Biomimetic Intelligence and Robotics 1 (2021) 100003

Contents lists available at ScienceDirect

Biomimetic Intelligence and Robotics

journal homepage: www.elsevier.com/locate/birob

Landing route planning method for micro drones based on hybrid optimization algorithm

Liang Shaoran, Song Bifeng, Xue Dong *

School of Aeronautics, Northwestern Polytechnical University, Xi'an, China

ARTICLE INFO

Keywords:
Route planning
Micro drones
Dragonfly optimization
Hybrid optimization

ABSTRACT

Aiming at the obstacle avoidance trajectory planning problem in the landing process of the micro drone, this paper proposes a swarm optimization algorithm that combines the dragonfly optimization method and the differential evolution method. The orthogonal learning mechanism is adopted to realize the adaptive switch between the two algorithms. In the process of landing route planning, the planning plane is first obtained by making the gliding plane tangent to the obstacle. In the planning plane, the projection of obstacle is transformed into multiple unreachable line segments. By designing an optimization model, the 3D landing route planning problem is transformed into a 2D obstacle avoidance route optimization problem. Taking the shortest route as the optimization objective, the penalty factor is introduced into the cost function to avoid the intersection of the landing route and obstacle. During the optimization process, through the orthogonal learning of the intermediate iterative results, the hybrid algorithm can adaptively select the next iterative algorithm, so it can give full play to the respective advantages of the two algorithms. The optimization results show that, compared with the single optimization algorithm, the hybrid optimization algorithm proposed in this paper can better solve the problem of landing route planning for micro-small UAVs.

1. Introduction

Micro drones can generally be used to perform reconnaissance missions in urban building environments. In order to improve the efficiency of reconnaissance and surveillance, UAVs are usually required to land near the target area for long-term surveillance. Landing safely in the target area is a prerequisite for the successful execution of such tasks. Generally speaking, through satellite remote sensing technology and high-altitude unmanned aerial vehicle reconnaissance technology and other means, the obstacle information in the landing area, such as buildings and hillsides, can be predicted in advance. Utilizing the priori information of obstacles to plan the global path of the landing trajectory, it can effectively improve the landing safety of the UAV.

For the path planning problem, many scholars have conducted a lot of research. In order to solve the path planning problem of autonomous underwater vehicles, Yan et al. [1] proposed an improved water wave optimization algorithm based on the elite opposition-based learning strategy and the simplex method. The simulation results show that the overall performance of the improved algorithm is better than other algorithms, and it is an effective and feasible method to solve the problem of underwater vehicle path planning. Guo et al. [2] proposed an improved quantum-behaved particle swarm algorithm to solve the problem of automatic guided vehicle path planning. The algorithm has achieved satisfactory results in solving path planning problems under multi-objective constraints. Chen et al. [3] proposed an improved particle swarm algorithm based on fuzzy-logic to solve the problem of UAV path planning in three-dimensional space. By introducing fuzzy processing to avoid falling into local optimum, the effectiveness of the method is verified by simulation. Xu et al. [4] proposed a dynamic firefly algorithm with adaptive parameters in order to speed up route planning. Through the optimization strategy in the algorithm, the defect that the standard firefly algorithm is easy to fall into the local optimum is solved. The application of this method in two-dimensional route planning has achieved good results. Wu et al. [5] proposed an improved genetic algorithm for two-dimensional route planning. This method dynamically adjusts the crossover probability and mutation probability through an adaptive operator to avoid falling into the local optimum. The several improved algorithms mentioned above have shown satisfactory results in their respective application fields, and compared with the original algorithm, the performance of the algorithm itself has also been improved. However, these methods cannot be directly applied to the problem of UAV landing route planning. This is because in the process of landing, the drone must not only consider reasonably avoiding obstacles on the route, but also consider the limitations of the drone's own flight performance, such as the maximum climb angle and gliding angle, the maximum turning angle, and the

* Corresponding author at: School of Aeronautics, Northwestern Polytechnical University, Xi'an, China.
E-mail address: xuedong@nwpu.edu.cn (D. Xue).

https://doi.org/10.1016/j.birob.2021.100003

Available online 15 May 2021

minimum route length. Only routes that meet multiple constraints at the same time can become flyable routes.

The genetic optimization algorithm can solve practical problems more efficiently only when a good balance is achieved between the global exploration performance and the local exploitation performance. The hybridization research between different algorithms is one of the effective means to improve the performance of the algorithm by taking into account the global and local optimization capabilities. The hybrid of the two algorithms can make full use of the advantages of each method, improve the global convergence speed, and avoid prematurely falling into the local optimum. For example, literature [6] proposed an optimization method combining particle swarm algorithm and gravitational search algorithm, and achieved good results in multi-robot route planning. Literature [7] proposed a hybrid whale optimization algorithm with simulated annealing method, and conducted a more comprehensive test on the effectiveness of the hybrid algorithm. Literature [8] proposed an optimization method that hybridize gray wolf algorithm and symbiotic organisms search, and analyzed the convergence of the hybrid algorithm through linear difference theory. The simulation results showed that the hybrid algorithm can obtain a feasible path and has excellent performance.

Among many heuristic optimization algorithms, the dragonfly optimization algorithm [9], as a swarm intelligence optimization algorithm that simulates the biological habits of nature, has been successfully applied in many fields [10–12]. This algorithm takes into account both the dynamic and static flight behavior of dragonflies and has strong local exploitation capabilities. The differential evolution algorithm [13] is an optimization method based on individual evolution and competitive behavior in nature, and has strong global exploration capabilities. Therefore, in order to give full play to the respective advantages of the above two algorithms, this paper proposes an adaptive hybrid method based on the orthogonal learning mechanism [14]. This method can be used to switch the dragonfly algorithm and the differential evolution algorithm according to the iterative results during the solution process. So it can take into account global exploration and local exploitation performance. In this paper, the hybrid algorithm is applied to the UAV landing route planning problem and satisfactory results are achieved.

2. Problem analysis and optimization model design

In order to accomplish a long-term reconnaissance mission, micro drones need to land in advance and hide in the target area. In the research of this article, it is assumed that there are obstacles such as buildings and hillsides on the landing route. In order to optimize the landing route, it is necessary to build a flight environment model. Fig. 1 is a simplified schematic diagram of the flight environment. First, according to the prior information of obstacles (such as buildings, etc.), they are processed into cubes, and then a straight line is obtained by connecting the starting point A and ending point B of the landing route, and making a straight line BC on the ground, making the straight line AB and the straight line BC perpendicular to each other. Two straight lines can determine the gliding plane O. During the landing process, the drone is always on the gliding plane. The plane O intersects the obstacle to obtain the cross section of the obstacle. Fig. 2 is a simplified plan view obtained after the gliding plane is intersected with the obstacle. Optimizing the trajectory on the planning plane can transform the 3D route planning problem into the obstacle avoidance problem on the 2D plane. In order to make the obstacle avoidance problem specific, the coordinate system needs to be defined in Fig. 2. Taking the starting point A as the origin of the coordinates, rotate the straight line AB by a fixed angle to obtain the X axis, and the perpendicular line passing the point A on the gliding plane is defined as the Y axis direction.

In Fig. 2, vertical lines of line AB are made through each vertex of the obstacle. The part where the vertical line intersects the obstacle is defined as the unreachable position. Waypoints are arranged on the vertical lines. The starting point A, each waypoint and the end point B can be connected to obtain the initial target route.

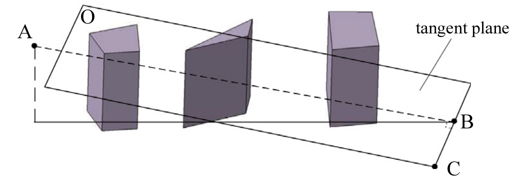

Fig. 1. A simplified schematic diagram of the flight environment.

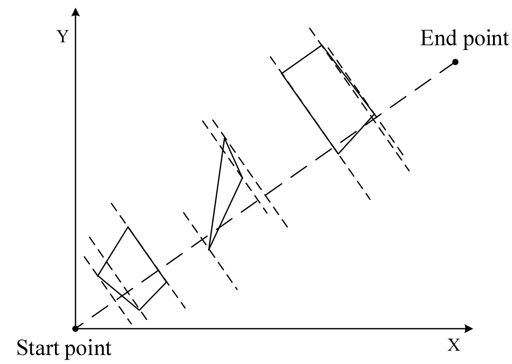

Fig. 2. A simplified plan view obtained after the gliding plane is intersected with the obstacle.

In the optimization process, not only the reasonable avoidance of obstacles must be considered, but also the influence of its own flight performance on the route. There are many constraints that affect the flight path of UAVs. This article mainly considers the following aspects. (1) Maximum gliding angle limit. During the landing process, in order to prevent the aircraft from stalling, the gliding angle of the route cannot exceed the maximum gliding angle of the aircraft. In the research of this article, the target waypoint is located in the glide plane designed above. By adjusting the position of the landing start point, the angle between the glide surface and the horizontal plane can be controlled, so that the gliding angle limit can be met at the early stage of planning. (2) Maximum turning angle limit. In a segmented route, when the drone switches from a certain route to the next, it will temporarily rush out of the route. Under the action of the controller, it gradually enters the next flight course. The time to enter the next flight path and the distance to rush out are not only related to the performance of the controller, but also to the maneuverability of the aircraft. If the angle between the two flight segments is too small, it will cause the aircraft to maneuver too much and reduce the flight safety. Therefore, the turning angle of the route should be less than the maximum allowable turning angle. Assuming that the current route segment is i and the maximum turning angle is $\theta_{\max}$, the following conditions need to be met:

$$\cos(\theta_{\max}) \le \frac{a_i^T a_{i+1}}{\|a_i\| \cdot \|a_{i+1}\|} (i = 1, 2, \dots, D) \tag{1}$$

where D is the total number of route segments and a_i is the vector of the first route segment.
(3) The shortest route limit. In a route composed of waypoints, the distance between two waypoints cannot be too close. This limitation is not only related to the flight performance of the aircraft itself, but also related to the capabilities of the navigation algorithm. Generally, it is expected that UAVs can fly long distances in a straight line. Assuming that the length of the shortest route segment is $l_{\min}$, the following conditions need to be met:

$$l_i \ge l_{\min} \tag{2}$$

The total length of the landing route is the sum of the length of each route segment. Taking the shortest total route length as the optimization objective, it can be expressed as:

$$W = \sum_{i=i}^{n-1} L(P_i, P_{i+1}) + L(P_A, P_1) + L(P_n, P_B) \tag{3}$$

where $L(P_i, P_{i+1})$ is the straight-line distance between the ith waypoint and the $i+1$th waypoint. $L(P_A, P_1)$ is the distance between the start point and the first waypoint, and $L(P_n, P_B)$ is the distance between the last waypoint and the end point. Suppose the coordinate of the ith waypoint P_i is (x_i, y_i).

The distance between two waypoints is $L(P_i, P_{i+1}) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - \sigma y_i)^2}$, where σ is the penalty factor. If P_i is in the reachable range, the value is 1. If P_i is in the unreachable range, the value is a positive number with a large value. By introducing a penalty factor, the waypoint in the iteration process can always be located outside the obstacle.

3. Dragonfly optimization-differential evolution hybrid algorithm

3.1. Dragonfly optimization algorithm

The dragonfly optimization algorithm is a population-based optimization algorithm [9], which simulates the static and dynamic flight behavior of the dragonfly. The behaviors of dragonfly populations in nature during flight include multiple behavior modes such as group flight in the same direction, predation behavior, and avoiding natural enemies. The dragonfly optimization algorithm mathematically models five types of behaviors, which are: separation behavior, alignment behavior, cohesion behavior, attraction to food, and distraction from enemy. The specific meaning of the algorithm is as follows:

(1) Separation behavior

This behavior describes keeping a certain distance between dragonfly individuals to prevent collisions. This behavior can be expressed by the following model.

$$S_i = -\sum_{j=1}^{N} X - X_j \tag{4}$$

where S_i represents the position vector generated by the separation behavior of the i th dragonfly individual. X represents the spatial position of the current individual, and N represents the number of other dragonflies within the neighborhood radius of the current individual. X_j indicates the spatial position of the jth dragonfly within the radius of the neighborhood.

(2) Alignment behavior

This behavior describes that the flight speeds between adjacent individuals tend to be consistent, which can be described by the following model:

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{5}$$

where A_i represents the position vector of the i th dragonfly individual due to the alignment behavior. V_j indicates the flight speed of other individuals within the radius of the neighborhood.

(3) Cohesion behavior

This behavior describes the tendency of individuals to move toward the center of the neighborhood, and can be described by the following model.

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \tag{6}$$

where C_i represents the position vector of the ith dragonfly individual due to the cohesion behavior.

(4) Attraction to food

This behavior describes the tendency of the individual to move to the position of the food, which can be described by the following model.

$$F_i = X^+ - X \tag{7}$$

where F_i is the position vector generated by attraction to food behavior. X^+ indicates the location of the prey.

(5) Distraction from enemy

This behavior describes an individual's instinct to avoid natural enemies, which can be described by the following model.

$$E_i = X^- - X \tag{8}$$

where E_i represents the position vector generated by the individual due to avoiding natural enemies. X^- indicates the location of natural enemies.

Through the expression of five flight behavior above, the position of the next generation individual can be updated by the following formula.

$$\begin{cases} \Delta X_{i+1} &= \left(sS_i + aA_i + cC_i + fF_i + eE_i\right) + w\Delta X_i \\ X_{i+1} &= X_i + \Delta X_{i+1} \end{cases} \tag{9}$$

where s, a, c, f, e represent separation factor, alignment factor, cohesion factor, food factor and enemy factor, respectively. w represents the inertia factor, X_i represents the position of the current individual in the population, and X_{i+1} represents the position of the next generation of individuals. ΔX_{i+1} represents the update step size when calculating the position of the next generation individual. Define the neighborhood radius of the dragonfly population as r, when the distance between two individuals is less than r, the two individuals are considered adjacent individuals. In order to speed up the convergence speed of the algorithm, r increases as the number of iterations increases until all individuals are located in the same neighborhood.

3.2. Differential evolution algorithm

Differential evolution algorithm is an optimization algorithm that simulates the competition and evolution of individuals in nature. Each individual is composed of several genes, and the number of genes is consistent with the dimension of the problem to be solved. The algorithm gradually iterates to obtain the optimal individual through mutation, crossover and selection operations. The mutation operation refers to randomly selecting individuals in the population as the basic genes, and then selecting different individuals to make differences to obtain differential genes. Combine the differential gene with the basic gene to obtain the mutated gene. Crossover operation refers to the crossover operation between the mutated gene and the parent gene to obtain the crossover gene. Then select the best to determine whether to retain the new gene after crossover to the next generation, or to keep the parent gene unchanged. The specific meaning of the algorithm is as follows.

(1) Mutation operation

Randomly select 3 unique individuals r_1, r_2, r_3, and mutate a new individual U_i based on these three individuals. The individual after the mutation operation is:

$$U_i = X_{r1} + F(X_{r2} - X_{r3}) \tag{10}$$

where F is the scaling factor, and the value range is [0,1].

(2) Crossover operation

Cross operation between the current individual X_i and U_i.

$$v_i^j = \begin{cases} U_i^j, R_j \le CR \\ X_i^j, R_j > CR \end{cases} \tag{11}$$

where v_i^j is the jth gene after the crossover operation, CR is the crossover probability, and the value range is [0,1]. It is a random number between 0 and 1.

(3) Select operation

The selection operation refers to selecting individuals with better cost function values between the parent individuals and the offspring individuals after the mutation and crossover operation. The selection operation can be expressed as:

$$X_i = \begin{cases} U_i', f(U_i') \le f(X_i) \\ X_i, f(U_i') > f(X_i) \end{cases} \tag{12}$$

3.3. Dragonfly optimization-differential evolution hybrid algorithm

With the help of orthogonal design theory [14], this paper proposes an optimization method that combines dragonfly algorithm and differential evolution algorithm. Orthogonal design is generally used in multi-factor and multi-level experiments. When there are many influencing factors in the experiment, if a comprehensive experiment is carried out, it will cause too many experiments and the experiment cost will be too high. Orthogonal design method can achieve the purpose of obtaining sufficient information while reducing the number of experiments through reasonable selection of experimental factors and experimental levels. The steps of the hybrid optimization algorithm are as follows:

(1) Initialize the selection factor η between [0,1];

(2) Initialize the maximum number of iterations. Initialize the individual dimension and the total population. Initialize the position of the individual and calculate the cost function of the current individual;

(3) Each individual carries a random number between [0,1] and initializes the random number. When the random number is less than η, the current individual belongs to the population a, and when the random number is greater than or equal to η, the current individual belongs to the population b;

(4) In the population a, the dragonfly algorithm is used to perform one-step iteration, and the best individual in the current population is selected as p_{a1}, and the second best individual is p_{a_2}, and orthogonal operations are performed on p_{a1} and p_{a_2} to obtain p_a. In the population b, a differential evolution algorithm is used for one step iteration. Select the best individual in the current population as p_{b1}, and the next best individual as p_{b2}, and perform orthogonal operations with p_{b1} and p_{b2} to obtain p_b. The orthogonal operation of p_a and p_b gets p.

(5) For the p, the number of variables from p_a is z_a, and the number of variables from p_b in is z_b, and the selection factor is recalculated $\eta = \frac{z_a}{z_a+z_b}$.

(6) Combine the population a and the population b into a population, and judge whether the optimal individual in the population satisfies the multiple constraints mentioned above.

(7) Jump to step (3) until the maximum number of iterations is reached or the accuracy requirements are met.

The calculation flow chart of the hybrid algorithm is shown in Fig. 3.

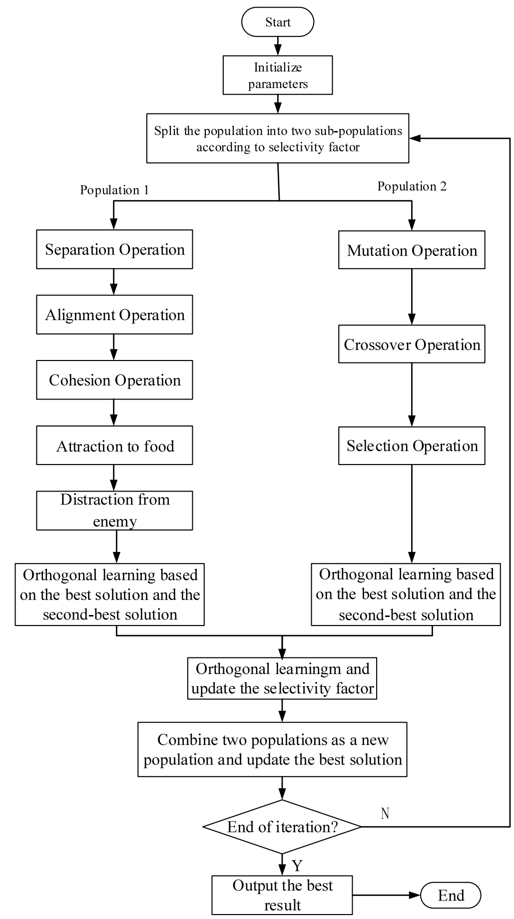

Fig. 3. The calculation flow chart of the hybrid algorithm.

4. Simulation

Through the environmental model preprocessing method proposed in the Part 2, the three-dimensional landing trajectory planning problem can be transformed into a two-dimensional plane path optimization problem. Fig. 4 shows a plan view of the gliding plane tangent to the obstacle. In Fig. 4, a vertical line perpendicular to the line connecting the starting point and the ending point is made through the apex of the obstacle, and the result in Fig. 5 is obtained. In the two-dimensional plane, each line segment represents the part where the perpendicular line passing the apex of the obstacle intersects the obstacle, which is the unreachable area. This article assumes that the flying range of the UAV is 1600 m × 1600 m. In Fig. 5, the coordinates of the end point B are $(1600, 1600)$. In order to verify the effectiveness of the proposed method, this paper conducts path planning simulation research based on the preprocessing results shown in Fig. 5.

In the planning diagram after preprocessing, the mathematical analytic formula of the vertical line can be obtained according to the coordinates of the obstacle vertex and the slope of the vertical line segment, and the mathematical expression of the unreachable part can be obtained by combining the boundary of the obstacle. The number of line segments in the planning diagram is the dimensionality of the individuals in the population.

In order to verify the performance of the hybrid method, this paper compares and analyzes the single dragonfly optimization algorithm and the differential evolution algorithm. The initial environment parameters in the simulation verification are the same. Fig. 6 shows the comparison result of the hybrid algorithm and two single optimization algorithms.

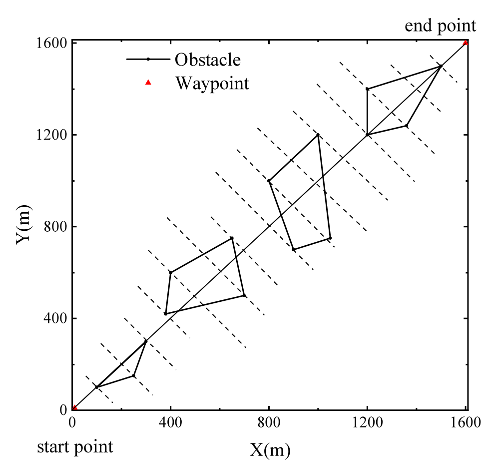

Fig. 4. A vertical line perpendicular to the line connecting the starting point and the ending point is made through the apex of the obstacle.

It can be seen from Fig. 6 that after iterative calculation, the three algorithms can obtain the optimal path. The final results are shown in Table 1. According to the final path length, the result of the hybrid algorithm is better than the other two independent algorithms.

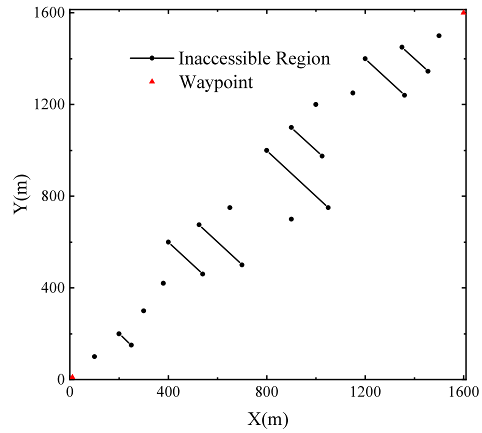

Fig. 5. Path planning simulation research based on the preprocessing results.

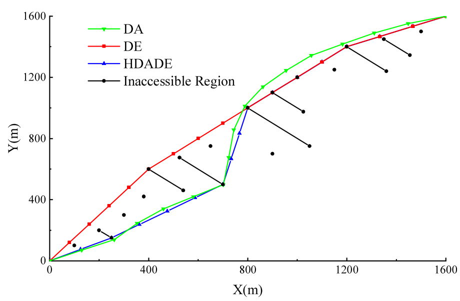

Fig. 6. Comparison result of the hybrid algorithm and two single optimization algorithms.

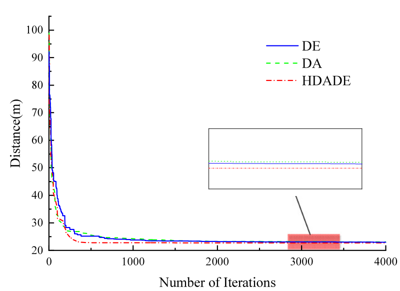

Fig. 7. The convergence curves of the three algorithms.

Table 1
Optimal path obtained by three algorithms.

	Optimal path (m)
DE	2383.9
DA	2385.2
HDADE	2364.4

Fig. 7 shows the convergence curves of the three algorithms. In the initial stage of the iteration, the dragonfly optimization algorithm converges better and faster than the differential evolution algorithm. The hybrid algorithm after orthogonal learning has a similar convergence speed with the dragonfly optimization algorithm in the initial stage. When the number of iterations is about 500 times, the hybrid algorithm obtains the optimal result. The iteration times of the other two methods are both greater than 1000 times. From the final result, the hybrid algorithm is also better than the other two algorithms. Therefore, after the introduction of the orthogonal learning mechanism, the hybrid algorithm has stronger global and local optimization capabilities than a single algorithm.

5. Conclusion

In this paper, an intelligent optimization method combining dragonfly optimization algorithm and differential evolution algorithm is realized through orthogonal learning mechanism. This method is applied to the landing route planning problem, and the UAV obstacle avoidance route design is completed. Compared with a single algorithm, the hybrid method has a faster convergence speed in the initial stage of the iteration, and the final calculation is better than the single algorithm. It needs to be pointed out that the results obtained in this paper are all obtained in a simulation environment, and it is one of the directions of future work to extend the current design results to the real environment.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was supported by National Natural Science Foundation of China, Grants No. 11902103 and 11572255 and China Postdoctoral Science Foundation Grant 2019M651294.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.birob.2021.100003.

References

[1] Z. Yan, J. Zhang, J. Tang, Path planning for autonomous underwater vehicle based on an enhanced water wave optimization algorithm, Math. Comput. Simulation 181 (2021) 192–241.
[2] X. Guo, M. Ji, W. Zhang, AGV Global path planning integrating with the control of multi-objectives and speed, Control Decis. 35 (2020) 1369–1376, (in Chinese).
[3] T. Chen, Y. Wang, Q. Wu, Z. Zhou, Three-dimensional path planning based on fuzzy-logic particle swarm optimization, Electron. Opt. Control 27 (2020) 1–5, (in Chinese).
[4] G. Xu, T. Zhang, Q. Lai, J. Pan, B. Fu, X. Zhao, A new path planning method of mobile robot based on adaptive dynamic firefly algorithm, Modern Phys. Lett. B 34 (2020) 2050322.
[5] M. Wu, E. Chen, Q. Shi, L. Zhou, Z. Chen, M. Li, Path planning of mobile robot based on Improved Genetic Algorithm, in: Proceeding of 2017 Chinese Automation Congress (CAC 2017), pp. 6696-6700.
[6] P.K. Das, H.S. Behera, B.K. Panigrahi, A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning, Swarm Evol. Comput. 28 (2016) 14–28.
[7] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.
[8] C. Qu, W. Gai, J. Zhang, M. Zhong, A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning, Knowl.-Based Syst. 194 (2020) 105530.
[9] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (2016) 1053–1073.
[10] C.M. Rahman, T.A. Rashid, A. Alsadoon, N. Bacanin, P. Fattah, A survey on dragonfly algorithm and its applications in engineering, Pub Date : 2020-02-19, doi: arxiv-2002.12126.
[11] S.R. Ks, S. Murugan, Memory based hybrid dragonfly algorithm for numerical optimization problems, Expert Syst. Appl. 83 (2017) 63–78.
[12] S. Mirjalili, A. Lewis, Novel performance metrics for robust multi-objective optimization algorithms, Swarm Evol. Comput. 21 (2015) 1–23.
[13] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.
[14] Z. Zhan, J. Zhang, Y. Li, Y. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (2011) 832–847.