Application of simulated annealing particle swarm optimization in complex three-dimensional path planning

Fang Wangsheng¹, Wang Chong¹* and Zhao Ruhua¹

¹ School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China

* Corresponding author’s e-mail: 6720190593@mail.jxust.edu.cn

Abstract. Particle Swarm Optimization (PSO) has achieved good results in UAV path planning, but there is still the phenomenon of abandoning the global optimal path and choosing the local optimal one. In order to improve the ability of particle swarm in path planning, a simulated annealing particle swarm algorithm is proposed. First, tent reverse learning is used to initialize the population so that the algorithm is evenly distributed in space. Then annealing operation is performed after iteration once, which has better local path judgment ability and avoids the phenomenon of local optimum to some extent, so as to find a more satisfactory path. Simulated annealing particle swarms can find a clear and satisfactory path with high stability through the complex three-dimensional path planning simulation of UAV.

1. Introduction

The path planning of an unmanned aerial vehicle (UAV) is to find an optimal path from the starting point to the ending point under the constraints of terrain, circumvention of local radar weapons, and flight conditions. In other words, the planned route meets its own security requirements and avoids the dangers around it. Track planning is a spatial search algorithm, usually divided into heuristic and evolutionary algorithms. Heuristic algorithm is a computational search using points in space. The process is complex and computationally intensive, which can easily lead to combined blasting and other problems. Evolutionary algorithms mostly use swarm intelligence algorithms, such as Ant Colony Optimization, Grey Wolf Optimizer and Particle Swarm Optimization.

In recent years, route planning based on swarm intelligence algorithm has developed rapidly, among which Particle swarm optimization (PSO) is more popular in route planning. PSO has attracted much attention because of its fast operation speed and easy implementation, but it still has the drawbacks of being trapped in local optimum and strong randomness. Chen Tianpei et al. [1] learned from the idea of Ant Colony Optimization pheromone, added pheromone for each particle, and introduced the empirical criteria of fuzzy system to avoid the disadvantage that traditional particle swarm algorithm is prone to local optimization. Finally, the effectiveness of the improved algorithm is shown by the three-dimensional path planning. Chen Shiming et al. [2] propose a hybrid mutation particle swarm optimization algorithm, which further divides the feasible domain and search boundary by splitting dimension reduction, reduces the search scope of the shortest path solution, and then shrinks the solution space by using directional and random mutation operations. Finally, the feasibility and superiority of the algorithm in three-dimensional path planning are verified by simulation experiments.

In summary, each scholar has made their own improvements by studying the principle of particle swarm, and the feasibility has been verified in experiments. On this basis, a simulated annealing
particle swarm optimization (SAPSO) is proposed, which uses tent mapping reverse learning to initialize the population, so that the individuals of the population are evenly distributed in space, and the search horizon of the algorithm is improved. Then, the simulated annealing algorithm is used to update the solution obtained each time, which has a better judgment ability and reduces the probability of the algorithm falling into the local solution. SAPSO has good convergence ability and can find a simple and satisfactory route through the three-dimensional path planning map of unmanned aerial vehicle complex terrain.

2. Simulated annealing particle swarm optimization

2.1. Particle swarm optimization

Particle swarm optimization (PSO) is an optimization algorithm extracted from a model of bird swarms searching for food. Each particle is the equivalent of a bird, with its own velocity and fitness value. Moreover, each particle iteratively updates its own location through individual extremum and global extremum. The steps to find the optimal solution are as follows:

1. Initialize the particle swarm;
2. All particles of the particle swarm are calculated circularly;
3. Calculate the particle fitness \( f_i \). Suppose that the best solution found by the particle itself is labeled \( p_{best} = f_i \). If \( f_i > p_{best} \), then \( p_{best} = f_i \);
4. Select the particle with the best fitness value of all particles as \( g_{best} \);
5. Calculate the particle velocity \( v_i \) according to the following formula:
   \[
   v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot [p_{best} - \text{pos}_i(t)] + c_2 \cdot r_2 \cdot [g_{best} - \text{pos}_i(t)]
   \]
   In the above formula, \( t \) represents the number of iterations, \( w \) is the inertial weight, \( c \) is the learning factor, \( r \) is a random number, and its value range is \([0,1] \);
6. Update the \( \text{pos}_i(t) \) of the particle position according to the following formula:
   \[
   \text{pos}_i(t) = \text{pos}_i(t) + v_i(t + 1)
   \]
7. If the maximum number of iterations is not reached or the accuracy requirement is not met, continue to perform steps (2) to (6).

2.2. Tent mapping reverse learning initialization population

The results show that initializing population with chaotic mapping is random, traversal and bounded, and can improve the search efficiency of the algorithm. Tent mapping has more homogeneous sequences than other mappings, so tent mapping is used to initialize particle swarms and to optimize the initial species through reverse learning strategies. By selecting good individuals for the next generation of learning, the search scope of the population is expanded, thus improving the search efficiency. The tent mapping mathematical expression is as follows:

\[
\begin{align*}
   x_{n+1}^n = \begin{cases} 
   2x_n, & 0 \leq x_n \leq \frac{1}{2} \\
   2(1-x_n), & \frac{1}{2} \leq x_n \leq 1 
   \end{cases}
\end{align*}
\]

The definition of reverse solution is: a feasible solution \( x = (x_1, x_2, ..., x_D) \) in D-dimensional space, \( x = [a, b] \), then the reverse solution is \( x = (x'_1, x'_2, ..., x'_D) \), where \( x'_i = a + b - x_i \).

The specific steps to initialize the population using tent mapping reverse learning are:

1. Using tent mapping to generate the location \( x_{ij} \) \((i = 1, 2, ..., D; j = 1, 2, ..., N)\) of \( N \) particle swarms in a given space as initial population \( OP \);
2. Generate reverse individual \( x'_{ij} \) corresponding to each individual \( x_{ij} \) in \( OP \) as reverse population \( FP \) according to the principle of reverse solution;
3. Combining \( OP \) and \( FP \), the fitness values of \( 2N \) individuals were sorted in ascending order, and the first \( N \) individual particles were selected as the initial population.
2.3. Simulated annealing algorithm

The simulated annealing algorithm was proposed by N. Metropolis in the mid-20th century. The main content is based on the fact that when a solid is heated, the particles in the solid will get a large amount of energy and move irregularly; when the heating ends and the solid is cooled, the particles in the solid can move regularly, and their energy reaches the equilibrium state at that temperature. Simulated annealing algorithm is often used to solve global optimal problems. The main ideas for finding the optimal solution are as follows:

1. The initial temperature $T$ is large enough, the initial solution is $x$, the current iteration number is $k = 0$, and the iteration number of each temperature is $\text{iter}_{\text{max}}$;
2. Make $k = 1, 2, ..., \text{iter}_{\text{max}}$, loop execution step (3), (4);
3. The solution is $x'$, and the energy difference $\Delta E = E(x) - E(x')$. If $\Delta f < 0$, then $x'$ is accepted as the new solution, that is, $x = x'$; otherwise, $p = \exp(-\Delta E/T)$ is calculated to make $x = x'$ the probability of $p$;
4. Terminate the loop if the condition is met. The condition for loop termination is that consecutive $x'$ is not accepted;
5. According to the cooling formula $T = d(T)$, a new temperature is obtained, and steps (2) to (5) are executed in a cycle until the Metropolis criterion is met.

2.4. Simulated annealing particle swarm optimization

Although the particle swarm optimization algorithm can search for the best location, it lacks the dynamic adjustment of speed and is prone to fall into local optimum because of poor handling of discrete optimization problems. Therefore, the combination of simulated annealing algorithm and particle swarm optimization algorithm makes up for a certain degree of deficiency, and can jump out of the current solution and continue to search for high-quality solutions. The main ideas of the simulated annealing particle swarm optimization algorithm are as follows:

1. Particle swarms are initialized using the tent reverse learning mechanism described above;
2. All particles of the particle swarm are calculated circularly;
3. Calculate the particle fitness $f_i$. Suppose that the best solution found by the particle itself is labeled $p_{\text{best}}$, if $f_i > p_{\text{best}}$, then $p_{\text{best}} = f_i$;
4. Select the particle with the best fitness value of all particles as $g_{\text{best}}$;
5. Initialize the temperature $T_0$ to be large enough, set the particle symbol to $p_i$, and cool the temperature according to the following formula:
   \[ T_0 = f(p_i)/\ln 5 \]
   \[ T_{k+1} = \lambda T_k \]  
   Where $\lambda$ is the attenuation parameter;
6. Determine the fitness value $\text{fit}(p_i)$ of particle $p_i$ at temperature $T$ according to the following formula:
   \[ \text{fit}(p_i) = \exp(-(f(p_i) - f(g_{\text{best}}))/T)/\sum_{i=1}^{N} \exp(-(f(p_i) - f(g_{\text{best}}))/T) \]  
7. Select a particle from all particles to determine its global optimal substitution value $p_i''$, and update the particle speed and position according to the formula (1) and (2);
8. Determine the final value of the particle, update the individual extreme $p_{\text{best}}$ and global extreme $g_{\text{best}}$;
9. Perform annealing operations;
10. If the termination operation is not reached, proceed to step (6)~ (9); otherwise, output the result.

3. Brief description of unmanned aerial vehicle problem

3.1. Environmental modeling

This paper studies the global path planning for unmanned aerial vehicles in known environments and establishes a three-dimensional digital elevation map model to simulate complex mountainous
environments. A cylinder is used to simulate the no-pass zone, which makes the optimization environment more complex. A straight line from the start to the end must pass through the no-pass zone. Therefore, it is our research institute to plan a least-cost route that does not cross the prohibited traffic zones.

4. Cost function design

\[
J_{\text{cost}} = w_1 J_{\text{length}} + w_2 J_{\text{height}} + w_3 J_{\text{smooth}}
\]

(7)

Where \(w_i \geq 0\) and \(\sum_{i=1}^{3} w_i = 1\). \(J_{\text{cost}}\) represents the total cost of a path, and \(J_{\text{length}}\), \(J_{\text{height}}\), and \(J_{\text{smooth}}\) represent the total cost of length, height, and smoothness, respectively. \(w_1\), \(w_2\) and \(w_3\) are the corresponding weights.

When each algorithm is used for path planning, the cost function is used to evaluate the quality of the generated path, which is also the basis for the iterative evolution of the algorithm population. The cost function determines the efficiency and quality of algorithm execution, and it is also a performance indicator of path planning. In order to better assess the quality of the route, this paper constructs a fitness function by considering the path's height cost, length cost and deflection angle size. Assume that each path consists of \(n\) points.

4.1. Path length cost

Route length is one of the most important indexes to evaluate the quality of a route. Aircraft carry limited fuel and power. The shorter the path, the less energy and time it takes to fly. The cost of introducing path length is as follows:

\[
J_{\text{length}} = \sum_{j=0}^{n} \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2 + (z_{j+1} - z_j)^2}
\]

(8)

4.2. Path high cost

The stable flight height of the UAV is also an important part of the UAV track planning process. For most aircraft, the flight height should not change much. Stable flight height helps to reduce the burden on the control system and save more fuel. Therefore, the cost of introducing track elevation:

\[
J_{\text{height}} = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (z_i - \frac{1}{n} \sum_{j=1}^{n} z_j)^2}
\]

(9)

4.3. Smoothness cost

When an unmanned aerial vehicle is making a turn, it needs some energy because of the air resistance. At the same time, this kind of operation also exerts some pressure on the body, the smaller the angle, the greater the pressure; moreover, the more energy consumption, the lower the flight efficiency. Therefore, the smoothness of flight is also a key factor in the cost of flight.

\[
J_{\text{smooth}} = (x_n - x_1) \sum_{j=1}^{n-2} \text{arccos} \left( \frac{\overrightarrow{\phi_{j+1} \phi_j} \cdot \overrightarrow{\phi_{j+1}}} {||\overrightarrow{\phi_{j+1}}|| ||\overrightarrow{\phi_{j}}||} \right)
\]

(10)

\[
\overrightarrow{\phi_j} = \left( x_{j+1} - x_j, y_{j+1} - y_j, z_{j+1} - z_j \right)
\]

(11)
5. Experimental simulation

To verify the effectiveness of the proposed algorithm, this paper conducts simulation experiments on UAV path planning under MATLAB 2018b environment. The starting and ending positions are (10,90) and (100,10), with coordinates of four cylindrical threat zones: (20,80), (40,50), (55,50), (100,30). The weighting coefficients $w_1$, $w_2$ and $w_3$ of the flight path cost function are 0.5, 0.3, 0.2, $c_1=1.5$, $c_2=2$, population number is 20, inertial weight $w=1$, iteration number is 400, respectively. Figure 1 shows the 3-D aircraft charts before and after improvement. In order to eliminate the occurrence of accidental events, each algorithm runs independently 10 times, and the average value of each optimization result is calculated. The average convergence route is shown in Figure 2.

Figure 1. Optimal roadmap of each algorithm
From Figure 1, it can be seen that the route planned by SAPSO is simple and safe, and the PSO optimal route works equally well. However, from Figure 2, it can be seen that the convergence effect of PSO is poor and precocious phenomena occur. These results show that PSO is random and easy to fall into the local optimal state. On the contrary, SAPSO has a good convergence curve and strong stability.

6. Concluding remarks
A simulated annealing particle swarm optimization algorithm is proposed for the poor convergence and strong randomness of particle swarm optimization algorithm. First, tent mapping reverse learning is used to initialize the population, then simulated annealing algorithm is used to refine the optimal solution and enhance the optimization ability of the algorithm. From the results of three-dimensional route planning for complex terrain, SAPSO has strong route planning ability and good stability.

Acknowledgements
This work was financially supported by Regional foundation of the National Natural Science Foundation of China (62062037).

References
[1] Chen Tianpei, Wang Yuhui, Wu Qingxian, Zhou Zeyu. Three-dimensional path planning based on fuzzy logic particle swarm optimization [J]. Electro-optical control, 2020,27(06): 1-5.
[2] Chen Shiming, Xie Jing, Chen Wendong, Fang Huajing. Three-dimensional spatial path planning based on HPSO algorithm [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2013,41(02): 109-113+119.
[3] Wang Zhendong, Zeng Yong, Wang Junling, Hu Zhongdong. Intrusion detection based on BP network with beetle swarm optimization [J]. Science and Technology, 2020,20(32): 13249-13257.
[4] Wang Zhendong, Liu Yaodi, Yang Shuxin, Wang Junling, Li Dahai. Network intrusion detection based on beetle swarm optimization and improvement of Regularized Extreme Learning Machine [J/OL]. Journal of Automation: 1-20 [2021-02-07].
https://doi.org/10.16383/j.aas.c190851.
[5] Fu Xingwu, Hu Yang. Three-dimensional path planning based on Improved Particle Swarm Optimization [J/OL]. Electro-optical control: 1-5 [2021-02-07].
Http://kns.cnki.net/kcms/detail/41.1227.TN.20201201.1055.016.html.
[6] Lu Xin, Mu Xiaodong, Zhang Jun, Wang Zhen. Chaotic sparrow search optimization algorithm
[7] Song Xiaoyu, Gao Minghai, Zhao Ming. Hybrid artificial bee swarm algorithm with adaptive search strategy [J]. Computer Engineering and Application, 2019, 55 (22): 53-59+85.

[8] Liao Weilin, Cheng Shan, Shang Dong, Wei Zhaobin. Multi-strategy particle swarm optimization algorithm [J/OL]. Computer engineering and application: 1-11-10 [2020-08-04]. Http://kns.cnki.net/kcms/detail/11.2127.TP.20200617.1330.024.html

[9] Huang Chencheng, Wei Xia, Huang Deqi, Ye Jiahao. Shuffled frog leaping grey wolf algorithm for solving high dimensional complex functions [J]. Control theory and application, 2020,37(07): 1655-1666.

[10] Li Yang, Li Weigang, Zhao Yuntao, Liu Ao. Grey Wolf Optimizer based on Levy's flight and random walk strategy [J]. Computer Science, 2020,47 (08): 291-296.

[11] YongBo, Chen, et al. "Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm." Neuro computing 266 (2017): 445-457.