An Improved Manta Ray Foraging Optimizer for Mobile Robot Path Planning

Xiangdong Wu[1,2], Kaoru Hirota[1,2], Bijun Tang[1,2], Tianrui Liao[1,2], Yaping Dai[1,2], Zhiyang Jia[1,2,*]

[1]School of Automation, Beijing Institute of Technology, No. 5 Zhongguancun South Street, Haidian District, Beijing 100081, China
E-mail: wuxiangdongphd@bit.edu.cn
[2]State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing, 100081, China
E-mail: wuxiangdongphd@bit.edu.cn

Abstract. A path planner based on improved manta ray foraging optimizer (IMRFO) is proposed for global path planning of mobile robot, in which manta ray foraging optimizer (MRFO) and Cauchy mutation are combined to find the optimal path with the designed cost function. The proposed IMRFO not only is effective to find an optimal path, but also improves the global search accuracy, speed and stability in path planning of MRFO. To compare the performance of mobile robot path planning based on IMRFO, MRFO and Particle Swarm Optimization (PSO), simulation experiments are implemented by MATLAB 2019a in the Windows10 operating system. With 25 times independent experiments, it is shown that IMRFO acquires an effective path and its performance is superior to MRFO and PSO with the evaluation of statistical results and convergence curves. The IMRFO provides a powerful tool in dealing with mobile robot path planning problem.

Keywords: Mobile Robot, Global Path Planning, Manta Ray Foraging Optimizer, Cauchy Mutation

1. INTRODUCTION

To improve the work efficiency, it is important for the mobile robot to find the optimal path from the start point to the goal point without collision. Moreover, path planning is a non-deterministic polynomial (NP)-hard problem. The importance and difficulty make path planning become a long-term research topic. The path planning algorithms are mainly classified into two categories. One is the traditional algorithm, such as artificial potential field algorithm (APF) [1], visibility graph algorithm [2], rapidly-exploring random trees (RRT) algorithm [3], A* algorithm [4] and so on. Although these methods get paths with obstacles avoidance, they have different kinds of disadvantages. The APF is easy to fall into the local optimal solution and the goal point is unreachable. The visibility graph algorithm lacks flexibility that the visibility graph must be reconstructed when starting point and goal point are changed. The path generated by RRT is tortuous and the search direction is not guided. The A* algorithm has disadvantages of large memory overhead and long calculation time. The other one is intelligent optimization algorithm. The intelligent optimization method has the advantages of simplicity, efficiency and adaptability compared with the traditional method, and has become a research hotspot with these advantages. Hossain et al. [5] develop a new algorithm for autonomous robot path planning based on Bacterial Foraging Optimization technique. Mo et al. [6] present a new method of global path planning by combining Biogeography-based Optimization (BBO), PSO and approximate voronoi boundary network (AVBN). In [7], a modified membrane-inspired algorithm based on PSO is proposed to solve the multi-objective mobile robot path planning. Wang et al. [8] adopt the ant colony algorithm as initial values and used a simulated annealing genetic algorithm to optimize the paths. Xue et al. [9] design a non-dominated sorting genetic algorithm for multi-objective path planning in static environments. An improved chicken swarm optimization (ICSO) algorithm based on Levy flight strategy and nonlinear weight reduction strategy is proposed and applied in robot path planning in [10]. However, no optimization algorithm performs best for all engineering problems and a room of improvements still exists according to the No-Free-Lunch theorem [11]. So new attempts of optimization algorithm shall continue for path planning.

In line with this, a path planner based on IMRFO is proposed for global path planning of mobile robot. The MRFO is proposed by Zhao et al. in 2020 [12]. This algorithm emulates three unique foraging strategies of manta rays, including chain foraging, cyclone foraging, and somersault foraging. The MRFO has successfully been applied in defining the uncertain parameters of fuel cells [13], extracting the accurate parameters of three-diode equivalent model (3DeM) of solar generating units (SGUs) [14], and so on. So the MRFO has the potential to be applied in mobile robot path planning. For the path planner based on MRFO, navigation point model is used to build the environment model. The MRFO is used to find the optimal path with the cost function of the shortest path and no collision for global path, and spline function is applied to smooth the global path. However, the performance of MRFO is not stable enough to find its optimal path in the preliminary experiments. To improve the performance of MRFO for path planning, a path planner based on IMRFO is proposed, in which the somersault foraging strategy of MRFO is combined with Cauchy mutation.

Although the MRFO has the merits of less computational expense, higher performance and solution accuracy, it is difficult to escape from local optimum when applied into path planning with limited iterations.

The IMRFO are implemented with increased populations diversity to make individuals get out of local minimum more easily than MRFO. Based on the IMRFO, the path planner not only has the ability to find an optimal path with shortest length of path and no collision, but also improves the global search accuracy, speed and stability in path planning of MRFO.

The simulation experiments of mobile robot path planning in a static working environment are implemented using MATLAB 2019a in the Windows10 operating system. The working environment of mobile robot is designed with multiple obstacles, which are represented by circles considering size of obstacles and robot. For comparison of path planning results in effectiveness, accuracy, convergence speed and stability, path planners based on IMRFO, MRFO and PSO are carried out to find an optimal path with shortest path and no collision in the same environment.

The details of IMRFO for path planning with obstacle avoidance is introduced in Section 2. The results of experiments on path planning with PSO, MRFO and IMRFO are described in Section 3.

2. CONSTRUCTION OF PATH PLANNER BASED ON IMRFO

2.1. Environment Modeling

Among the commonly used environment models, including topology map model, grid map model, navigation point model and so on, the navigation point model [15-16] is suitable for MRFO algorithm to deal with obstacles and related boundary constraint conditions.

In the navigation point model, circles are used to describe the obstacles. To consider the size of the mobile robot, the obstacles' radius R is obtained through expansion with the circumradius of mobile robot r as shown in **Fig. 1**. With this kind of description of obstacles, the robot is considered as a point in the robot configuration space, which is convenient for path planning with obstacle avoidance. So a circle equation given by (1) is used to describe the obstacles in the working space of mobile robot.

$$(x - x_o)^2 + (y - y_o)^2 = R^2, \tag{1}$$

where (x_o , y_o) is the center coordinate of the obstacle, R is the radius of the expanded obstacle.

With the environment modeling method mentioned above, the path is generated by connecting a series of navigation points with short straight lines. For the navigation points, three coarse navigation points are determined by intelligent optimization algorithm at first. Then one hundred precise navigation points determined by the fitting results of spline function with the coarse navigation points, the start point and the goal point are used to constitute the path.

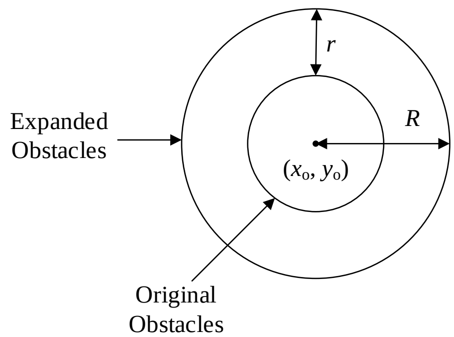


Fig. 1 Expanded obstacles size with mobile robot size.

2.2. Cost Function Construction

To find an optimal path based on the navigation points with obstacle avoidance, cost function is constructed with two objectives, the shortest path and no collision.

For the length of the path, the Euclidean distance between two navigation points is used to construct the path length function as

$$l = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} , \tag{2}$$

where l is the path length from the start point to the goal point, n is the total number of precise navigation points generated by spline function, (x_i , y_i) and (x_{i+1} , y_{i+1}) are the adjacent precise navigation points.

For the requirement of no collision, a variable V_k is designed to keep the mobile robot away from the obstacles given by

$$v_k = \max(1 - \frac{\sqrt{(xx - x_{ok})^2 + (yy - y_{ok})^2}}{R_k}, 0), \tag{3}$$

$$V_k = \mathrm{mean}(v_k), \tag{4}$$

where xx and yy are the sets of abscissa and ordinate of all precise navigation points on a path respectively, (x_{ok} , y_{ok}) is the center coordinate of the k th obstacle, R_k is the radius of the k th expanded obstacle, v_k is the set to determine whether the path passes through the k th obstacle. If the path passes through the k th obstacle, there must be a number greater than 0 in the set v_k . If not, the numbers in the set v_k are all 0.

With (2) and (4), the total cost function is given by

$$f = l(1 + w\sum_{k=1}^{K} V_k), \tag{5}$$

where K is the total number of obstacles, w is the safe weight coefficient and set to 100 in this paper. If the path does not pass through the K obstacles, the term $\sum_{k=1}^{K} V_k$ is equal to 0. Otherwise, the term is equal to a number greater than 0. So with the safe weight coefficient w, a larger fitness value will obtain by (5) when the path passes through the obstacles. It aims to get the path of smallest fitness value with (5), so the path through obstacles of larger fitness value will be discarded to ensure a feasible path.

2.3. MRFO

The MRFO is a novel intelligent optimization algorithm inspired by three unique foraging strategies from manta rays, including chain, cyclone and somersault foraging. The initialization of MRFO is similar with other intelligent optimization algorithm in a random way as described by

$$x_{\text{rand}}^d = L_{\text{b}}^d + r(U_{\text{b}}^d - L_{\text{b}}^d), \tag{6}$$

where x_{rand}^d is a random position in the search space, L_{b}^d and U_{b}^d are the minimum and maximum boundary of the d th dimension in the search space, r is a random vector with the range of [0,1].

In the chain foraging strategy dedicated to the local search ability, the position of each individual except the first one is updated by the best solution found so far and the individual's position in front of it. The mathematical model of chain foraging strategy is depicted by

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + r(x_{\text{best}}^d(t) - x_i^d(t)) + \\ \alpha(x_{\text{best}}^d(t) - x_i^d(t)),\ i = 1 \\ x_i^d(t) + r(x_{i-1}^d(t) - x_i^d(t)) + \\ \alpha(x_{\text{best}}^d(t) - x_i^d(t)),\ i=2,3,\ldots,N \end{cases}, \tag{7}$$

$$\alpha = 2r\sqrt{|\log(r)|}, \tag{8}$$

where $x_i^d(t)$ is the position of i th individual of the swarm in d th dimension at time t, $x_{\text{best}}^d(t)$ is the best solution found so far, N is the size of population.

As for the cyclone foraging strategy, the position of each individual is updated in two types. In the first type, the individual position of the manta ray swarm is updated by the best solution and the individual's position in front of it similar with the chain foraging strategy, but in the way of cyclone foraging. The mathematical model of the first type is described by (9)-(10). The second type takes the random positions of the manta rays swarm instead of the best solution to contribute to the global search ability. The second type is given by (6) and (11). The two types are switched with the comparison result between $t/T_{\max}$ and random number in [0, 1].

$$x_i^d(t+1) = \begin{cases} x_{\text{best}}^d(t) + r(x_{\text{best}}^d(t) - x_i^d(t)) + \\ \beta(x_{\text{best}}^d(t) - x_i^d(t)),\ i = 1 \\ x_{\text{best}}^d(t) + r(x_{i-1}^d(t) - x_i^d(t)) + \\ \beta(x_{\text{best}}^d(t) - x_i^d(t)),\ i=2,3,\ldots,N \end{cases}, \tag{9}$$

$$\beta = 2e^{r_1 \frac{T_{\max} - t + 1}{T_{\max}}} \sin(2\pi r_1), \tag{10}$$

$$x_i^d(t+1) = \begin{cases} x_{\text{rand}}^d + r(x_{\text{rand}}^d - x_i^d(t)) + \\ \beta(x_{\text{rand}}^d - x_i^d(t)),\ i = 1 \\ x_{\text{rand}}^d + r(x_{i-1}^d(t) - x_i^d(t)) + \\ \beta(x_{\text{rand}}^d - x_i^d(t)),\ i=2,3,\ldots,N \end{cases}, \tag{11}$$

where $T_{\max}$ is the maximum number of iterations, r_1 is a random number among [0, 1].

The foraging strategy of MRFO switches between the chain foraging and cyclone foraging strategy randomly. After either of the two strategies is performed, the somersault foraging strategy is implemented. With the somersault foraging strategy enhancing the local search ability, the individual position is updated by somersaulting around the best solution found so far. The mathematical equation of this process is expressed as

$$x_i^d(t+1) = x_i^d(t) + S(r_2 x_{\text{best}}^d(t) - r_3 x_i^d(t)), \tag{12}$$

where S represents the somersault factor and is set to 2 here, r_2 and r_3 are two random numbers like r_1.

2.4. IMRFO

From the perspective of exploration and exploitation with the framework of MRFO, the value of $t/T_{\max}$ controls the balance of exploration and exploitation in the whole search process. $t/T_{\max} < rand$ leads to an exploration process, while $t/T_{\max} > rand$ leads to an exploitation process. When MRFO is implemented in the later period of $T_{\max}$, exploration process is implemented in a lower probability. This result may lead to fail to find the best solution. So a proper disturbance should be added in the local search process to enhance the global search ability of MRFO.

To achieve the goal mentioned above, Cauchy mutation is introduced into the somersault foraging strategy of MRFO, to mainly enhance the ability of exploration by escaping the local search space in the later period of $T_{\max}$. Moreover, the exploration of MRFO in the earlier period is also enhanced as well. The 1D standard Cauchy function is defined as

$$f(x) = \frac{1}{\pi}\frac{1}{x^2+1},\ -\infty < x < \infty. \tag{13}$$

The corresponding probability density function of Cauchy function is shown as

$$F(x)=\frac{1}{2}+\frac{1}{\pi}\arctan x\,. \tag{14}$$

The Cauchy mutation is introduced as the mutation step size. When the individual of MRFO falls into the local optimum, a larger step size helps the individual jump out of the local optimum. When the individual is close to convergence and searching for the optimal solution, a smaller step size accelerates the convergence speed of the individual.

The new mathematical model of somersault foraging strategy is given by

$$x_i^d(t+1)=x_i^d(t)+C(r_2x_{\text{best}}^d(t)-r_3x_i^d(t))\,, \tag{15}$$

where C is a random number of Cauchy distribution. Then by combining the new somersault foraging strategy in (15) with the chain and cyclone foraging strategy in MRFO, IMRFO is achieved with enhanced ability of exploration.

With the environment modeling and the cost function construction mentioned before, the general path planner algorithm flow of IMRFO is summarized in the flow chart shown in **Fig. 2**.

3. EXPERIMENTS OF IMRFO FOR PATH PLANNING WITH OBSTACLE AVOIDANCE

To verify the performance of the path planner based on IMRFO, experiments are simulated using MATLAB 2019a in the Windows10 operating system. The results of comparison experiments are implemented with basic MRFO and PSO in the same simulated working environment of the mobile robot. A simulated working environment for the path planner of the navigation point model is shown in **Fig. 3.** The circles represent the obstacles considering the size of the mobile robot as described in (1). The square and the pentagram are used to show the positions of the start point and the goal point, respectively.

For the simulation of path planning with PSO, MRFO and IMRFO, the size of population N and the maximum number of iterations $T_{\max}$ are set to 150 and 500, respectively. The maximum boundary U_{b} and minimum boundary L_{b} of search space are set to -8 and 8 according to the working environment of mobile robot in this simulation. In PSO, the inertial weight coefficient w_{PSO} which is set to 1 decreases linearly with a damping ratio 0.98, and the learning coefficients c_1 and c_2 are set to 1. The results of each algorithm are implemented 25 times independently, and the statistical results are listed in **Table 1**. In the table, the mean, optimal, worst and std are used to show the mean fitness value, the optimal fitness value, the worst fitness value and the standard deviation, respectively.

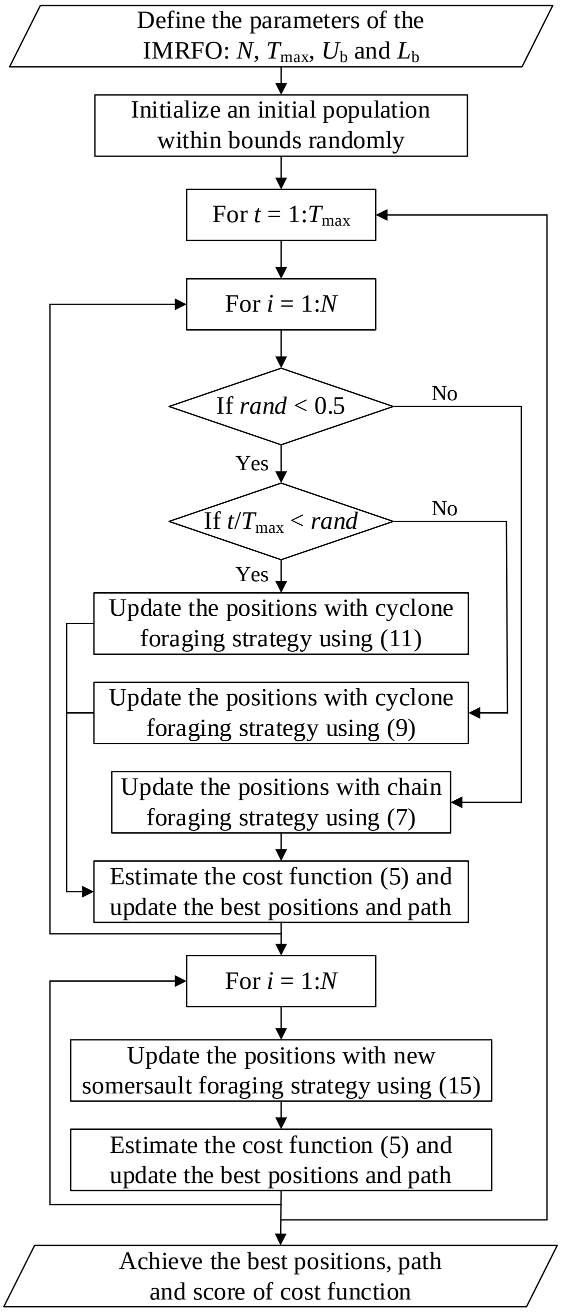


Fig. 2 Procedure of path planner based on IMRFO.

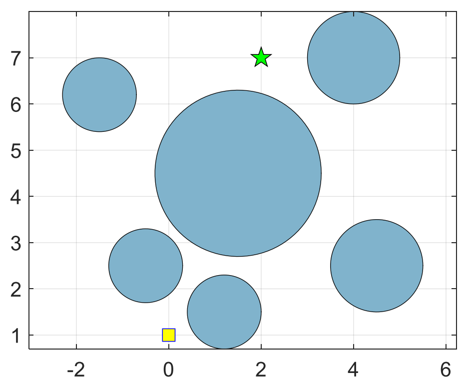
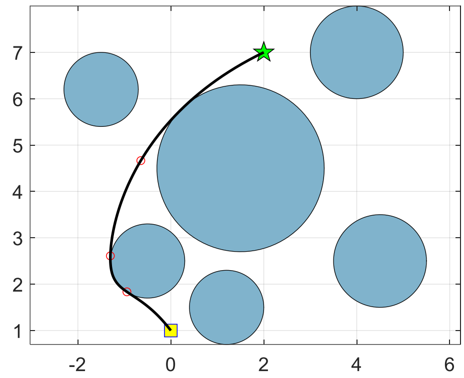


Fig. 3 Working environment of the mobile robot.

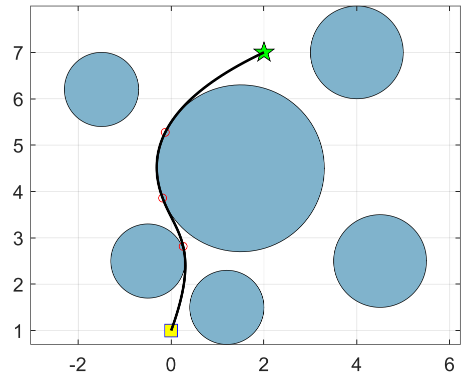
Table 1. The statistical results of the three algorithms.

	PSO	MRFO	IMRFO
Mean	8.0772	7.4444	7.2567
Optimal	7.8832	7.2276	7.2275
Worst	8.8507	7.8975	7.8732
Std	0.3131	0.3077	0.1285

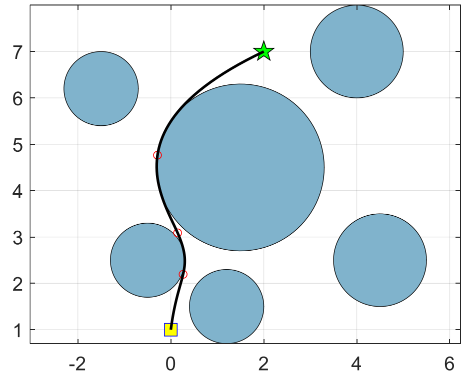
Fig. 4 shows the path planning results of PSO, MRFO and IMRFO, respectively. The results planned by the three algorithms all satisfy the path planning requirements. But the length of the path planned by PSO is longer than the path found by MRFO and IMRFO. The path planners based on the MRFO and IMRFO have better performance in length accuracy than PSO, while the path planning result of PSO falls into local optimum.



(a) Path planned by PSO



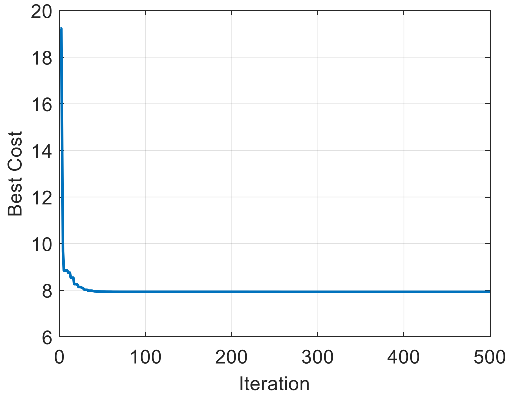
(b) Path planned by MRFO



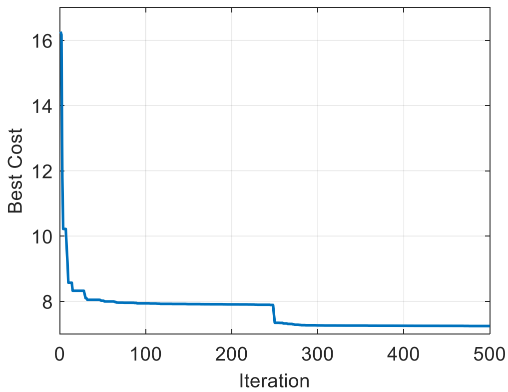
(c) Path planned by IMRFO

Fig. 4 The comparative path planning results.

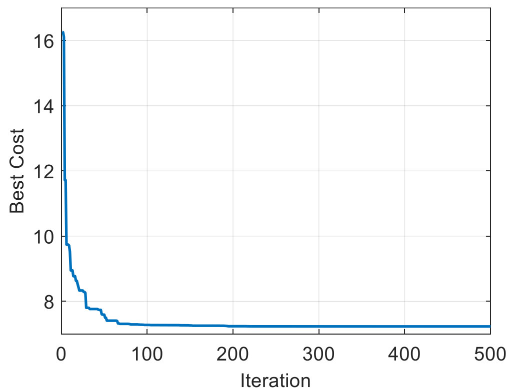
The convergence curves of path planning results corresponding to **Fig. 4** are shown in **Fig. 5**. The PSO attain its optimal value in iteration 40, which is the fastest convergence speed among the three algorithms. However, it is a bad performance for PSO to fall into local optimum with the fastest convergence speed. The MRFO and IMRFO find their optimal value in iteration 285 and 194 respectively. Although both MRFO and IMRFO find a shorter length of path finally, IMRFO has a better performance in convergence speed than MRFO.



(a) Convergence curve of PSO



(b) Convergence curve of MRFO



(c) Convergence curve of IMRFO

Fig. 5 The comparative convergence curves.

With independent experiments in 25 times, the statistical results are illustrated **Table 1,** and the comparative cost values are shown in **Fig. 6**. The statistical result of the IMRFO owns the minimum mean, optimal, and worst fitness values. Meanwhile, the standard deviation of the IMRFO is also the smallest, which indicates that the IMRFO has the best stability to search for its optimal path among the three algorithms. To show the stability for search global optimum more concretely in **Fig. 6,** the cost values of PSO are always stuck into the local optimum, while MRFO fails to find its optimal value for 8 times. Compared with the results of PSO and MRFO, the IMRFO just fails to find its optimal value for 1 time, and always maintains the global optimal value with a better algorithm stability.

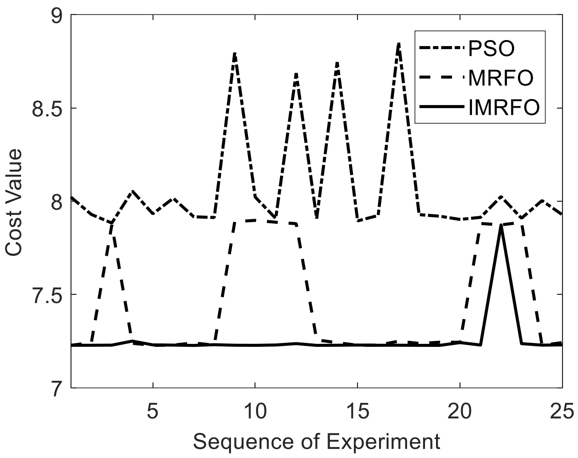


Fig. 6 The comparative cost values of the independent experiments in 25 times.

In summary, from the above experimental results, IMRFO is effective to find an optimal path and has better performance in convergence speed, accuracy and stability compared with PSO and MRFO.

4. CONCLUSION

The experimental results show that IMRFO, MRFO and PSO all find a feasible path with obstacles avoidance. However, PSO converges to the local optimum with the fastest speed among the three algorithms in a useless way. Meanwhile, IMRFO finds the global optimal path with a better convergence speed and accuracy than MRFO. Moreover, IMRFO finds the global optimal path more stably in 25 times independent experiments than MRFO and PSO.

Compared to the experimental results of PSO and MRFO, IMRFO not only has the ability to find an effective path with shortest length of path and no collision, but also has better performance than MRFO and PSO in dealing with the path planning problem. So the IMRFO with Cauchy mutation has an enhanced ability of exploration compared with MRFO. This enhanced exploration enables IMRFO to jump out of the local optimal path and increases the search accuracy, speed and stability of IMRFO for the global optimal path.

Our initial target is achieved with the superior performance of IMRFO for path planning than the basic MRFO and PSO. The proposed IMRFO provides a powerful tool in dealing with mobile robot path planning problem. From the results of this paper, applying the proposed method to rescue robot in dangerous areas will be opened from now on. Dangerous areas are difficult to be described with the specific shape, but convenient to be represented by circles with estimated radii. And rescue robot needs to find the optimal path from its position to the rescue target. The proposed method meets the requirement of rescue robot mentioned above, which contributes to the effective accomplishment of rescue missions.

Acknowledgements

This work is supported by the State Recruitment Program of Global Experts under Grant WQ20141100198.

REFERENCES:

[1] P. B. Kumar, H. Rawat, and D.R. Parhi, “Path planning of humanoids based on artificial potential field method in unknown environments,” Expert Systems, vol. 36(2), pp. 12360, 2019.

[2] T. Q. Toan, A. A. Sorokin, and V. T. H, “Trang. Using modification of visibility-graph in solving the problem of finding shortest path for robot,” 2017 International Siberian Conference on Control and Communications (SIBCON), IEEE, pp. 1-6, 2017.

[3] L. Zhang, Z. Lin, J. Wang J, and B. He, “Rapidly-exploring Random Trees multi-robot map exploration under optimization framework,” Robotics and Autonomous Systems, pp. 103565, 2020.

[4] X. Dai, S. Long, Z. Zhang, and D. Gong, “Mobile robot path planning based on ant colony algorithm with A* heuristic method,” Frontiers in neurorobotics, vol. 13, pp. 15, 2019.

[5] M. A. Hossain, and I. Ferdous, “Autonomous robot path planning in dynamic environment using a new optimization technique inspired by Bacterial Foraging technique,” Robotics & Autonomous Systems, vol. 64, pp. 137-141, 2015.

[6] H. Mo, and L. Xu, “Research of biogeography particle swarm optimization for robot path planning,” Neurocomputing, vol. 148, pp.91-99, 2015.

[7] X. Wang, G. Zhang, J. Zhao, H. Rong, F. Ipate, and R. Lefticaru, “A modified membrane-inspired algorithm based on particle swarm optimization for mobile robot path planning,” International journal of computers communications & control, vol. 10(5), pp. 732-745, 2015.

[8] L.Wang, J. Guo, Q. Wang, and J. Kan, “Ground robot path planning based on simulated annealing genetic algorithm,” 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, pp. 417-424, 2018.

[9] Y. Xue, “Mobile robot path planning with a non-dominated sorting genetic algorithm,” Applied Sciences, vol. 8(11), pp. 2253, 2018.

[10] X. Liang, D. Kou, and L. Wen, “An Improved Chicken Swarm Optimization Algorithm and its Application in Robot Path Planning,” IEEE Access, vol. 8, pp. 49543-49550, 2020.

[11] D. H. Wolpert, and W. G. Macready, “Coevolutionary free lunches,” IEEE Transactions on evolutionary computation, vol. 9(6), pp.721-735, 2005.

[12] W. Zhao, Z. Zhang, and L. Wang, “Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications,” Engineering Applications of Artificial Intelligence, vol. 87, pp.103300, 2020.

[13] S. I. Selem, H. M. Hasanien, and A. A. El-Fergany, “Parameters extraction of PEMFC's model using manta rays foraging optimizer,” International Journal of Energy Research, vol. 44(6), pp.4629-4640, 2020.

[14] M. A. El-Hameed, M. M. Elkholy, and A. A. El-Fergany, “Three-diode model for characterization of industrial solar generating units using Manta-rays foraging optimizer: Analysis and validations,” Energy Conversion and Management, vol. 219, pp.113048, 2020.

[15] J. Yang, and L. Li, “Improved biogeography-based optimization algorithm for mobile robot path planning,” Chinese Intelligent Systems Conference, Springer, Singapore, pp. 219-229, 2017.

[16] X. Li, D. Wu, J. He, M. Bashir, and M. Liping, “An Improved Method of Particle Swarm Optimization for Path Planning of Mobile Robot,” Journal of Control Science and Engineering, pp. 1-12, 2020.