAHS: $HMCR_{max} = 0.99$, $HMCR_{min} = 0.9$, $PAR_{max} = 1$, $PAR_{min} = 0.3$, $c_1 = 1.5$, $c_2 = 1.8$; the parameter of NGHS: $p_m = 0.05$; the parameters of HS: $HMCR = 0.9$, $PAR = 0.3$, $BW = 0.01$; the parameters of PSO: $c_1 = c_2 = 2$, $w = 0.6$, $v_{max} = 1$, $v_{min} = -1$; the parameters of DE: $F = 0.5$, $CR = 0.9$; the parameters of RMDE are shown in [25]; the parameters of SRDE are shown in [26].

Table 1 shows that: 1) Among HS, NGHS and ISAHS, except for the 'best' of f_2 , the proposed ISAHS algorithm is obviously better than other two algorithms in optimization accuracy, average value and stability, which indicates that ISAHS has been greatly optimized compared with the classical harmony search algorithm. 2) Compared with PSO and DE algorithms, it can be seen that the proposed ISAHS algorithm has better optimization accuracy, average value and stability than other algorithms in $f_1, f_3 \sim f_6$, and f_8 standard test functions. However, among the four test functions f_2, f_7, f_9 , and f_{10} , the optimization results of DE algorithms are better, but the 'Mean' and 'Std' of ISAHS are very close to DE algorithms in f_2 and f_9 functions.

Fig. 5 is the convergence characteristics of 7 algorithms above. To ensure that they can be compared in the same coordinate system, this paper selects an evaluation value from each generation of PSO and DE algorithms, while HS algorithms select an evaluation value every 60 generations. From Fig. 5, the following conclusions can be drawn: 1) The convergence characteristics of ISAHS are significantly better than other 6 algorithms except for f_2, f_7 , and f_{10} . 2) By observing f_1, f_3 and f_6 , it is obvious that the biggest characteristic of ISAHS is its slow convergence speed at the beginning and high convergence speed at the end, which is helpful to avoid falling into local extremum and ensure an overall fast convergence speed.

TABLE 1. Optimization results.

Function		ISAHS	NGHS	HS	PSO	DE	RMDE	SRDE
f_1	Best	3.00e-265	6.65e-37	7.37e-08	2.83e-09	4.25e-62	1.51e-184	9.99e-164
	Mean	1.88e-252	5.74e-34	1.87e-07	1.14e-04	3.73e-60	8.33e-176	7.51e-156
	Std	0.00e-00	1.44e-33	1.10e-07	2.12e-04	6.70e-60	0.00e-00	2.87e-155
f_2	Best	2.01e-02	1.79e-02	1.91e-02	3.29e-01	3.20e-06	0.00e-00	0.00e-00
	Mean	4.65e-00	4.78e-00	5.31e+01	1.64e+01	1.75e-00	2.47e-00	9.76e-01
	Std	3.17e-00	5.46e-00	7.31e+01	3.12e+01	1.30e-00	1.62e-00	1.81e-00
f_3	Best	0.00e-00	0.00e-00	3.69e-05	2.39e+01	0.00e-00	1.99e-00	0.00e-00
	Mean	0.00e-00	0.00e-00	7.30e-01	6.26e+01	9.16e-00	8.29e-00	3.02e-00
	Std	0.00e-00	0.00e-00	9.03e-01	2.59e+01	6.07e-00	5.67e-00	3.05e-00
f_4	Best	0.00e-00	9.90e-03	7.57e-09	6.89e-02	0.00e-00	7.40e-03	2.71e-02
	Mean	0.00e-00	4.80e-02	3.24e-02	1.64e-01	2.09e-02	5.86e-02	7.68e-02
	Std	0.00e-00	2.47e-02	2.47e-02	7.60e-02	3.17e-02	3.93e-02	4.76e-02
f_5	Best	8.88e-16	4.44e-15	2.44e-04	3.16e-05	4.44e-15	8.88e-16	4.44e-15
	Mean	4.32e-15	1.94e-14	7.99e-04	1.56e+00	4.88e-15	9.34e-02	1.16e-01
	Std	6.49e-16	6.82e-15	5.03e-04	1.04e+00	1.32e-15	3.61e-01	3.53e-01
f_6	Best	3.71e-266	6.48e-40	2.65e-07	5.15e-10	7.18e-62	2.85e-183	5.36e-162
	Mean	1.74e-250	5.58e-34	8.73e-07	8.05e-04	5.36e-59	2.66e-177	1.36e-153
	Std	0.00e-00	1.31e-33	4.15e-07	1.80e-03	1.15e-58	0.00e-00	6.96e-153
f_7	Best	8.99e-27	1.50e-03	6.42e-00	4.42e-08	1.08e-39	2.42e-123	4.49e-82
	Mean	1.52e-12	3.57e-00	1.78e+02	8.73e-04	1.77e-37	9.48e-114	5.14e-74
	Std	8.19e-12	1.34e-01	1.64e+02	2.50e-03	2.47e-37	3.27e-113	2.81e-73
f_8	Best	2.78e-129	1.98e-19	4.07e-04	0.0075	1.74e-29	6.54e-94	1.80e-91
	Mean	3.20e-87	1.53e-18	7.44e-04	0.2546	1.49e-27	2.68e-83	5.06e-87
	Std	1.75e-87	2.18e-18	2.15e-04	1.0516	5.09e-27	1.35e-82	1.05e-86
f_9	Best	3.42e-29	1.75e-04	1.70e-03	6.60e-03	5.63e-23	8.07e-61	5.40e-46
	Mean	4.70e-22	5.30e-03	1.18e-01	7.15e-02	4.06e-22	4.22e-11	1.53e-07
	Std	2.41e-21	6.50e-03	9.03e-02	5.85e-02	2.81e-22	2.11e-10	6.44e-07
f_{10}	Best	1.29e-12	1.20e-05	4.7869	3.67e-06	2.82e-38	3.20e-114	4.84e-73
	Mean	5.76e-07	3.18e-03	27.8907	2.9e-03	5.00e-36	1.65e-87	3.15e-63
	Std	1.20e-06	2.85e-03	19.6892	6.7e-03	8.35e-36	9.05e-87	1.37e-62

TABLE 2. Optimization results of solving static path planning problem based on ISAHS, NGHS, HS, PSO, DE, and SRDE.

Test example		ISAHS	NGHS	HS	PSO	DE	SRDE
Environment 1	Best	170.6001	171.0647	182.6149	172.0583	179.7753	170.9168
	Mean	180.2136	202.4938	214.7871	227.3567	266.2537	225.9100
	Std	13.1915	24.1286	27.6330	28.9652	55.5319	44.5941
	Time	18.2104s	18.6194s	18.9686s	14.5680s	15.9369s	17.3131s
	Success	23	8	19	3	12	10
Environment 2	Best	168.2779	168.5472	176.4310	169.7591	177.0182	169.2020
	Mean	172.9642	176.5649	205.2298	221.8777	310.8856	230.6292
	Std	9.9797	12.0467	28.3795	64.5069	75.3799	45.0985
	Time	11.4926s	11.4941s	12.2101s	11.8568s	10.0522s	10.7190s
	Success	28	24	20	9	7	8
Environment 3	Best	168.6786	169.0078	181.1992	172.8342	196.1707	168.6902
	Mean	173.0144	175.2665	203.0196	208.6557	258.4059	217.0124
	Std	4.0258	7.1757	14.3451	34.1995	29.0104	44.0035
	Time	4.5768s	3.8809s	5.5565s	3.9349s	3.8717s	4.1013
	Success	30	29	20	16	24	21

B. PATH PLANNING SIMULATION EXPERIMENT IN STATIC ENVIRONMENT

This part mainly compares the static path planning results of ISAHS, NGHS, HS, PSO, DE and SRDE. Firstly, three different static environments for mobile robot are set,

as shown in Fig. 6. The blue objects in environment 1 are mainly concave obstacles; the blue objects in environment 2 are mainly polygonal and circular obstacles; the blue objects in environment 3 are mainly circular obstacles. In the three environments, the starting position (S) is [30, 30] when

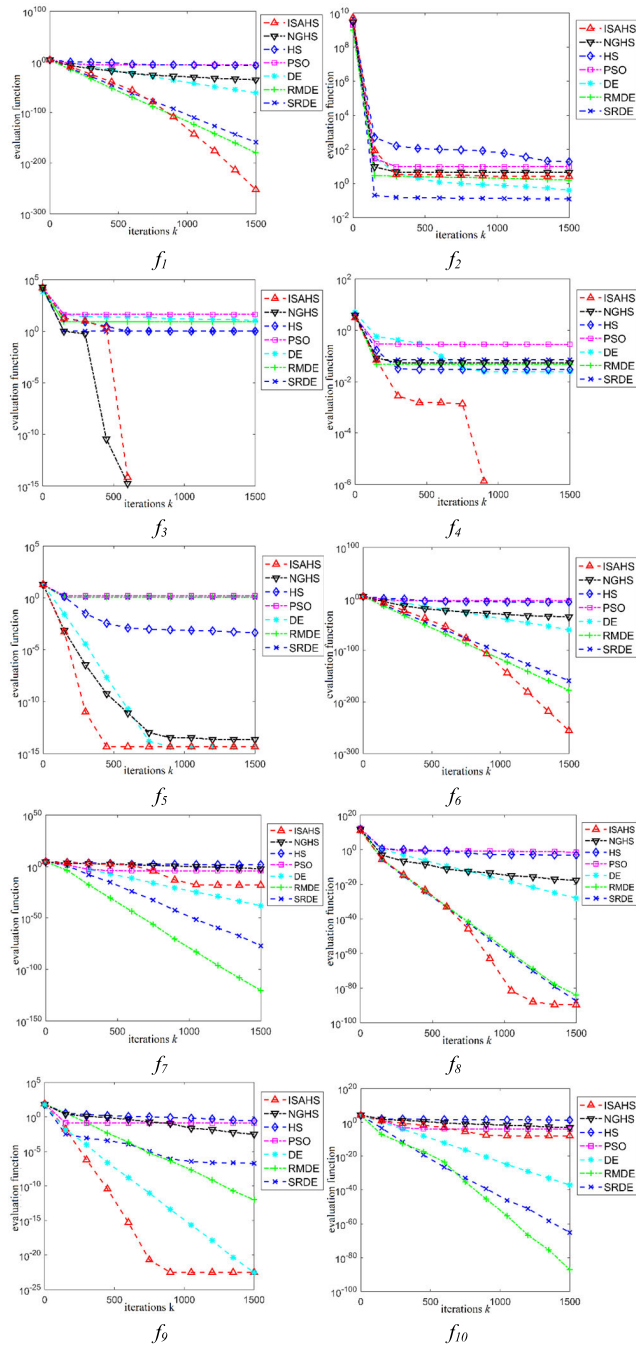


FIGURE 5. Convergence characteristics of each algorithm.

the goal position (G) is [160, 160]. In addition, the variable range is [-100, 100], the variable dimension (D) is 20, the maximum iterations (T) of HS algorithms are changed to 9×10^3 , the maximum iterations of PSO and DE algorithms are changed to 150, and the parameters of other algorithms are consistent with Section A.

To evaluate the search accuracy, stability and average consumption time for solving path planning problem, the above 6 algorithms have been run independently for 30 times in the three static environments respectively. Table 2 are the

TABLE 3. Optimization results of solving static path planning problem based on ISAHS, WOA, and IWOA.

Test example		Best	Mean	Std	Time	Success
Environment 1	ISAHS	170.6001	180.2136	13.1915	18.2104s	23
	WOA	173.9080	201.1990	17.1645	15.9807 s	3
	IWOA	174.2718	201.7048	17.9359	16.4616s	2
Environment 2	ISAHS	168.2779	172.9642	9.9797	11.4926s	28
	WOA	170.7138	191.9093	14.4851	10.2181s	18
	IWOA	169.2456	180.9450	13.4854	10.1492 s	20
Environment 3	ISAHS	168.6786	173.0144	4.0258	4.5768s	30
	WOA	171.6544	189.7522	20.1713	3.6726 s	26
	IWOA	169.8532	183.0307	15.0028	3.8463s	28

optimization results, where ‘Best’, ‘Mean’, ‘Std’, ‘Time’ and ‘Success’ respectively represent the best value, mean value, standard deviation, average consumption time and the successful times of each algorithm in 30 independent simulation experiments. The definition of ‘success’ is that the path obtained by above algorithms is not crashed with obstacles.

Table 2 shows that: 1) The ‘Best’ and ‘Mean’ of ISAHS in three environments are superior to the other five algorithms, which means that the search accuracy of ISAHS is higher in solving static path planning problem. 2) The ‘Std’ and ‘Success’ of ISAHS are significantly superior to those of NGHS, PSO and DE algorithms, which means that the proposed ISAHS algorithm has higher stability.

Fig. 6 are static path planning results and convergence characteristics. The left figure is the best path in 30 independent path planning simulation experiments, and the right figure is the corresponding convergence characteristics of each algorithm. Combined with the ‘Mean’ in table 2, it can be seen that all algorithms can find a collision free path, but the ISAHS is the best. In addition, although NGHS has fast convergence speed, its final search accuracy is poor due to premature convergence. On the contrary, the convergence speed of ISAHS is relatively slow in the early stage to avoid falling into the local extremum, while the convergence speed is relatively fast in the late stage to improve the search accuracy. Therefore, ISAHS is better and more stable in solving static path planning problem.

Compared with classical optimization algorithms as PSO, DE and SRDE, whale optimization algorithm (WOA) proposed by Mirjalili and Lewis in 2016, has fewer parameters, faster convergence speed and excellent performance in many optimization problems [36]. Therefore, this paper also compares the static path planning results between the proposed algorithm ISAHS and the latest optimization algorithms WOA and IWOA proposed in [37].

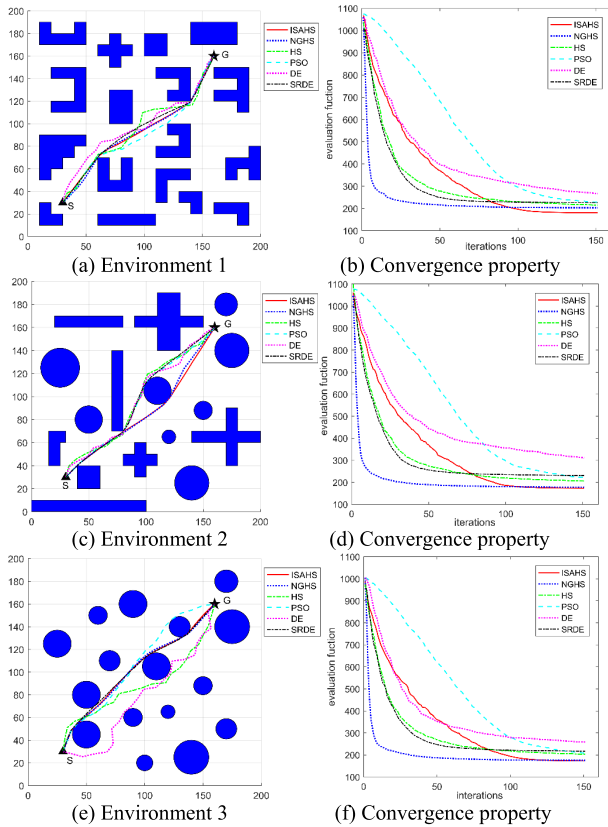


FIGURE 6. Static path planning based on ISAHS, NGHS, HS, PSO, DE, and SRDE.

From table 3, it is obvious that the proposed algorithm ISAHS performs better in ‘Best’, ‘Mean’, ‘Std’ and ‘Success’ compared with the latest optimization algorithms WOA and IWOA. Fig. 7 also shows that the proposed algorithm ISAHS has lower convergence speed compared with WOA and IWOA, but its final search accuracy is better.

To evaluate the impact of the change of dimension (D) and population size (M) on path planning, Fig. 8 shows the changes of the mean value and standard deviation of the evaluation function of each algorithm in Environment 1. When $D = 10$ or $M = 30$, the maximum iterations of HS algorithms are 6000, and the maximum iterations of PSO algorithm and DE algorithms are 100; when $D = 20$ or $M = 60$, the maximum iterations of HS algorithms are 9000, and the maximum iterations of PSO algorithm and DE algorithms are 150; when $D = 30$ or $M = 90$, the maximum iterations of HS algorithms are 12000, and the maximum iterations of PSO algorithm and DE algorithms are 200.

From Fig. 8(a) and Fig. 8 (b), with the increase of dimension D , the mean value and standard deviation of ISAHS algorithm are still smaller than other 5 algorithms, which indicates that the proposed ISAHS algorithm is more suitable for solving high-dimensional optimization problem. In Fig. 8(c) and Fig. 8(d), it’s obvious that the proposed ISAHS algorithm still performs better than other 5 algorithms with the increase of population size M . In addition, the mean value and standard deviation will be worse if the population

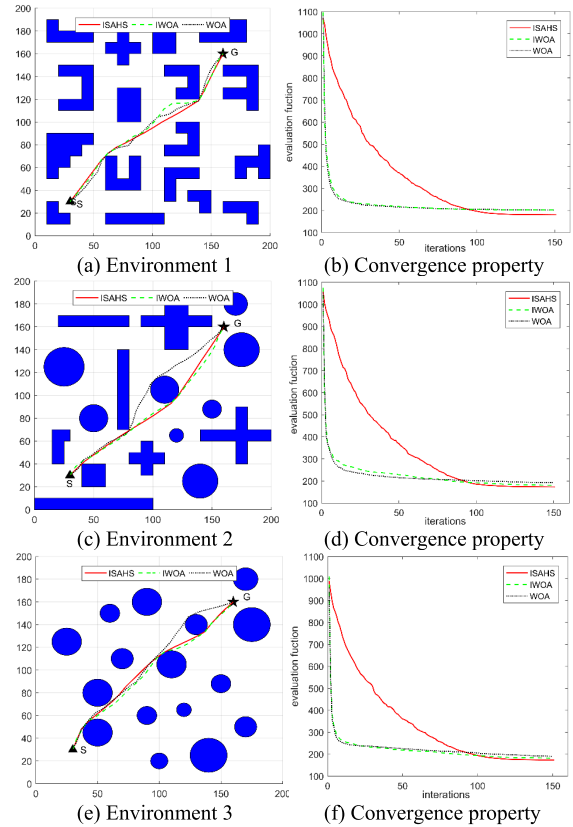


FIGURE 7. Static path planning based on ISAHS, WOA and IWOA.

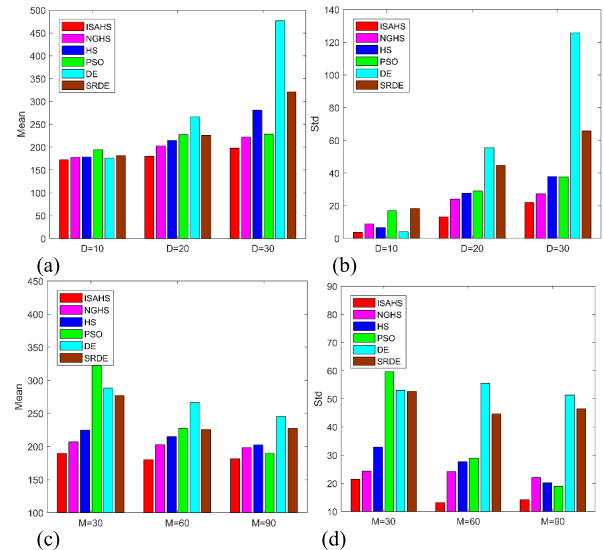


FIGURE 8. The changes of ‘Mean’ and ‘Std’ with the increase of dimension D or population size M .

sized M is small, so we should make the value of M larger relatively.

C. SIMULATION EXPERIMENT FOR GLOBAL DYNAMIC PATH PLANNING

To verify the effectiveness of global dynamic path planning method proposed above, this part will do further simulation

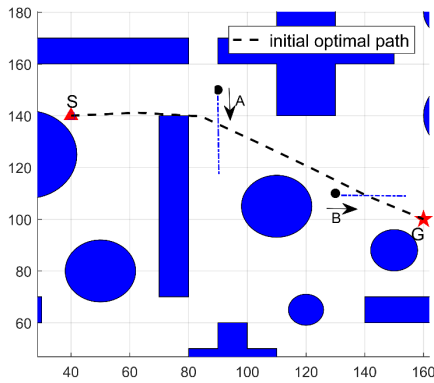


FIGURE 9. Dynamic environment.

TABLE 4. Motion parameters.

Category	Parameters
obstacle A	point 1 (90, 150); point 2 (90, 130); moving speed 1.1; course angle $\pm 90^\circ$
obstacle B	point 1 (130, 110); point 2 (150, 110); moving speed 1.1; course angle $0^\circ, 180^\circ$
robot	start position (40, 140); goal position (160, 100); moving speed 2
other	rolling window radius is 12; minimum collision distance is 5

TABLE 5. Minimum distance between mobile robot and dynamic obstacles.

Path	Minimum distance from obstacle A	Minimum distance from obstacle B
Initial optimal path	0.9562	2.3769
Actual path	5.6350	7.6450

Fig. 10 is the result of global dynamic path planning. From (a) to (d), the positions of the mobile robot before the collision with the dynamic obstacle A, after avoiding collision with the dynamic obstacle A, before the collision with the dynamic obstacle B, and after avoiding collision with the dynamic obstacle B are respectively presented. The black dotted line is the initial optimal path, and the red solid line is the actual moving trajectory of mobile robot after obstacle avoidance. In addition, Table 5 shows the minimum contact distance between the mobile robot and dynamic obstacles when the mobile robot is moving on the initial optimal path and the modified path respectively. According to Fig. 10 and table 5, it can be seen that the designed global dynamic path planning method can avoid all preset moving obstacles.

V. CONCLUSION AND FUTURE WORK

A global dynamic path planning method based on improved self-adaptive harmony search algorithm (ISAHS) and Morphing algorithm is proposed in this paper. Compared with PSO, WOA algorithms, DE algorithms or other improved HS algorithms, the designed ISAHS algorithm apparently has higher stability and search accuracy, better convergence characteristics and can obtain a better initial optimal path in the static environment. In addition, combined with Morphing algorithm, the proposed global dynamic path planning method can avoid the dynamic obstacles in real time.

This study is an initial exploration of HS algorithm for dynamic path planning. Further research on the improvement of population-based meta-heuristic algorithms can be pursued in a few aspects. Note that mobile robot dynamic path planning is a hot research problem, how to overcome different dynamic obstacle still value to study. One interesting research direction might be to embed other new strategies into the HS search structure. It would also be interesting to construct some significant hybrid algorithms.

ACKNOWLEDGMENT

The authors would like to thank Geem for the useful information about harmony search algorithm. In particular, they are grateful to Dexuan Zou for providing information about his proposed algorithm-NGHS.

COMPLIANCE WITH ETHICAL STANDARD

Conflict of interest the authors declare that there is no conflict of interests regarding the publication of this paper. All the authors have no conflict of interest. This article does not contain any studies with human participants performed by any of the authors.

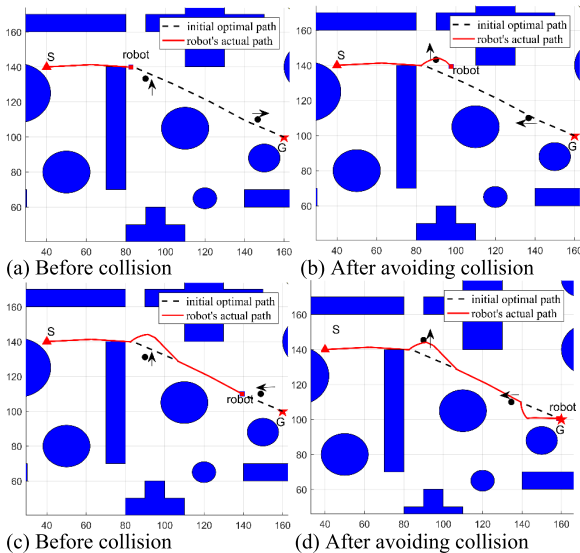


FIGURE 10. Global dynamic path planning.

experiment based on Section B. Fig. 9 shows the dynamic environment for mobile robot, which mainly adds two obstacles that move back and forth along a straight line (dashed blue line) on the basis of environment 2 (Fig. 7). We assume that the mobile robot will travel at a constant speed on the initial optimal path. The local dynamic path planning will be carried out if it is predicted that the moving robot will collide with dynamic obstacles. The motion parameters of dynamic obstacles and mobile robot are shown in table 4.

REFERENCES

- [1] W. Gao, Q. Tang, B. Ye, Y. Yang, and J. Yao, "An enhanced heuristic ant colony optimization for mobile robot path planning," *Soft Comput.*, vol. 24, no. 8, pp. 6139–6150, Apr. 2020.
- [2] T. F. Abaas and A. H. Shabeeb, "Path planning optimization of a mobile robot based on intelligence algorithm," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 881, Aug. 2020, Art. no. 012047.
- [3] M. S. Gharajeh and H. B. Jond, "An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system," *Ain Shams Eng. J.*, May 2021, doi: 10.1016/j.asej.2021.05.005.
- [4] C. Zhang, L. Zhou, Y. Li, and Y. Fan, "A dynamic path planning method for social robots in the home environment," *Electronics*, vol. 9, no. 7, p. 1173, Jul. 2020.
- [5] X. Yingqi, S. Wei, Z. Wen, L. Jingqiao, L. Qinhui, and S. Han, "A real-time dynamic path planning method combining artificial potential field method and biased target RRT algorithm," *J. Phys., Conf. Ser.*, vol. 1905, no. 1, May 2021, Art. no. 012015.
- [6] B. Yang, J. Yan, Z. Cai, Z. Ding, D. Li, Y. Cao, and L. Guo, "A novel heuristic emergency path planning method based on vector grid map," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 6, p. 370, May 2021.
- [7] S. Xiao, X. Tan, and J. Wang, "A simulated annealing algorithm and grid map-based UAV coverage path planning method for 3D reconstruction," *Electronics*, vol. 10, no. 7, p. 853, Apr. 2021.
- [8] T. Lin, "A path planning method for mobile robot based on A* and ant-colony algorithms," *J. Innov. Soc. Sci. Res.*, vol. 7, no. 1, pp. 157–162, 2020.
- [9] C. Zong, X. Han, D. Zhang, Y. Liu, W. Zhao, and M. Sun, "Research on local path planning based on improved RRT algorithm," *Proc. Inst. Mech. Eng.*, vol. 235, no. 8, pp. 2086–2100, Mar. 2021.
- [10] P. Zhong, T. Li, M. Liu, X. Chen, and X. Zhang, "Robot path planning method based on improved genetic algorithm and Morphine algorithm," *Mach. Tool Hydraul.*, vol. 47, no. 24, pp. 33–38, Dec. 2019.
- [11] Z. Chen, Z. Zhang, J. Li, and T. Jiang, "Mobile robots path planning based on improved artificial potential field," *Comput. Eng. Appl.*, vol. 55, no. 23, pp. 29–34, 2019.
- [12] C. Zhuge, Z. Tang, and Z. Shi, "A local path planning algorithm for UGV based on multilayer Morphine search tree," *Robot.*, vol. 36, no. 4, pp. 491–497, Apr. 2014.
- [13] A. V. Le, N. H. K. Nhan, and R. E. Mohan, "Evolutionary algorithm-based complete coverage path planning for tetrahedron tiling robots," *Sensors*, vol. 20, no. 2, p. 445, Jan. 2020.
- [14] Y. Liu, N. Qi, W. Yao, J. Zhao, and S. Xu, "Cooperative path planning for aerial recovery of a UAV swarm using genetic algorithm and homotopic approach," *Appl. Sci.*, vol. 10, no. 12, p. 4154, Jun. 2020.
- [15] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107376.
- [16] Y. Li, S. Wang, and B. Yang, "An improved differential evolution algorithm with dual mutation strategies collaboration," *Expert Syst. Appl.*, vol. 153, Sep. 2020, Art. no. 113451.
- [17] L. Wang, K. Jiang, T. Wei, C. Jiang, J. Zha, and S. Chi, "Estimation of parameters of probability integral method model based on improved fireworks algorithm," *Surv. Rev.*, vol. 53, no. 379, pp. 366–382, Jul. 2021.
- [18] G. Li and W. Chou, "Path planning for mobile robot using self-adaptive learning particle swarm optimization," *Sci. China Inf. Sci.*, vol. 61, no. 5, May 2018, Art. no. 052204.
- [19] B. Fan and C. Zhang, "Hybrid differential evolution algorithm for mobile robot path planning," *J. Trans. Univ. Sci. Technol.*, vol. 40, no. 1, pp. 6–12, Feb. 2019.
- [20] Q. Zhu, X. Tang, Y. Li, and M. O. Yeboah, "An improved differential-based harmony search algorithm with linear dynamic domain," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104809.
- [21] J. Wu, J. Yi, L. Gao, and X. Li, "Cooperative path planning of multiple UAVs based on PH curves and harmony search algorithm," in *Proc. IEEE CSCWD*, Wellington, New Zealand, Apr. 2017, pp. 540–544.
- [22] S. Kundu and D. R. Parhi, "Navigation of underwater robot based on dynamically adaptive harmony search algorithm," *Memetic Comput.*, vol. 8, no. 2, pp. 125–146, Apr. 2016.
- [23] H. Binol, E. Bulut, K. Akkaya, and I. Guvenc, "Time optimal multi-UAV path planning for gathering its data from roadside units," in *Proc. IEEE VTC-Fall*, Chicago, IL, USA, Aug. 2018, pp. 1–5.
- [24] D. Zou, L. Gao, J. Wu, and S. Li, "Novel global harmony search algorithm for unconstrained problems," *Neurocomputing*, vol. 73, no. 3, pp. 3308–3318, 2010.
- [25] H. Ouyang, L. Gao, and X. Kong, "Random mutation differential evolution algorithm," *J. Northeastern Univ.*, vol. 34, no. 3, pp. 330–334, Mar. 2013.
- [26] X. Shen, D. Zou, and X. Zhang, "Self-adaptive differential evolution algorithm with random mutation," *Electron. Sci. Tech.*, vol. 31, no. 2, pp. 51–55, Feb. 2018.
- [27] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *J. Simul.*, vol. 76, no. 2, pp. 60–68, Feb. 2001.
- [28] M. Dubey, V. Kumar, M. Kaur, and T.-P. Dao, "A systematic review on harmony search algorithm: Theory, literature, and applications," *Math. Problems Eng.*, vol. 2021, pp. 1–22, Apr. 2021.
- [29] L. Abualigah, A. Diabat, and Z. W. Geem, "A comprehensive survey of the harmony search algorithm in clustering applications," *Appl. Sci.*, vol. 10, no. 11, p. 3827, May 2020.
- [30] H.-B. Ouyang, L.-Q. Gao, S. Li, X. Kong, and D.-X. Zou, "On the iterative convergence of harmony search algorithm and a proposed modification," *Appl. Math. Comput.*, vol. 247, pp. 1064–1095, Nov. 2014.
- [31] D. Zou, L. Gao, S. Li, J. Wu, and X. Wang, "A novel global harmony search algorithm for task assignment problem," *J. Syst. Softw.*, vol. 83, no. 10, pp. 1678–1688, Oct. 2010.
- [32] D. Zou, L. Gao, S. Li, and J. Wu, "Solving 0–1 knapsack problem by a novel global harmony search algorithm," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1556–1564, Mar. 2011.
- [33] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, "A novel global harmony search algorithm for reliability problems," *Comput. Ind. Eng.*, vol. 58, no. 2, pp. 307–316, Mar. 2010.
- [34] H. Ouyang, L. Gao, S. Li, X. Kong, Q. Wang, and D. X. Zou, "Improved harmony search algorithm: LHS," *Appl. Soft Comput.*, vol. 53, pp. 133–167, Apr. 2017.
- [35] H.-B. Ouyang, L.-Q. Gao, and S. Li, "Amended harmony search algorithm with perturbation strategy for large-scale system reliability problems," *Int. J. Speech Technol.*, vol. 48, no. 11, pp. 3863–3888, Apr. 2018.
- [36] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
- [37] K. Wu and S. Tan, "Path planning of UAVS based on improved whale optimization algorithm," in *Proc. ACAS*, vol. 41, no. S2, pp. 107–114, Dec. 2020.
- [38] Y. Chen and H. Xiao, "Mobile-robot dynamic path planning based on improved A* and Morphine algorithms," *CAAI Trans. Intell. Syst.*, vol. 15, no. 3, pp. 546–552, May 2020.



YONGBIN QUAN received the B.S. degree from Guangzhou University. He is currently pursuing the M.S. degree in electronic information. He has published some articles on the robot path planning. His current research interests include intelligent optimization algorithms and robotic path planning.



HAIBIN OUYANG received the M.S. and Ph.D. degrees in control theory and control engineering from Northeastern University (NEU), Shenyang, China, in 2012 and 2016, respectively. He is currently an Associate Professor with the School of Mechanical and Electric Engineering, Guangzhou University, Guangzhou, China. He has published over 50 journal articles such as *Information Sciences*, *Applied Soft Computing*, *Soft Computing*, and *Applied Mathematics and Computation*. His

current research interests include intelligent optimization algorithm, robotics path planning, artificial intelligence, and optimal control.



analysis and processing, and intelligent optimization algorithm research.

CHUNLIANG ZHANG received the M.S. degree in mechanical engineering from Xi'an Jiaotong University, China, in 1997, and the Ph.D. degree in mechanical engineering from Zhejiang University, China, in 2004. He is currently a Professor with the School of Mechanical and Electric Engineering, Guangzhou University, China. He has published over 150 journal articles. His research interests include condition monitoring and control of complicated operating environment, robotics, signal



control theory and control application, and pattern recognition.

LI-QUN GAO received the M.S. and Ph.D. degrees in automatic control from Northeastern University, Shenyang, China, in 1985 and 1991, respectively. He is currently a Professor with the College of Information Science and Engineering, Northeastern University. He has published over 100 journal articles including *Information Sciences*, *Expert Systems with Applications*, and *Neurocomputing*. His current research interests include intelligence optimization and application,

• • •



80 publications in international journals, including *Journal of International Financial Markets, Institutions and Money*, *European Journal of Finance*, *International Review of Economics and Finance*, *Applied Energy*, and *Water Resources Research*. His current research interests include quantitative finance and financial management, intelligent algorithm, and its application.

STEVEN LI received the B.Sc. degree from Tsinghua University, China, the MBA degree from Melbourne Business School, and the Ph.D. degree from the Delft University of Technology, The Netherlands. He is currently a Professor of finance with RMIT University, Melbourne, Australia. He has previously taught at University of South Australia, Queensland University of Technology, Edith Cowan University, and Tsinghua University. He has published extensively with over