Computers, Materials & Continua CMC, vol.65, no.3, pp.2295-2308, 2020

3D Trajectory Planning of Positioning Error Correction Based on PSO-A* Algorithm

Huaixi Xing[1], Yu Zhao[1], Yuhui Zhang[1] and You Chen[1, *]

Abstract: Aiming at the yaw problem caused by inertial navigation system errors accumulation during the navigation of an intelligent aircraft, a three-dimensional trajectory planning method based on the particle swarm optimization-A star (PSO-A*) algorithm is designed. Firstly, an environment model for aircraft error correction is established, and the trajectory is discretized to calculate the positioning error. Next, the positioning error is corrected at many preset trajectory points. The shortest trajectory and the fewest correction times are regarded as optimization goals to improve the heuristic function of A star (A*) algorithm. Finally, the index weights are continuously optimized by the particle swarm optimization algorithm. The optimal trajectory is found by the A* algorithm under the current evaluation index, so the ideal trajectory is planned. The experimental results show that the PSO-A* algorithm can quickly search for ideal trajectories in different environment models, indicating that the algorithm has certain feasibility and adaptability, and verifies the rationality of the proposed trajectory planning model. The PSO-A* algorithm has better convergence accuracy than the A* algorithm, and the search efficiency is significantly better than the grid search A star (GS-A*) algorithm. The PSO-A* algorithm proposed in this paper has certain engineering application value. The researchers will study the real-time and systematic nature of the algorithm.

Keywords: Trajectory planning, PSO-A*, error correction, intelligent aircraft.

1 Introduction

Rapid trajectory planning in complex battlefield environment is an important problem that need to be solved urgently in the field of intelligent aircraft control [Zheng (2019); Pan (2012); Pan (2018); Yue and Xia (2019)]. Not only should the commander consider the variability of the air environment and the time-varying dynamic threats, but also meet the performance constraints of the UAV itself [Zhang, Zhang, Wang et al. (2018)]. As such, trajectory planning is a multi-constrained optimization problem. Moreover, the correlation between different constraints leads to a large amount of data in the search space, which makes it difficult to implement trajectory planning [Chen, Wang, Liu et al. (2019); Chen, Wang, Xia et al. (2019); Zhang, Jin, Sun et al. (2018)]. In addition, the fuel

[1] Air Force Engineering University, Xi'an, 710038, China.

[*] Corresponding Author: You Chen. Email: 17865160875@163.com.

Received: 02 June 2020; Accepted: 11 July 2020.

or power that the aircraft can carry when completing a flight mission is limited, so the entire flight must be limited by certain flight restrictions. The selection of trajectory must satisfy the premise of aircraft performance constraints, so that the aircraft can quickly and successfully reach the target position, ensuring the successful completion of the mission. The inertial navigation system obtains the current position information through the previously calculated position information and the measured acceleration and angular velocity. The accumulated positioning error is roughly proportional to the duration of the flight. Therefore, if the positioning error is too large during navigation, the aircraft will not be able to autonomously navigate along the preset trajectory route. In order to ensure the positioning accuracy of the inertial navigation system that works for a long time, it is necessary to correct the positioning error during the flight to ensure the intelligent aircraft to fly along a predetermined trajectory. Therefore, the authors establish an environment model in which the positioning error is corrected at a specific trajectory point and the trajectory planning under the constraint of navigation error can be achieved.

Because trajectory planning is an NP-hard problem [Wei, Sun and Lv (2018); Zhang and Xue (2017); Xia, Hu and Luo (2017)], the performance of an applied algorithm directly affects the quality of the planned trajectory. Intelligent optimization algorithms such as genetic algorithm (GA), particle swarm optimization (PSO), and ant colony algorithm (ACO) have been widely applied in this field [Huang and Zhao (2018a); Fang and Xu (2017)]. Liu et al. [Liu, Wang, Liu et al. (2017)] added greed and mutation strategies to improve the gray wolf algorithm in order to overcome the shortcomings of development capabilities. Zhang et al. [Zhang, Quan, Wen et al. (2020)] mixed two or more optimization algorithms to solve the trajectory planning problem. The search mechanism of a single algorithm changed, and the convergence accuracy and path finding ability were effectively improved [Kamboj (2016)]. The above algorithms have certain advantages in realizing low-dimensional planning of trajectories on the premise of determining the number of trajectory points, but they are not suitable for specific scenarios where the number of trajectory points is uncertain. The more trajectory points, the greater the dimension. The calculation of the algorithm is extremely complicated, which affects its search performance, resulting in slow convergence speed, poor optimization ability, and bad quality of the planned trajectory. Therefore, the scope of engineering application is relatively limited.

The A* algorithm Zhan et al. [Zhan, Wang, Chen et al. (2015); Zhang, Li, Zhang et al. (2016)], as a classic heuristic search algorithm [Huang, Zhao and Han (2018b)], can efficiently search for the optimal trajectory with the advantage of simplified calculation. According to the notion of a step-by-step search in the state space, each node is evaluated by the heuristic function [Wang, Ma and Xie (2010)]. The best node with a certain directivity is selected for expansion until the aircraft reaches the predetermined position. The A* algorithm has strong applicability without limitation of the number of trajectory points. Moreover, the existence of the heuristic function greatly reduces the search space and improves the search efficiency, although the performance of the A* algorithm greatly depends on the choice of the heuristic function. Whether the heuristic function is reasonable, directly affects its convergence speed and accuracy.

In order to implement the three-dimensional trajectory planning of intelligent aircraft

under multiple constraints and uncertain conditions, the authors proposed a particle swarm optimization-A* (PSO-A*) algorithm. First of all, a reasonable cost function consisting of weighted evaluation indicators was designed to determine the feasibility of the trajectory point to be expanded. Next, the initial trajectory under different weights was planned by the A* algorithm. The trajectory cost, called the fitness value, was regarded as the basis for updating the position of the particle swarm. Finally，weight assignment was continuously updated by the PSO algorithm to search for cost functions suitable for complex environmental constraints and aircraft performance requirements. The trajectory corresponding to the optimized cost function was also continuously updated. In the later stage of the iteration, the ideal trajectory route was planned.

The rest of the paper is organized as follows：

In the second chapter, a model for correcting positioning errors at trajectory points is established. In the third chapter, trajectory point selection based on particle swarm optimization algorithm is described in detail. In the fourth chapter, the trajectory planning of two sets of trajectory point information is simulated. Finally, conclusions are drawn in chapter five.

2 Trajectory planning model based on node error correction

2.1 Environmental model and constraints

We assume that the aircraft is flying in the airspace of $100km \times 100km \times 10km$ as shown in Fig. 1. The starting position is point A and the end position is point B.

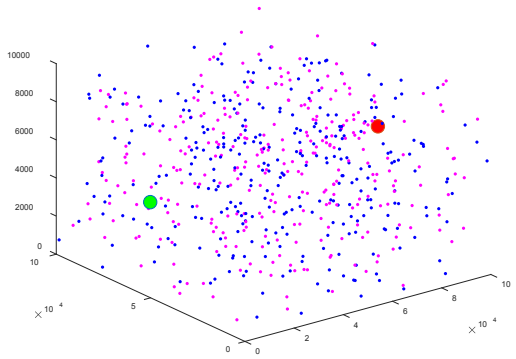


Figure 1: Airspace environment diagram

2.1.1 Navigation error setting

Considering that the flight trajectory of the aircraft is affected by the positioning and navigation accuracy, it is defined that the positioning error of the aircraft in three-dimensional space is represented by a horizontal error and a vertical error [De Filippis, Guglieri and Quagliotti (2012)]. Both are proportional to the flight distance and are defined as in Eq. (1).

$$\begin{cases}\Delta L = \delta \cdot r \\ \Delta H = \delta \cdot r\end{cases} \tag{1}$$

In Eq. (1), ΔL and ΔH are the accumulated errors in the horizontal and vertical directions. δ is the unit error, and r is the flight distance. Assuming that the error of the aircraft's starting point is 0, the author stipulates that the vertical error and the horizontal error should be less than θ units when reaching the end point. In order to simplify the problem, the author assumed that when the vertical error and the horizontal error are both less than θ units, the aircraft can still follow the planned trajectory.

2.1.2 Error correction point setting

In order to reach the end position successfully, the aircraft needs to constantly correct the positioning errors during the flight. There are certain fixed positions (called correction points) where the positioning errors can be corrected in the airspace. When the aircraft passes through a correction point, the corresponding errors can be eliminated according to this correction point type. As shown in Fig. 1, the red points are the horizontal correction points, represented by l_k ; the blue points are the vertical correction point, represented by h_k , and k represents the correction point number. For example, when the aircraft passes through a horizontal correction point, the horizontal error is eliminated, while vertical error cannot be corrected. If the vertical and horizontal errors can be corrected in time, the aircraft will not deviate from the flight trajectory and can finally reach the end point after passing through several correction points.

2.1.3 Constraints to avoid deviation from trajectory

When the accumulated positioning errors exceed the limit, the aircraft will deviate greatly from the predetermined trajectory. Even if the correction point is passed through, the aircraft cannot be successfully guided back to the predetermined trajectory. As a result, the aircraft cannot reach the end point successfully. In this paper, it is stipulated that k_i represents the number of a vertical correction point, and $i = 1,2,\cdots,m$ is the order of performing vertical correction; k_j represents the number of a horizontal correction point, and $j = 1,2,\cdots,n$ is the order of performing horizontal correction. $\Delta H(\bullet,\bullet)$, $\Delta L(\bullet,\bullet)$ indicate the vertical and horizontal errors accumulated after passing through adjacent correction points. The specific constraints are as follows:

Conditions for successful correction of vertical positioning error:

$$\Delta H\left(h_{k_i}, h_{k_{i-1}}\right) \le \alpha_1 \quad , \quad \Delta L\left(h_{k_i}, l_{k_j}\right) \le \alpha_2 \tag{2}$$

In Eq. (2), l_{k_j} represents the nearest horizontal correction point before correction point h_{k_i} .

Conditions for successful correction of horizontal positioning error:

$$\Delta H\left(l_{k_j}, h_{k_i}\right) \le \beta_1 \ , \ \Delta L\left(l_{k_j}, l_{k_{j-1}}\right) \le \beta_2 \tag{3}$$

In Eq. (3), h_{k_i} represents the nearest vertical correction point before correction point l_{k_j} .

2.1.4 Other constraints

To simplify the planning model, it is assumed that the aircraft speed is always uniform. Moreover, when the aircraft makes a maneuvering turn, it is assumed that only the turning time is recorded without any accumulated positioning error.

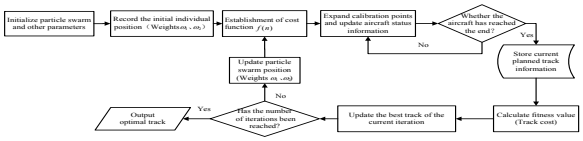
2.2 Objective function planning

The flight range is limited by energy consumption and flight time allocation. In this paper, the shortest trajectory and the least number of correction points are regarded as the optimization principles. Finding an optimal trajectory enables the aircraft to quickly reach a designated location to perform combat missions.

$Q(k_i)$ represents the coordinate vector of the correction point k_i which the aircraft passes through, k_i represents the number of the correction point to perform the correction, and i represents the sequence of performing the correction, $i = 1, 2, ..., r$. The coordinates of the starting point A and the end point B are indicated by $Q(A)$ and $Q(B)$, respectively. A counting function $n(k)$ is defined as Eq. (4).

$$n(k) = \begin{cases} 1, & \textit{Correction point k passed by the aircraft} \\ 0, & \textit{Correction point k not passed by the aircraft} \end{cases} \tag{4}$$

In Eq. (5), the objective function is established with the shortest trajectory and the least correction points as trajectory planning indicators.



(5)

3 Trajectory planning algorithm

3.1 A algorithm*

The A* algorithm is a heuristic algorithm, which is widely used in the field of trajectory planning. The cost of a correction point in the trajectory is estimated by the heuristic function, so that the flight direction intelligently tends to the specified position, which is the core of the A* algorithm. The A* algorithm takes up less storage space with low complexity and less calculation. The heuristic function is defined as Eq. (6).

$$f(n) = g(n) + h(n) \tag{6}$$

In Eq. (6), n is the expanded node; $g(n)$ is the actual cost from the initial node to the node n in the state space; $h(n)$ is the estimated cost from node n to the end node. The specific search rules are as follows: The trajectory gradually extends outward from the initial node. Before each extension, the set of nodes to be extended consists of all nodes that meet the constraints. Whether the current node is feasible can be evaluated by comparing the cost of each node to be extended. The node with the smallest cost function value $f(n)$ is regarded as the next new node, and the set of nodes to be extended in the next round is updated until the target node meets the constraints. The following is the

algorithm flow [Huang, Fei, Liu et al. (2017)]:

Step 1 Initialize. Create an Open list and a Close list. The starting point information is placed in the Open list, and the Close list is cleared.

Step 2 Extension of the trajectory. Select the node with the smallest $f(n)$ among all the nodes that meet the constraints as the extension node,and delete it from the Open list and load it into the Close list.

Step 3 Constraint judgment. If the current expansion node does not meet the termination condition, repeat Step 2; otherwise, perform Step 4.

Step4 Trajectory backtracking. Traverse the parent node in order from the last node, and store all the nodes in the result array, which is the final trajectory information of the result.

*3.2 3D trajectory planning algorithm based on PSO-A **

Essentially, the choice of cost function determines the performance of the algorithm, so the importance of the A* algorithm lies in designing a reasonable heuristic function. Since the trajectory planning in this paper is based on the shortest trajectory and the least correction points, the cost function is defined as follows:

$$f(n) = \omega_1 \cdot \sum_{i=1}^{n} \Delta P(i) + \omega_2 \cdot R(n) \tag{7}$$

In Eq. (7), $R(n)$ represents the Euclidean distance from the current node to the end node, the penalty function $\Delta P(i)$ represents the vertical offset of the i-th node in the direction of the $i-1$-th node to the end node, meaning the degree of deviation from the specified trajectory.. The penalty function can speed up the search for the best correction point and avoid falling into partial optimal and deadlock. ω_1 and ω_2 are weight coefficients, and $\omega_1 + \omega_2 = 1$. The two indicators of the cost function have the consistent effect, one of which makes the aircraft try to avoid invalid flight and reduce the route, the other of which makes the aircraft quickly approach the end point and pass through fewer correction points. Because ω_1 and ω_2 have different contributions to the search efficiency of the algorithm, and may not be on the same order of magnitude, the setting of the weight coefficient is necessary, which determines the merits of the cost function and affects the search ability of the algorithm.

As a classic swarm intelligence algorithm, PSO has unique advantages in parameter optimization. In order to obtain a better overall trajectory, it is necessary to optimize the weight coefficients to enhance the rationality of the cost function. In this paper, a three-dimensional trajectory planning method combining A * algorithm and particle swarm optimization algorithm (PSO-A *) is proposed. In the PSO algorithm, the weight coefficient update depends on the level of fitness value. The updated weight coefficients and cost function form a one-to-one mapping. The corresponding current optimal trajectory is called a non-inferior solution. When the search is completed, the global optimal solution containing the information of the optimal trajectory is determined. The specific algorithm flow is shown in Fig. 2.

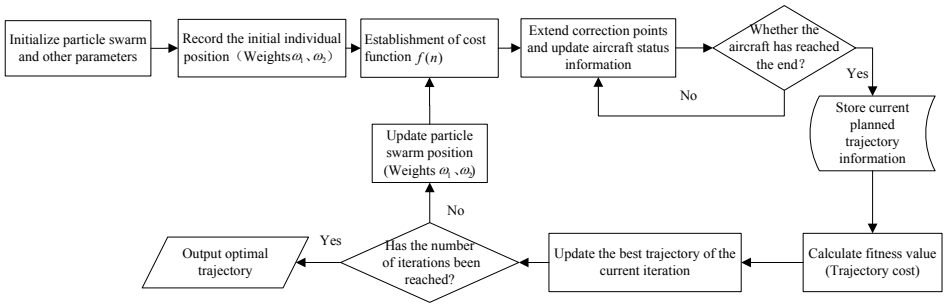


Figure 2: PSO-A* algorithm flow chart

4 Results

Environment parameter setting: The planning space is the area shown in Fig. 1.

Data set 1: A total of 612 correction points, including 305 vertical correction points and 306 horizontal correction points. The coordinate of the starting point A is (0,50,5) km, the coordinate of the ending point B is (100,59.65,5.02) km,

$\alpha_1 = 25$, $\alpha_2 = 15$, $\beta_1 = 20$, $\beta_2 = 25$, $\theta = 30$, $\delta = 0.001$.

Data set 2: A total of 325 correction points, including 158 vertical correction points and 167 horizontal correction points. The coordinate of starting point A is (0,50,5) km, and the coordinate of ending point B is (100,74.86,5.50) km,

$\alpha_1 = 20$, $\alpha_2 = 10$, $\beta_1 = 15$, $\beta_2 = 20$, $\theta = 20$, $\delta = 0.001$.

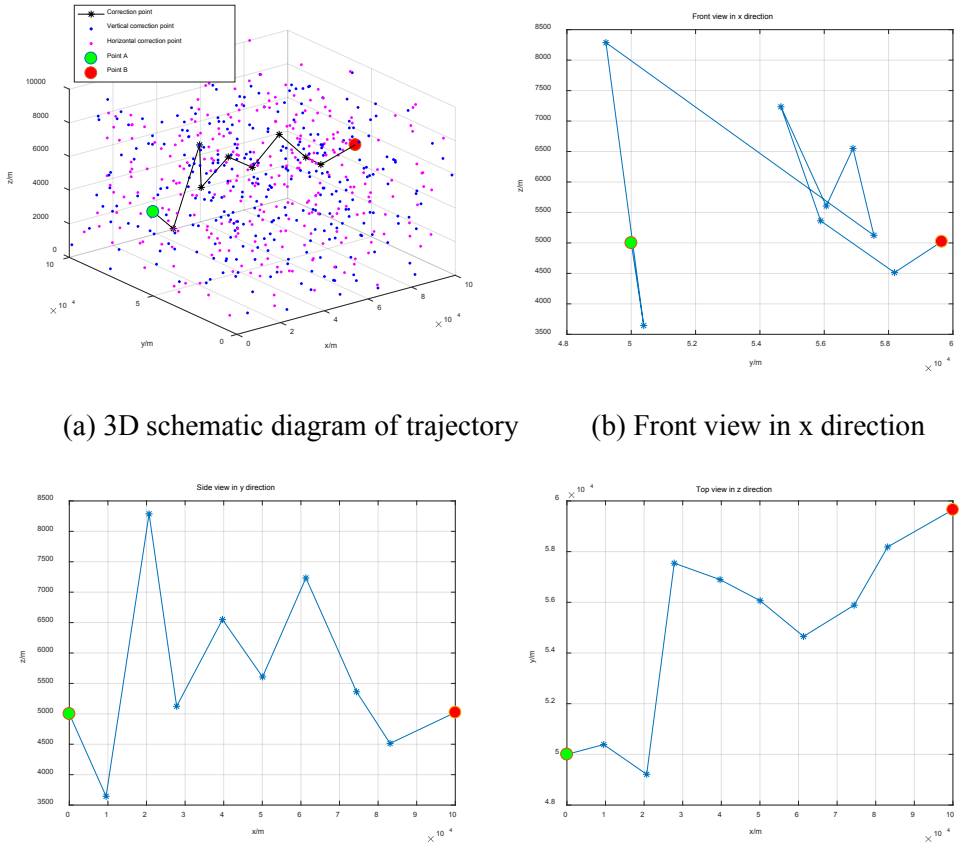
4.1 Obtaining the optimal trajectory

The simulation experiment was conducted on the MATLAB R2016b platform with Intel Core i5-4210H CPU@2.90 GHz, 4 GB memory, and 64-bit operating system. The PSO-A* algorithm is applied to the established model for trajectory planning. The number of particles 40, the learning factor is $c_1 = 2$, $c_2 = 2$, the inertia weight is $\omega = 0.8$, and update times are 100. The optimal trajectory of dataset 1 is obtained when the weight distribution are $\omega_1 = 0.43$, $\omega_2 = 0.57$. A total of 8 correction points are passed through, including 4 vertical correction points and 4 horizontal correction points. The flight range is 105160.54 m, as shown in Fig. 3. The optimal trajectory of dataset 2 is obtained when the weight distribution are $\omega_1 = 0.31$, $\omega_2 = 0.69$,. A total of 12 correction points are passed through, including 6 vertical correction points and 6 horizontal correction points. The flight range is 110004.89 m, as shown in Fig. 4.

Figs. 3(b), 3(c), 3(d), 4(b), 4(c) and 4(d) are three views of trajectory 1 and trajectory 2 in the X, Y, and Z directions, respectively, which intuitively show the offset of the intelligent aircraft during the flight. At the same time, Tabs. 1 and 2 record the correction point number and the positioning errors before correction in dataset 1 and dataset 2, respectively. In order to facilitate statistics, the horizontal correction point is indicated by 0, and the vertical correction point is indicated by 1. It can be seen from Figs. 3(b), 3(c), 3(d), 4(b), 4(c) and 4(d) that the horizontal offset is much larger than vertical offset, indicating that the algorithm searches for extended correction points with the horizontal offset as the main cost.

The flight range is closely related to the distribution of correction points. The positioning errors before correction in Tabs. 1 and 2 meet all constraints, further verifying the feasibility of all correction points and the effectiveness of the algorithm.

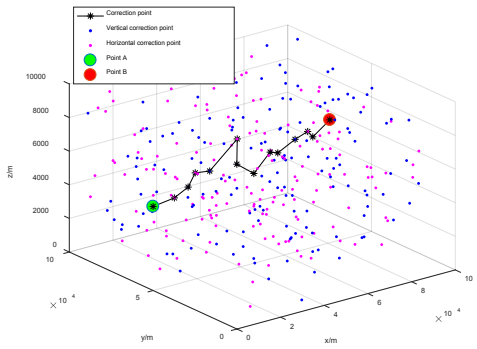
The adjacent extended points are different types of correction points, which means that the horizontal errors and vertical errors are alternately corrected during the flight. To minimize the cost of the trajectory, when approaching next correction point during flight, the positioning error before correction must not only satisfy the constraints, but also approach the upper limit of the allowable positioning error as much as possible under the guidance of the algorithm heuristic function.



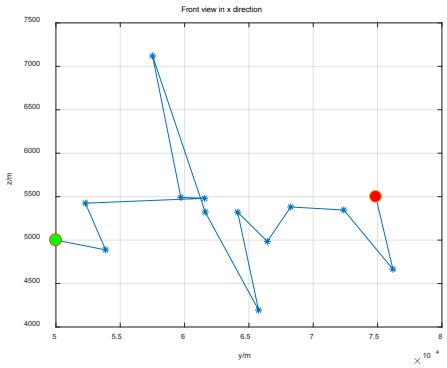
(a) 3D schematic diagram of trajectory (b) Front view in x direction

(c) Side view in y direction (d) Top view in z direction

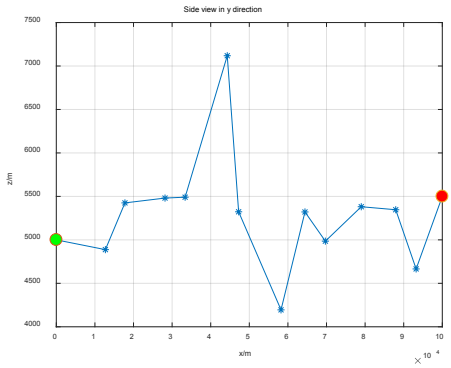
Figure 3: Results for dataset 1



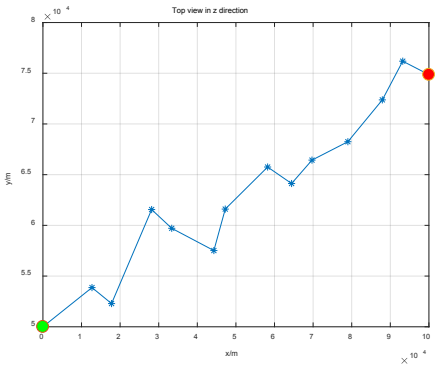
(a) 3D schematic diagram of trajectory



(b) Front view in x direction



(c) Side view in y direction



(d) Top view in z direction

Figure 4: Results for dataset 2

Table 1: Correction points information in dataset 1

Correction point number	Vertical positioning error before correction	Horizontal positioning error before correction	Correction point type
0	0	0	A
503	13.38792	13.38792	1
69	8.8073432	22.1952621	0
237	21.306825	12.4994831	1
155	11.2008147	23.7003005	0
540	17.9140674	6.7132501	1
250	11.4293251	18.1425752	0
340	24.1984492	12.7691241	1
277	12.002376	24.7715001	0
612	28.3532805	16.3509045	B

Table 2: Correction points information in dataset 2

Correction point number	Vertical positioning error before correction	Horizontal positioning error before correction	Correction point type
0	0	0	A
163	13.2879	13.2879	0
114	18.6221	5.3342	1
8	13.922	19.2561	0
309	19.4463	5.5243	1
305	5.9687	11.493	0
123	15.1731	9.2044	1
231	9.4367	18.6411	0
160	18.1539	8.7172	1
92	5.7762	14.4933	0
93	15.2609	9.4847	1
61	9.8342	19.3189	0
292	16.3881	6.5539	1
326	6.9605	13.5144	B

4.2 Performance comparison of different algorithms

4.2.1 Standard A algorithm*

The parameter setting is consistent with Section 4.1. The author uses the standard A* algorithm for trajectory planning. $\omega_1 = 1$, $\omega_2 = 1$, and the optimal flight range of dataset 1 is $1.0635 \times 10^5 m$ with 8 correction points, including 4 horizontal and 4 vertical correction points. The optimal flight range of dataset 2 is $1.1139 \times 10^5 m$ with 12 correction points, including 6 horizontal and 6 vertical correction points. Tab. 3 and Tab. 4 list all the information of the correction points. Compared with the results in Section 4.1, Part of the correction points obtained by the PSO-A* algorithm is replaced. Although the number of correction points is equal to the former, the cost of the trajectory is relatively large, which indicates that the A* algorithm lacks partial search capability and has obvious randomness in extension of correction points.

Table 3: Correction points information in dataset 1 solved by the A* algorithm

Correction point number	Vertical positioning error before correction	Horizontal positioning error before correction	Correction point type
0	0	0	A
521	9.62651	9.62651	0
64	21.75541	12.1289	1
80	11.4211	23.55	0
170	23.3982	11.9771	1
278	10.457	22.4341	0
369	21.8931	11.4361	1
214	13.3136	24.7497	0
397	22.33068	9.01708	1
612	16.9727	25.9898	B

Table 4: Correction points information in dataset 2 solved by the A* algorithm

Correction point number	Vertical positioning error before correction	Horizontal positioning error before correction	Correction point type
0	0	0	A
163	13.2879	13.2879	0
114	18.6221	5.3342	1
8	13.922	19.2562	0
309	19.4463	5.5243	1
121	11.252	16.7763	0
123	16.6036	5.3516	1
49	11.7902	17.1418	0
160	18.304	6.5138	1
92	5.7762	12.29	0
93	15.2609	9.4847	1
61	9.8342	19.3189	0
292	16.3881	6.5539	1
326	6.9605	13.5144	B

4.2.2 GS-A algorithm*

Grid search (GS) algorithm is a search algorithm similar to traversal and has strong stability. However, the amount of calculation of the GS algorithm is proportional to the square of the grid resolution. The higher the resolution, the more complicated the calculation and the longer the convergence time. In the simulation, it is assumed that the grid spacing is 1 and the grid size is 100×100. The authors performed trajectory planning on dataset 1 by grid search and A*(GS-A*) algorithm. Fig. 5 shows the optimal flight range under different weight distribution. The optimal trajectory is consistent with the result obtained by the PSO-A* algorithm, but the corresponding optimal ratio of weight coefficient is approximately 1. Obviously, the mapping from the cost function to the optimal trajectory is many-to-one, which indicates that the PSO-A* algorithm has certain fault tolerance when finding weight combinations, and the optimal weight distribution is not unique.

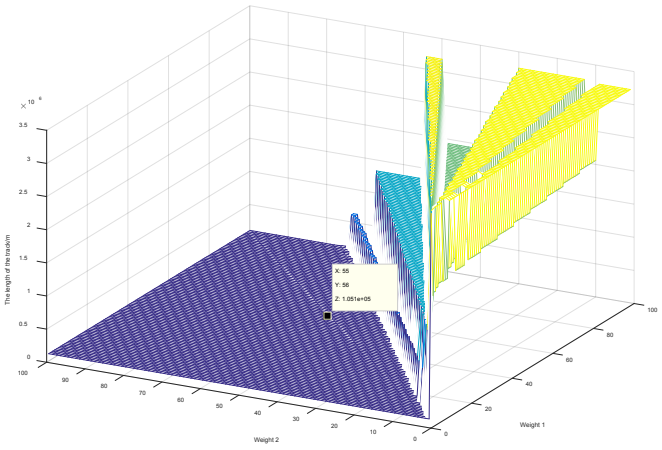


Figure 5: Weight distribution

4.2.3 Comparison of results obtained by several algorithms

The environmental parameters are changed, such as the number and the layout of correction points. In order to compare the optimization capabilities of different algorithms, trajectory planning is performed separately by A* algorithm, GS-A* algorithm and PSO-A* algorithm. The average values after 10 tests under each environmental condition are counted. The results are shown in Tab. 5.

Table 5: Trajectory evaluation index of different algorithms

Region	Algorithms	Flight range /km	Number of correction points	Time consumption /s
Dataset 1	A*	108.33	9.63	2.25
	GS-A*	105.16	8	40.81
	PSO-A*	105.41	8.14	11.33
Dataset 2	A*	113.75	12.91	3.17
	GS-A*	110.00	12	58.28
	PSO-A*	110.67	12.21	14.19

Comparing the several indicators in Tab. 5, the A* algorithm has the highest search efficiency and the shortest search time. Because once the heuristic function is determined, only a single search is needed to obtain a trajectory. However, the results have a certain randomness. The quality of the solution may be poor, resulting in a larger cost of trajectory.Because the GS-A* algorithm has the advantages of traversal search and strong overall search capability. The consistency of the optimal trajectory obtained through multiple experiments indicates that the GS-A* algorithm has good stability, but it will bring the disadvantages of more calculations and longer search time. PSO-A* algorithm can plan better trajectory results with fewer iterations, greatly reducing the trajectory search time and overcoming the defect that the convergence accuracy of the A* algorithm is limited by the heuristic function. The conclusions obtained above indicate that the PSO-A* algorithm can be well applied to the model in this paper and effectively realize 3D trajectory planning for intelligent aircraft. On the premise of ensuring that the aircraft successfully approaches the end point to complete the specified mission, the author's method can quickly search for a more ideal trajectory, and the performance is also greatly improved compared with the original algorithm.

5 Conclusions

By evaluating the optimal weight coefficient distribution of different indicators and constructing the most reasonable cost function, the optimal planning trajectory is obtained. The experimental results show that the PSO-A* algorithm can overcome the defects of the A* algorithm such as lack of optimization ability and low convergence accuracy under the same scenario. Although the convergence accuracy and stability of the PSO-A* algorithm are slightly inferior to the GS-A* algorithm, the calculation takes less time, and the convergence speed is significantly superior than the latter. In actual flight missions, it is completely acceptable to sacrifice part of the trajectory cost to shorten the search time.

Acknowledgement: The authors express sincere appreciation to the anonymous referees for their helpful comments to improve the quality of the paper.

Funding Statement: The authors received no specific funding for this study.

Conflicts of 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.

References

Chen, Y. T.; Wang, J.; Liu, S. J.; Chen, X.; Xiong, J. et al. (2019): Multiscale fast correlation filtering tracking algorithm based on a feature fusion model. *Concurrency and Computation: Practice and Experience*, e5533.

Chen, Y. T.; Wang, J.; Xia, R. L.; Zhang, Q.; Cao, Z. H. et al. (2019): The visual object tracking algorithm research based on adaptive combination kernel. *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4855-4867.

De Filippis, L.; Guglieri, G.; Quagliotti, F. (2012): Path planning strategies for UAVS in 3D Environments. *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 247-264.

Fang, Q.; Xu, Q. (2017): 3D route planning for UAV based on improved PSO algorithm. *Journal of Northwestern Polytechnical University*, vol. 35, no. 1, pp. 66-73.

Huang, C. Q.; Zhao, K. X. (2018a): Three dimensional path planning of UAV with improved ant lion optimizer. *Journal of Electronics & Information Technology*, vol. 40, no. 7, pp. 13-19.

Huang, C. Q.; Zhao, K. X.; Han, B. J. (2018b): Maneuvering decision-making method of UAV based on approximate dynamic programming. *Journal of Electronics & Information Technology*, vol. 40, no. 10, pp. 166 -171.

Huang, C.; Fei, J. Y.; Liu, Y.; Li, H.; Liu, X. D. (2017): Smooth path planning method based on dynamic feedback A* ant colony algorithm. *Transactions of the Chinese Society for Agricultural Machinery*, vol. 48, no. 4, pp. 34-40.

Kamboj, V. K. (2016): A novel hybrid PSO-GWO approach for unit commitment problem. *Neural Computing and Applications*, vol. 27, no. 6, pp. 1643-1655.

Liu, C. A.; Wang, X. P.; Liu, C. Y. (2017): Three-dimensional route planning for unmanned aerial vehicle based on improved grey wolf optimizer algorithm. *Journal of Huazhong University of Science and Technology (Nature Science Edition)*, vol. 45, no. 10, pp. 38-42.

Pan, H. Z. (2012): A daptive navigation control for quadrotor unmanned aerial vehicles. *Computer Simulation*, vol. 29, no. 5, pp. 106-110.

Pan, Y. F. (2018): *Autonomous Cooperative Control for Unmanned Aerial Vehicles (M.S. Thesis)*. Nanjing University of Aeronautics and Astronautics.

Wang, H. W.; Ma, Y.; Xie, Y. (2010): Mobile robot optimal path planning based on smoothing A* algorithm. *Journal of Tongji University (Nature Science)*, vol. 38, no. 11, pp. 1647-1650.

Wei, Z. C.; Sun, R. H.; Lv, Z. W. (2018): Path planning algorithm for WCE with joint energy replenishment and data collection based on multi-objective optimization. *Journal on Communications*, vol. 39, no. 10, pp. 26-37.

Xia, Z. Q.; Hu, Z. Z.; Luo, J. P. (2017): UPTP vehicle trajectory prediction based on user preference under complexity environment. *Wireless Personal Communications*, vol. 97, no. 3, pp. 4651-4665.

Yue, H. L.; Xia, P. Q. (2019): A wake bending unsteady dynamic inflow model of tiltrotor in conversion flight of tiltrotor aircraft. *Science China (Technological Sciences)*, vol. 52, no. 11, pp. 3188-3197.

Zhan, W. W.; Wang, W.; Chen, N. C.; Wang, C. (2015): Path planning strategies for UAV based on improved A* algorithm. *Geomatics and Information Science of Wuhan University*, vol. 40, no. 3, pp. 315-320.

Zhang, J. M.; Jin, X. K.; Sun, J.; Wang, J.; Sangaiah, A. K. (2018): Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-018-6562-8.

Zhang, Q.; Xue, S. (2017): An improved multi-objective particle swarm optimization algorithm. *Mathematical Problems in Engineering*, vol. 28, no. 7, pp. 482-490.

Zhang, S.; Li, X. R.; Zhang, P.; Li, B. (2016): UAV path planning based on improved A* algorithm. *Flight Dynamics*, vol. 34, no. 3, pp. 39-43.

Zhang, Y.; Quan, H.; Wen, J. H. (2020): Mobile robot path planning based on the wolf ant colony hybrid algorithm. *Journal of Huazhong University of Science and Technology (Nature Science Edition)*, vol. 48, no. 1, pp. 127-132.

Zhang, Z. Y.; Zhang, J.; Wang, P.; Chen, L. (2018): Research on operation of UAVs in non-isolated airspace. *Computers, Materials & Continua*, vol. 57, no. 1, pp. 151-166.

Zheng, J. (2019): *Research on Long-Term Integrated Navigation Method of UAV (M.S. Thesis)*. Harbin Institute of Technology.