Proceedings of the 3rd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/25/20240976

UAV path optimization algorithm and simulation case analysis

Junyue Hu

Jinan University-University of Birmingham Joint Institute, Jinan University, Guangzhou, China

jxh1516@student.bham.ac.uk

Abstract. Drones are much accounted for, with the advantage of low cost, flexibility, and easy-controlled. As its application fields become more extensive, from human production to scientific research, the requirements for UAV path planning are getting higher. Reasonable path planning minimizes resource consumption and is worthy of continuous exploration. This article lists one algorithm each among heuristic algorithms, graph search algorithms, and traditional algorithms: ant colony optimization (ACO), A* algorithm, and artificial potential field. After a discussion of the process and steps, a simulation of three algorithms and a comparison has been made. ACO focuses more on path planning with disordered points, while the A* algorithm and artificial potential field do well in avoiding obstacles when traveling with a given starting point and destination. More strengths and weaknesses are explained later. Path planning still needs more research to adapt to current needs, such as emergencies, dynamic changes in actual situations, consideration of difficult terrain and climate, and weakness of signal connection.

Keywords: Path planning, ACO, A* algorithm, artificial potential field, obstacle avoidance.

1. Introduction

Nowadays, drones are able to finish some tasks that are difficult to operate manually, such as bridge and tunnel damage detection, identification of dropped objects from high altitudes, aerial photography, and taking a significant role in rescue and geological detection in the scientific field. Due to the complexity of the flight environment, the actual flight path often deviates from the pre-flight planned path, and the UAV needs to be more realistic and make temporary adjustments.

Although it is inevitable to improve UAV information-sensing technology, pre-flight route planning must be continuously optimized and consider multiple factors. Hence, this paper makes a conclusion about frequently-used methods in UAV path planning with examples following and modified some basic methods.

2. Theoretical framework

2.1. Ant colony optimization (ACO)

An intelligent bionic optimization method known as ACO was created by Dorigo M. et al. based on the foraging behavior traits of ant groups in nature. For other colony members to follow, these ants leave pheromones on the ground to indicate a good path. ACO has been tested on the TPS first, and the result that ACO solves routing problems is highly recognized. This study chose the Ant colony system (ACS) over the Ant system (AS) for path planning optimization because ACS introduces offline pheromone updates (PU) in addition to local PU that is introduced at the conclusion of the build phase. A model $P = (S, \Omega, f)$ of the combinatorial optimization problem includes: a search space S defined on a finite set of discrete decision variables $X_i, i = 1, \ldots, n$; a set of Ω constraints between variables; and the objective function $f: S \rightarrow \mathbb{R}_0^+$ to be minimized.

The generic variable X_i takes values in $\{v_i^1, \ldots, v_i^n\}$.

The solution component $c_{ij} = (i, j)$ indicates that the solution after analyzing and site i must be followed by site j immediately. The pheromone value τ_{ij} is associated with edge joining site i and j of the solution component c_{ij}, which conclude the assignment $X_i = v_i^j$. Clearer vehicle routing is achieved by fully connected construction drawings $G_c(V, E)$, where C is the collection of all possible solution components, V and E is a set of vertices and edges respectively. To contact each vertex once, the vehicle can only move from one to the next. The quality of the solution found may affect the extent $\Delta\tau$ to which pheromone is deposited.

The algorithm flow is as shown below.

$$\begin{gathered} Set\ parameters, initialize\ pheromone\ trails \\ \boldsymbol{While}\ termination\ condition\ not\ met\ do \\ Construct\ ant\ solutions; \\ Apply\ local\ search; \\ Update\ pheromones; \\ \boldsymbol{endwhile}. \end{gathered} \tag{1}$$

The working steps are initialization, constructed solution, and update phase. The most important and fundamental component of AOS iterative operation is solution creation. Its primary goal is to build potential solutions in the problem domain using state transition laws. To develop a complete path PU during path planning, solution construction primarily chooses the next path point in accordance with the state transition rule. After the solution is constructed, you must carry out the PU procedure, which consists of two parts. (1) Pheromone volatilization, which is utilized to lessen the pheromone on the path, will affect future ant behavior and improve the algorithm's capacity for exploration; (2) Pheromone release: As ants move along a path, they release informational components [1, 2]. This raises the likelihood that the ant will select this route again in the future, improving the algorithm's capacity for improvement. Until the termination condition is satisfied, solution creation and pheromone updating are repeated [3].

The probability of going to point j from point i is given by:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum \tau_{il}^\alpha \cdot \eta_{il}^\beta}, & if\ c_{ij} \in N(s^p) \\ 0, & otherwise \end{cases}, \tag{2}$$

where $N(s^p)$ is the set of feasible components; that is, edges (i, l) where l is a city about to be visited by the ant k from i. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij}, which is given by the equation:

$\eta_{ij} = \frac{1}{d_{ij}}$, where $d_{ij}$is the distance between cities i and j.

In this method, τ_{ij} is different to the value in AS with offline pheromone update:

$$\tau_{ij} \Leftarrow \begin{cases} (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0, & if\ (i, l)\ belongs\ to\ best\ tours \\ \tau_{ij}, & otherwise \end{cases}, \tag{3}$$

Please follow these instructions as carefully as possible so all articles within a conference have the same style to the title page. This paragraph follows a section title so it should not be indented.

DOI: 10.54254/2753-8818/25/20240976

2.2. *A* algorithm*

The second algorithm is the A* algorithm.

There are two famous algorithms related to A* algorithm, which need to be mentioned first. That is Dijkstra's algorithm and breadth-first search algorithm. But either of it have some weakness and A star algorithm highly combine these two algorithms. The A-Star algorithm not only considers the actual distance cost value $g(x)$ between the current node and the starting point, but also considers the estimated cost value $h(x)$ between the node and the target point. This is also where the A-Star algorithm is more intelligent than these two algorithms and is considered to be one of the most effective methods for path search [4]. Manhattan distance is used to calculate the length instead of Euclidean distance in the grid [5].

Define the cost function $f(n)$ as the sum of these two cost values. The path formed by connecting the track nodes corresponding to the minimum total cost value is the globally optimal path. The cost function $f(n)$ is defined as follows: $f(n) = g(n) + h(n)$ [5]. It consists of two parts of the cost value $g(n)$ and $h(n)$, n represents the current node position; the cost of the visited node, denoted by the symbol $g(n)$, is the sum of the distances from the starting position S to the current position n; $h(n)$ represents the total cost estimate of the distance from the current position n to the target point G (the cost of the node to be visited) the calculation method of cost value and $h(n)$ is obviously a little different [6]. When calculating $h(n)$, it is needed to ignore the obstacles in the grid map because it does not represent the real value, but A heuristic estimate, tentative in nature. It is a heuristic and heuristic, so it is also called the heuristic function of the A-Star algorithm.

2.3. *Artificial potential field (APF)*

Based on the electrostatic particle interaction, the proposed APF algorithm. Similar particles' charges will produce a repulsive force, while opposite particles' charges will produce an attractive force. In order to create an attracting force, place the opposite charges of the particles at the beginning and destination points. The barriers and the UAV (unnamed aerial vehicle) are supposed to have the same charge in order to offer a collision-free and goal-reaching path. The obstruction produces repelling force under these circumstances. The following is the model's equation.

For basic charges: $F = -\frac{kq_1q_2}{r^2}$, where: k is the interactions constant; q_1 and q_2 are the electric charges of the particles; and r is the distance between the particles.

For overall force towards to UAV:

$$\vec{F} = \vec{F}_{goal} + \vec{F}_{obstacles}, \tag{4}$$

$$\vec{F}_{goal} = \frac{k_{VG}q_Vq_G}{|\overrightarrow{r_{VG}}|^2} \cdot \frac{\overrightarrow{r_{VG}}}{|\overrightarrow{r_{VG}}|}, \tag{5}$$

$$\vec{F}_{obstacles} = -\sum \begin{cases} \frac{k_{VO}q_Vq_O}{|\overrightarrow{r_{VO_i}}|^2} \cdot \frac{\overrightarrow{r_{VO_i}}}{|\overrightarrow{r_{VO_i}}|}, if\ |\overrightarrow{r_{VO_i}}| < d. \\ 0, otherwise \end{cases} \tag{6}$$

For the next direction and linear velocity of the UAV:

$$\theta^* = \arctan\left(\frac{F_w^y}{F_w^x}\right), \text{ where: } \theta^* \text{ is reference UAV's orientation.} \tag{7}$$

$$V^* = \mathrm{Limit}\left(|\overrightarrow{F_w}|, V_{max}\right), \tag{8}$$

where: V^* is reference UAV's linear velocity; and V_{max} is maximum UAV's linear velocity allowed; d is the distance of influence of the repulsive field. But in reality, the path of UAV might be restricted by many facts, such as UAV limited dynamics. It is not possible to change the UAV orientation rapidly. To prevent UAV from travelling near the obstacles, direction error needs consideration. In this case, the linear velocity equation is modified.

$$\alpha = \frac{\theta_{error}^{max} - \mathrm{abs}(\theta_{error})}{\theta_{error}^{max}}, \tag{9}$$

$$V^* = \begin{cases} \text{Limit}\left(\alpha, \left|\overrightarrow{F_w}\right|, V_{max}\right), if\ abs(\theta_{error}) \leq \theta_{error}^{max} \\ 0, otherwise \end{cases}. \tag{10}$$

And it needs to check the curvature k:

$$g\sqrt{n^2 - 1}/v^2 \leq k_{\max} = 1/R_{\min}. \tag{11}$$

E total energy function obtained by the above formula is:

$$E(t) = \int \left(mv(t)a(t) + I\omega(t)\beta(t) + \frac{2f_v}{r}v(t) + B\right)dt + P_s t, \tag{12}$$

where: m is UAV mass; I is UAV moment of inertia; f_v is a viscous friction coefficient; B is fixed power provided for the motors to overcome the static friction; P_s is other energy consumed for real-time detection, analysis and calculations ; $v(t)$ and $\omega(t)$ are linear velocity and angular velocity respectively; $\alpha(t)$ and $\beta(t)$ are linear and angular acceleration respectively; From the depicted equation, one can see that linear and angular velocity and acceleration have an impact on the terminal value of this complex energy equation [7].

But the path planning only by the attraction of the target point, it might be trapped in dilemma. Hence, based on the traditional APF, the attraction effect of the target point on the moving body relatively is weaken. To strengthen the coherence of path adjustment, the algorithm in this paper regards the pre-planned trajectory as consisting of continuous particles, and the gravity is provided by adjacent particles. The specific implementation is to insert one particle per unit length between adjacent pre-planned track points, and the path adjustment is reflected by the movement of the particle.

$$\vec{F}_{goal} = \frac{k_{VA}q_V q_A}{|\overrightarrow{r_{VA}}|^2} \cdot \frac{\overrightarrow{r_{VA}}}{|\overrightarrow{r_{VA}}|}. \tag{13}$$

The direction is that the moving body points to the gravitational source, which is the adjacent particle [8].

3. Simulation

3.1. Ant colony optimization (ACO)

For ACO, there is a typical example, the travel restrictor problem (TSP problem) [2]. Consider a journey in which the participant is required to return to the original place of departure after visiting each of the 31 provincial capital cities on the itinerary only once. The path requirements are all paths. Given the latitude and longitude of each city, the optimized path with the least length can be figured out.

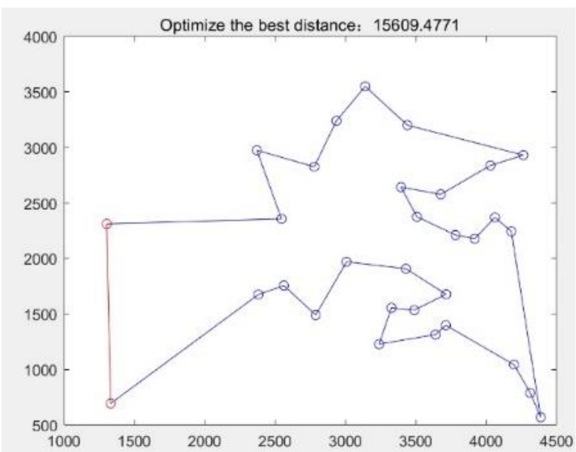

Figure 1. Path planning result by ACO.

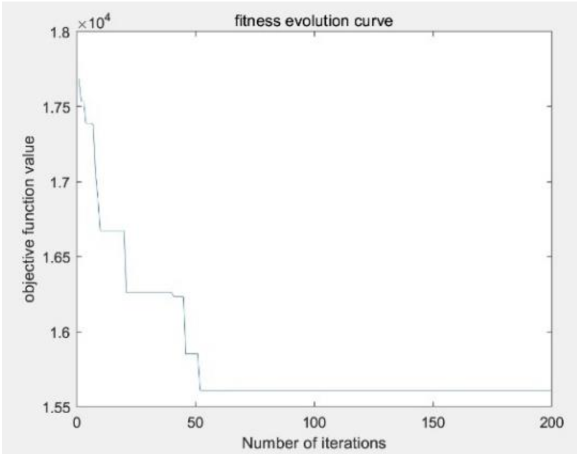

Figure 2. Path length changes with algorithm iteration times.

3.2. *A* algorithm*

For the A star algorithm, the goal point, beginning point, and obstacles are set randomly. Then these two cases are solved. figure 1 greatly tells that the A star algorithm focuses more on the overall situation. It will not form a local optimal solution, but is flexible and highly optimizable. Starting from the green point and finding the optimal path to the red point endpoint, the black square obstacles in the grid is needed to avoid. The pictures below show randomly set obstacles and starting and ending points. The iterative search process progresses from dark blue to red in the grid.

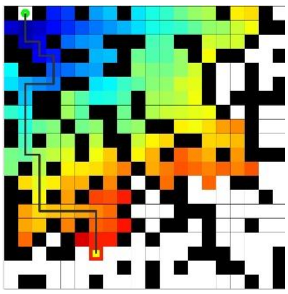

Figure 3. Result of how the A*algorithm works.

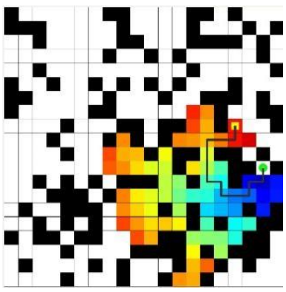

Figure 4. Result of how the A*algorithm works.

3.3. *APF*

The search environment is set to a 200×200 grid, with 5 threats each from radar, missiles, and anti-aircraft guns randomly generated. The UAV wants to fly from (-90, -90) to the (90, 90) coordinate point at a constant speed. Set the attractive field constant K_a is 2.5, the repulsion field constant K_b is 5.4. The minimum safe distance should be a small enough value relative to the threat radius, here set to d=0.1

[8]. This result is figured out by the modified function of the APF. Otherwise, the SAV will be trapped between the two biggest circles in the middle.

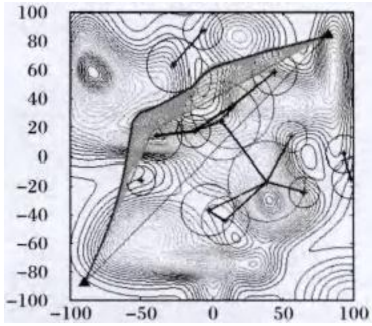

Figure 5. Simulation results of APF [8].

3.4. Compare

After viewing the examples above, a summary can be made. The three algorithms represent three categories of path optimization models. ACO model, A star algorithm, and artificial potential field method are respectively heuristic optimization model, graph search algorithm and traditional model [9]. They all fall under the category of path planning, but ACO focuses on the shortest connection arrival time or the shortest path length over a number of places. The artificial potential field technique and the A star algorithm can both be utilized to avoid obstacles. Moreover, the artificial potential field is useful when dealing with graphical data, such as environment maps. Although the A* algorithm has the benefit of global optimization, it increases the amount of calculation and sometimes causes transition situations. With large amounts of data, the calculation speed is low [10].

4. Conclusion

Listing three methods from different categories, simulating and comparing, this paper shows ACO contributes to path planning of multiple passing points, while A*star algorithm and artificial potential are good at path planning with obstacle avoidance. The above methods are all relatively basic path optimization methods. However, the flight environment is more complex, with factors such as wind speed, direction, and terrain changes. More comprehensive optimization models need to be produced. Drone flying is increasingly popularized and used in human and scientific fields such as disaster relief and surface exploration. Continuous improvement and optimization of algorithms require constant attention.

References

[1] Dorigo M 1991 Proceedings of the First European Conference on Artificial Life Paris France
[2] Ying W, Jianying X 2002 J. Sys. Simul. 01 31-33
[3] Songcan Z, Jiexin P, Yanna S 2020 Compu. Eng. and App. 56 08 10-19
[4] Yadong L. 2020 South China Uni. of Tech.
[5] Yongming B, Xiaoyang M, Fei G 2019 Mechatronics 25 06 9-15
[6] Xing Y, Xu Z, Shixu C 2018 Proceedings of the Fifth High-Resolution Earth Observation Academic Annual Conference 585-597
[7] Szczepanski R, Tarczewski T, Erwinski K 2022 IEEE Access 10 39729-39742
[8] Jiaru D, Changping D, Yao Z 2016 Compu. App. 36 01 287-290
[9] Qingjian L, Lisheng S, Gang L 2023 Prog. in Aeronautical Eng. 14 02 24-34
[10] Shuilin G 2021 Civil Aviation Uni. of China