HPSO-WOA-SFO: A Novel Hybrid Swarm Intelligence Approach for Enhancing Discrete Road Path Planning

You Wu[1], Xi Hu[2], and Guosheng Zhu[1](✉)

[1] School of Cyber Science and Technology, Hubei University, Wuhan 430062, China
zhuguosheng@hubu.edu.cn

[2] School of Artificial Intelligence, Jianghan University, Wuhan 430056, China

Abstract. The Hybrid Particle Swarm Optimization-Whale Optimization Algorithm-Sailfish Optimizer (HPSO-WOA-SFO) is proposed for solving multi-obstacle discrete road path planning. This paper proposes to utilize the advantage of the two-population update iteration of the sailfish algorithm to integrate the PSO and WOA into the SFO to enhance its exploitation ability and exploration ability, respectively. Meanwhile, the two communication mechanisms between the two populations of the SFO are studied in depth, and their algorithmic advantages and application scenarios are analyzed. Comparative experiments with four representative path planning algorithms and ablative experiments involving HPSO-WOA-SFO are conducted. The results demonstrate that, on average, HPSO-WOA-SFO outperforms the comparative algorithms by 21.40% in terms of global optimal convergence accuracy and is 10.71% faster in terms of convergence speed. Moreover, the proposed algorithm rapidly escapes local optima and enhances global optimality by 17.47% when trapped in local optima during the optimization process.

Keywords: Evolutionary Algorithm · Discrete Space Optimization Problem · Sailfish Optimization Algorithm · UAV Path Planning

1 Introduction

Today, the civilian application areas of UAVs are rapidly expanding, including real-time surveillance, providing wireless coverage, remote sensing, search and rescue, cargo transportation, security and surveillance, precision agriculture, and

This work was supported in part by the National Natural Science Foundation of China under Grant 61901298; in part by the Young Talents Science and Technology Innovation Planning Program of Education Department of Hubei Province under Grant T2022045; in part by the Scientific Research Project of Education Department of Hubei Province under Grant B2022280; in part by the Scientific Research Foundation of Jianghan University under Grant 2023KJZX18.

X. Chen et al. (Eds.): IoTaaS 2023, LNICST 585, pp. 385–395, 2025.
https://doi.org/10.1007/978-3-031-70507-6_29

civil infrastructure inspection [1]. One of the most important problems that need to be explored is the UAV path planning problem, which aims to find the optimal path between the source and the destination [2]. The unmanned aerial vehicle path planning problem is regarded as a highly complex global optimization problem with NP-hard characteristics [3], Therefore, employing metaheuristic algorithms to thoroughly search the discrete road space, aiming to discover the optimal and fastest paths, has become one of the predominant approaches for addressing this issue.

The key to utilizing metaheuristic algorithms for solving the discrete road path planning problem lies in their ability to efficiently search for optimal paths. Among the mainstream optimization algorithms are several noteworthy ones, including Genetic Algorithm [4], Particle Swarm Optimization (PSO) [5], Whale Optimization Algorithm (WOA) [6], and Grey Wolf Optimizer (GWO) [7]. The Sailfish Optimizer (SFO) [8] is also among these algorithms. Notably, the Sailfish Optimizer is considered a highly promising optimization algorithm and has garnered significant attention from researchers. This algorithm features the dual-population search characteristic, enabling it to simultaneously perform exploration and exploitation during iterations. It also incorporates an effective population communication mechanism. As a result, when compared to traditional single-meta heuristic algorithms, the Sailfish Optimizer exhibits various advantages, including high iteration efficiency, strong flexibility, and remarkable scalability.

In addition to its applications in road path planning, metaheuristic algorithms have been employed in various other domains, such as forecasting residential electricity consumption. One such example is the hybrid optimized grey seasonal variation index model [9]. This model has demonstrated enhanced accuracy in predicting residential electricity consumption. Furthermore, in the context of hydropower generation prediction, metaheuristic algorithms have played a crucial role. A novel Optimized Grey Seasonal Variation Index model [10] has been proposed, leveraging the power of metaheuristic optimization to improve forecasting accuracy. Additionally, for the prediction of hydropower generation, a novel Weighted Average Weakening Buffer Operator [11] has been developed, incorporating metaheuristic techniques for improved performance. In the realm of online social networks, a hybrid clustered SFLA-PSO algorithm [12]has been introduced, showcasing the versatility of metaheuristic algorithms in tackling real-world problems".

Recently, various enhanced applications of SFO have been published in journals across different fields. The scalability of the Sailfish Optimizer has been fully realized. For instance, researchers have proposed improved SFO algorithm models with three distinct characteristics dynamic discrete, dynamic continuous, and static continuous based on the varying nature of berth allocation problems for ships, in order to compare their advantages and disadvantages [13]. Moreover, the Sailfish Optimizer has been hybridized with the Grey Wolf Optimizer and applied to the electric vehicle charging station scheduling problem in urban settings, yielding significant optimization outcomes for a substantial number of vehicles [14]. As well, in the feature selection area some researchers proposed to map the continuous solution space of the SFO algorithm to the BSFO in binary space using the Sigmoid transformation function, and also mixed it with the AβHC algorithm to become the ASF algorithm to enhance its development capability [15].

This paper proposes a Hybrid Particle Swarm Optimization - Whale Optimization Algorithm - Sailfish Optimizer (HPSO-WOA-SFO) to solve UAV discrete road path planning. By integrating PSO and WOA with two populations, the SFO algorithm's exploitation and exploration capabilities are effectively enhanced. Moreover, in order to improve the population communication efficiency, this study analyzes the update and exchange process of the two populations in the SFO algorithm, adapting it to the path planning problem addressed. Then, the experimental results of HPSO-WOA-SFO in this paper show that HPSO-WOA-SFO is 21.40% better than the comparison algorithm in the global optimal convergence accuracy on average.

The remaining sections of this paper are organized as follows. Section 2 describes a hybrid PSO-SFO and analyze its principles. Section 3 describes and analyzes the hybrid WOA-SFO formulation and principles. Section 4 describes the application process of the above two hybrid algorithms in the dual population and discusses the characteristics of node update mechanism. Section 5 summarizes the algorithm design of HPSO-WOA-SFO and the corresponding experimental results.

2 Enhancing Population Exploitation with Hybrid PSO-SFO Algorithm

The sailfish optimization algorithm uses a sailfish population and a sardine population for exploitation iterations and exploration iterations [8] respectively. Where the purpose of the sailfish population is to find better values around the elite sailfish nodes to replace the elite sailfish nodes, thus acting as a exploitation population. Therefore, this paper applies the PSO's historical optimal node and current optimal node's to the current node iteration of the sailfish swarm. The node iteration of the sailfish population is extended from iterating with the current 'elite sailfish' (the best node in the sailfish population) and 'injured sardine' (the best node in the sardine) as parameters to iterating with its own historical optimum, 'elite sailfish', 'injured sardine' as parameters for iteration, which can maximize the use of each iteration of each node has an advantage in the path information, The iterative process is shown in Eq. (1) below:

$$
\begin{aligned}
N^i_{new_{SF}} = & \, r_1 \times \left(X_{best} - \lambda_i \times r_2 \times \left(X_{best} - X^i_{old_{SF}}\right)\right) \\
& + (1 - r_1) \times \left(X^i_{elite_{SF}} - \lambda \times r_3 \times \left(\frac{X^i_{elite_{SF}} + X^i_{injured_S}}{2}\right) - X^i_{old_{SF}}\right)
\end{aligned} \tag{1}
$$

where X_{best} is the historical optimal value of the individual of this sailfish node, $X^i_{elite_{SF}}$ is the location of the optimal individual of the sailfish node in the i-th iteration, $X^i_{injured_S}$ is the location of the optimal individual of the sardine in the i-th iteration, $X^i_{old_{SF}}$ is the location of the current primary UAV node in the i-th iteration, r_1, r_2 and r_3 are uniformly distributed random numbers ranging from 0 to 1, and λ_i is the coefficients generated in the i-th iteration, as shown in Eq. (2):

$$\lambda_i = 2 \times rand(0,1) \times PD - PD \tag{2}$$

where PD is the sardine distribution density, which represents the proportion of sardine population in the whole population at each iteration. The number of sardine nodes decreases as the sardines run out of energy and are captured by sailfish nodes, which is calculated as shown in (3):

$$PD = 1 - \left(\frac{N_{SF}}{N_{SF} + N_S} \right) \tag{3}$$

where N_{SF} and N_S represent the number of sailfish population and the number of sardine population at each iteration, respectively. Since the sardine population will be removed from the sardine population and replaced by the elite sailfish when a better node than the elite sailfish appears in the sardine population, so the number of sardine population should be more than the sailfish population as much as possible. So the number of sardine population should be as much as possible than the sailfish population to ensure the iterative efficiency of sardine diversity. As shown in Eq. (1), this paper uses a random number to adjust the proportion of the influence of the traditional sailfish algorithm and the PSO algorithm in the iteration of sailfish nodes, which can prevent the algorithm from falling into the local optimum to some extent. This value can also be replaced by a fixed value to find the most suitable coefficient for the current problem.

3 Enhancing Population Exploration with Hybrid WOA-SFO Algorithm

In the sailfish algorithm, the sardine population is iterated by exploring the local optimum of the neighborhood of its non-sardine nodes by fleeing in the opposite direction according to the position of the sailfish selected when this iteration is made. In this paper, we propose to combine the exploration method of node spiral movement of WOA into the single vector movement of sardines, which makes the iteration results of sardines more diversified and thus improves the exploration efficiency of each sardine node.

In the sardine iterative algorithm, when AP $\geq$ 0.5 the sardine node randomly selects any sailfish node to escape so that the algorithm explores as much as possible the unknown domain in the early stage, while when AP $<$ 0.5 the sardine turns to select the position of the elite sailfish iteratively to assist the algorithm to break through the local optimum as the premise of exploration. In this paper, we propose to add a random factor p to let the sardine node choose the modified spiral iteration formula of WOA to iterate in the direction of the tangent of the spiral in each round of iteration, which uses the global optimum of the current iteration (the elite sailfish node) as a parameter as shown in Eq. (4):

$$X^i_{new_S} = \begin{cases} \begin{cases} \vec{A} \times \vec{D} + AP & \text{if } AP \geq 0.5 \\ \vec{A} \times \vec{D'} + AP & \text{if } AP < 0.5 \end{cases} & \text{if } p < 0.5 \\ \vec{D''} \times \ell^{bl} \times cos(2\pi l) + AP & \text{if } p \geq 0.5 \end{cases} \tag{4}$$

$$\vec{D} = \left| \vec{C} * X^i_{rand_{SF}} - X^i_{old_S} \right| \tag{5}$$

$$\vec{D'} = \left| \vec{C} * X^i_{elite_{SF}} - X^i_{old_S} \right| \tag{6}$$

$$\vec{D''} = \left| X^i_{elite_{SF}} - X^i_{old_S} \right| \tag{7}$$

where i represents the current iteration number,$\vec{D}$, $\vec{D'}$ and $\vec{D''}$ represent the relative distance between the current sardine node and the selected individual sailfish node in the ith iteration. b is a constant defining the shape of the helix, l is a random number in the range $[-1, 1]$. AP is the value representing the average energy consumption of the sardine population when this iteration, as shown in Eq. (8):

$$AP = a \times (1 - (2 \times Itr \times \varepsilon)) \tag{8}$$

a controls the range of AP, ε is the reciprocal of the two times maximum iteration number, when AP decreases from a to 0, the population iteration ends. By the parameter AP control sardine nodes can have the ability to effectively break through the local optimum in both the early and late stages of the algorithm iteration, to ensure that the algorithm can stably find the near-optimal solution.

$\vec{A}$ and $\vec{C}$ are coefficient vectors, as shown in Eq. (9) and (10):

$$\vec{A} = 2 \times \vec{\alpha} \times \vec{r} - \vec{\alpha} \tag{9}$$

$$\vec{C} = 2 \times \vec{r} \tag{10}$$

where $\vec{\alpha}$ is used to denote the hunting density of the sailfish nodes whose value decreases linearly from 2 to 0 with iterations, $\vec{A}$ is a random value in the range $[-\alpha, \alpha]$, and $\vec{r}$ is a uniformly distributed random vector from 0 to 1.

The main purpose of the sardine population is to obtain as many better candidate solutions from the current path solution space as possible in the limited iteration period, and deliver them to the sailfish nodes for iteration. Unlike the iterative approach of the sailfish population, this paper argues that the iterative approach of the sardine focuses more on operative stochastic exploration rather than structural learning with other locally optimal nodes as the algorithm in Sect. 2, which also makes the algorithm more efficient.

4 Application and Analysis of the Dual-Population Mechanism

The interaction between the sailfish and sardine populations occurs when a candidate solution in the sardine population outperforms the elite sailfish. At this point, the elite sailfish removes the sardine and replaces its own candidate solution. Throughout the algorithm's iterations, the sailfish continuously extracts new local optima from the sardine population, repeatedly breaking through the current iteration's local optima. During this process, the number of sardines gradually decreases as the sailfish captures them. Since the sardines' iteration process is independent of the sardine population as a whole and only depends on the sailfish nodes, the number of sardine nodes is linearly related to the diversity of randomly generated candidate solutions in the current iteration. Therefore, the sardine capture mechanism during the interaction between the sailfish and sardine populations influences the diversity of the entire algorithm.

After a certain number of iterations, the entire algorithm tends to stabilize at an approximate optimal value. This is because it becomes challenging for the sailfish iteration algorithm to develop solutions better than the current ones. On the other hand, the sardine population's iteration process, which essentially involves random exploration radiating from a starting point towards sailfish nodes, can easily lead to incorrect exploration directions. Thus, when a sardine is captured by a sailfish, the behavior of reiterating with a new random starting point is a key mechanism for the sailfish optimization algorithm to achieve local optima.

Through experimental analysis, this paper concludes that generating a new random sardine node immediately after capturing one can maintain the sardine population's diversity exploration efficiency. Conversely, retaining captured sardine nodes and generating a batch of new sardine nodes after reaching a certain quantity can make the algorithm's iterations more breakthrough-oriented. However, since the sardine node capture behavior is not frequent, the number of sardines generated at once significantly affects the sardine population's iteration progress. It may result in only a few replenishments occurring even after 500 iterations. Therefore, the value should be adjusted according to the actual situation during the formal algorithm iterations.

5 Experiments

5.1 Experimental Environment

In this paper, the specified global terrain model is obtained from the GEBCO_2022 dataset released by the Global Bathymetric Chart of the Oceans (GEBCO): the coordinates are LNG:40.0500, LAT:154.0000; the size of the sampling matrix is $20\,\text{km} * 20\,\text{km} * 20\,\text{km}$; the size of the 3D grid model is $200 * 200 * 200$; the hardware environment of the experiment is M2 chip, 8G memory; the operating system is MAC, and the algorithm is realized by MATLABr2022b software.

In the HPSO-WOA-SFO algorithm, 200 nodes are used for iteration, in which the number of sailfish nodes is 100 nodes and the number of sardine nodes is 100 nodes, the upper bound of each dimension value of the node is 1, and the lower bound is 0, with a total of 3,395 dimensions, and the maximum number of iterations is 400 iterations, and the initial value of the node is randomly generated by the initialization of the algorithm, and the ε is 0.00125 and the AP behavior threshold is 0.5.

5.2 Results and Analysis

Comparison Experiment. In the unified experimental environment, HPSO-WOA-SFO with PSO, SFO, GWO, and WOA performs path planning optimization experiments in the path planning solution space, and the simulation results are shown in Figs. 1 and 2:

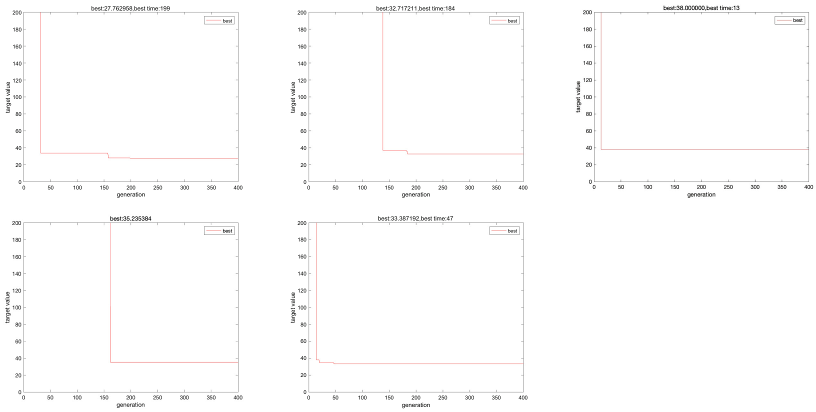

Fig. 1. (a) Iterative plot of optimal values of HPSO-WOA-SFO (b) Iterative map of optimal values of SFO (c) Iterative map of GWO optima (d) Iterative map of WOA optima (e) Iterative map of PSO optima.

Figure 1 shows the convergence optimal solution of each iteration of each path planning algorithm in the path planning solution space, and Fig. 2 shows the visualization path of the optimal solution at the end of the iteration in the 3D map of seabed topography and geomorphology. As shown in the bold part of Table 1, the experiments show that HPSO-WOA-SFO has obvious advantages in both convergence speed and convergence optimal value, as shown in Fig. 1(a), a better local optimal solution is found at the 32nd iteration and 161st iteration, and the global optimal value of 27.76 is finally found after the 199th iteration. Comparison with the SFO shown in Fig. 1(b) shows that, when improving the SFO Comparing with the SFO shown in Fig. 1(b), it can be seen that after improving the sailfish iteration process and the sardine iteration process in the

Table 1. Iterative Optimization Comparison Table of HPSO-WOA-SFO with Other Algorithms.

Algorithm	Global optimum/iter	First local optimum/iter
HPSO-WOA-SFO	**27.76**/199	**33.64**/32
SFO	32.71/186	37.08/140
GWO	38.00/13	38.00/13
WOA	35.23/162	35.23/162
PSO	33.38/47	38.05/16

SFO, the local optimal solution searching speed and the convergence accuracy of the local optimal solution of the HPSO-WOA-SFO algorithm are both improved. For the path planning problem with decentralized roads, each search for a better local optimal solution represents a breakthrough in the optimization result, as shown in Figs. 1(c) and (d), GWO and WOA are trapped in the local optimum in the solution space and cannot break through although they have found the local optimal solution. GWO, as shown in Fig. 1(c), finds the optimal value after the 13th iteration, and fails to find a more efficient path after the next 387 iterations, indicating that GWO still needs to be improved when exploring the data in the discrete solution space. PSO algorithm, as shown in Fig. 1(e), finds the local optimal solution and successfully improves the convergence accuracy at the 16th iteration, indicating that PSO algorithm is very efficient in this path The PSO algorithm is effective in this path planning problem, but it is unable to find other local optimal solutions, and the subsequent iterations also fall into the local optimum.

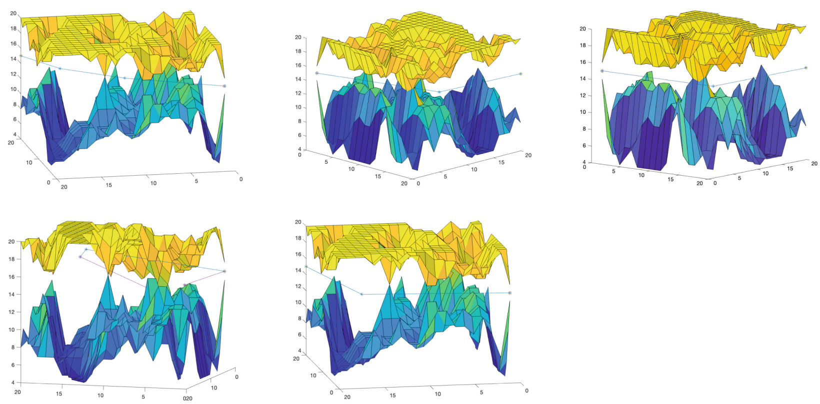

Fig. 2. (a) Iterative plot of optimal values of HPSO-WOA-SFO(b) Iterative map of optimal values of SFO (c) Iterative map of optima (d) Iterative map of WOA optima (e) Iterative map of PSO optima.

HPSO-WOA-SFO Ablation Experiment. The HPSO-WOA-SFO ablation experiment is divided into two groups of comparison experiments, HPSO-SFO and HWOA-SFO. The HPSO-SFO ablation experiment combines the iterative process of the sardine school in the original SFO with the HPSO-SFO; and the HWOA-SFO ablation experiment uses the iterative process of the sailfish school in the original SFO in combination with the HWOA-SFO and compares the two groups of ablation experimental subjects with the HPSO-WOA-SFO for a control.The HPSO-WOA-SFO ablation experiments of HPSO-SFO and HWOA-SFO were conducted to compare the experimental data as shown in Fig. 3:

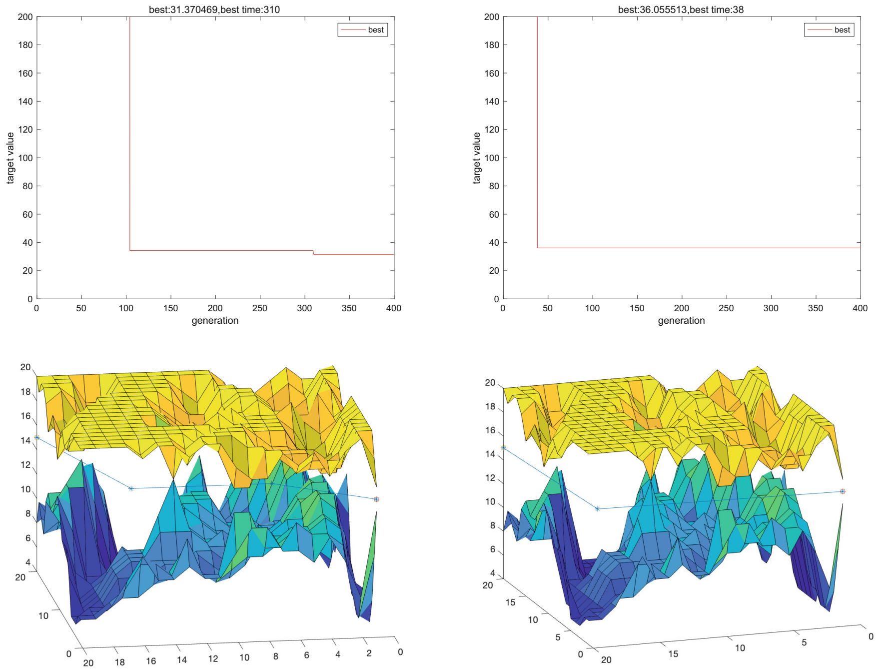

Fig. 3. (a) Iterative map of optimal values of HPSO-SFO (b) HWOA-SFO optimal value iterative map (c) Optimal roadmap for HPSO-SFO (d) HWOA-SFO optimal roadmap.

HPSO-SFO, as shown in Fig. 3(a), searches the local optimum value at the 104th iteration indicating its weak searching ability, but breaks through the optimum value to 34.27 at the 310th iteration. The fact that HPSO-SFO can break through the convergence accuracy of the difference of 2 in spite of its weak global optimization searching ability indicates the validity of its ability of convergence accuracy. IEFM, as shown in Fig. 3(b), searches the local optimum value to 36 at the 38th iteration, and has an advantage in the value of local optimum solution compared with the data in Table 1, showing the characteristics

Table 2. Comparison table of iterative optimization search for HPSO-WOA-SFO ablation experiments.

Algorithm	Global optimum/iter	First local optimum/iter
HPSO-WOA-SFO	**27.76**/199	**33.64**/32
HPSO-SFO	31.37/310	34.27/104
HWOA-SFO	36.00/38	36.00/38

of HWOA-SFO in terms of its optimization searching ability. HWOA-SFO, as shown in Fig. 3(b), searches for the local optimal solution value of 36 at the 38th iteration, which is advantageous in the value of the local optimal solution compared with the data in Table 1, showing the characteristics of HWOA-SFO in the optimization ability, and falls into the local optimum after that showing the limitation of the original SFO algorithm (Table 2).

6 Conclusion

This paper aims to address the issue of relatively strong discreteness in unmanned aerial vehicle path planning. It combines the scalability of the sailfish optimization algorithm with the improvement in exploitation and exploration capabilities through the integration of the Particle Swarm Optimization and Whale Optimization Algorithm. This enhancement allows the sailfish iterative algorithm to fully absorb the path structures of other excellent nodes in the population, thereby developing better solutions. Simultaneously, the sardine iterative algorithm efficiently explores the solution space, fully harnessing the advantages of the algorithm. Furthermore, this paper analyzes the strengths and weaknesses of the dual-population exchange mechanism implementation and demonstrates that within the sailfish optimization algorithm, the dual-population exchange mechanism plays a crucial role in determining the algorithm's diversity and optimization capability.

The experimental results show that HPSO-WOA-SFO can improve the global optimal convergence accuracy by 21.40% and the convergence speed by 10.71% compared with various path planning algorithms, and can quickly break through the local optimal when the algorithm is stuck in the local optimal, which makes the algorithm's global optimal value improve by 17.47%.

References

1. Hazim, S., et al.: Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges. IEEE Access **7**, 48572–48634 (2019)
2. Shubhani, A., Neeraj, K.: Path planning techniques for unmanned aerial vehicles: a review, solutions, and challenges. Comput. Commun. **149**, 270–299 (2020)
3. Faiza, G., et al.: Implementation of bio-inspired hybrid algorithm with mutation operator for robotic path planning. J. Parallel Distrib. Comput. **169**, 171–184 (2022)

4. John, H.H.: Genetic algorithms. Sci. Am. **267**(1), 66–73 (1992)
5. James, K., Russell, E.: Particle swarm optimization. In: Proceedings of ICNN 1995 - International Conference on Neural Networks, pp. 1942–1948. IEEE, Perth (2010)
6. Seyedali, M., Andrew, L.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
7. Seyedali, M., Lewus, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
8. Shadravan, S., et al.: The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Eng. Appl. Artif. Intell. **80**, 20–34 (2019)
9. Xiong, X., Hu, X., Guo, H.: A hybrid optimized grey seasonal variation index model improved by whale optimization algorithm for forecasting the residential electricity consumption. Energy **234**, 121127 (2021)
10. Xiong, X., et al.: A novel Optimized initial condition and Seasonal division based grey Seasonal Variation Index model for hydropower generation. Energy **328**, 120180 (2022)
11. Li, Z.K., et al.: A novel weighted average weakening buffer operator based fractional order accumulation seasonal grouping grey model for predicting the hydropower generation. Energy **277**, 127568 (2023)
12. Hu, X., et al.: A hybrid clustered SFLA-PSO algorithm for optimizing the timely and real-time rumor refutations in online social networks. Expert Syst. Appl. **212**, 118638 (2023)
13. Issam, E.H., et al.: Comparison of planning models for dynamic berth allocation problem using a sailfish-based algorithm. In: Proceedings of the 24th International Conference KES2020, pp. 3112–3120. Elsevier Procedia, Online (2020)
14. Rajasekaran, R., et al.: A novel intelligent transport system charging scheduling for electric vehicles using grey wolf optimizer and sailfish optimization algorithms. Part A: Recov. Utiliz. Environ. Effects **44**(2), 3555–3575 (2022)
15. Ghosh, K.K., et al.: Improved binary sailfish optimizer based on adaptive-hill climbing for feature selection. IEEE Access **8**, 83548–83560 (2020)