Automation and Machine Learning (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/autml.2024.050110
ISSN 2516-5003 Vol. 5 Num. 1

Quadcopter UAV Trajectory Planning Based on Improved Dung Beetle Algorithm

Chuande Xu[1,a], Su Xu[1,b,*], Guangyu Du[1,c], Wenxuan Cui[2,d]

[1]School of Electronic Engineering, Jiangsu Ocean University, Lianyungang, Jiangsu, 222000, China
[2]School of Mechanical Engineering, Jiangsu Ocean University, Lianyungang, Jiangsu, 222000, China
[a]2023220630@jou.edu.cn, [b]1995000021@jou.edu.cn, [c]2023220621@jou.edu.cn, [d]2022210501@jou.edu.cn
[*]Corresponding author

Keywords: Dung Beetle Algorithm, Levy Flight, Chaotic Mapping

Abstract: Aiming at the problems of dung beetle algorithm in trajectory planning, such as easy to fall into local optimum, long flight distance and high energy consumption, this paper proposes an improved dung beetle algorithm for trajectory planning of quadrotor UAV. The algorithm introduces chaotic mapping, changes the probability distribution of the initialised population, and further proposes to introduce Lévy flights at the greedy dung beetle update position to improve the convergence speed of the algorithm. Finally, by comparing the Keplerian optimisation algorithm, the lemur optimisation algorithm, the dung beetle algorithm and the improved dung beetle algorithm for trajectory planning in two environments, the experimental results show that the algorithm proposed in this paper has the advantages of faster convergence speed, shorter flight distance, and is not easy to fall into the local optimum.

1. Introduction

The problem of trajectory planning is one of the key research areas for UAVs. The UAV [1] trajectory planning problem can be regarded as a multi-constraint optimisation problem. More and more optimisation algorithms are applied to UAV trajectory planning in order to break through the optimisation problems such as its shortest trajectory, lowest energy consumption and shortest flight time. In recent years, the algorithms used for UAV trajectory planning are mainly divided into two categories: traditional algorithms [2] and intelligent algorithms [3].

Improved UAV trajectory planning with artificial potential field method is proposed in Literature 4 [4], which introduces a collision factor to avoid unreasonable obstacle avoidance manoeuvres and shorten the length of the UAV flight path. A hybrid improved symbiotic organism search and sine-cosine particle swarm optimisation method for UAV path planning is proposed in Literature 5, which firstly employs chaotic logic mapping to improve the diversity of the initial population. Then, a difference strategy, a novel decay function and a population regeneration strategy are introduced to improve the performance of the algorithm [5]. A UAV path planning method based on improved genetic algorithm is proposed in the literature 6, where the parameters of the GA crossover and mutation steps are modified using PSO to drive the trajectory planning to optimality [6]. Article 7 proposes a VPF-RRT* algorithm to plan the planning path; secondly, an anti-environmental perturbation method based on Deep Recurrent Neural Network PI (DRNN-PI) controllers is proposed to enable the USVs to eliminate the environmental perturbations and maintain their trajectories along the planning path [7]. Article 8 proposes a random tree algorithm based on a potential field orientated greedy strategy applied to UAV path planning, which employs a potential field as an aid to the random tree expansion process. It rationally triggers the greedy strategy based on the principle of field-strength descent gradient optimisation, which accelerates the expansion process of the random tree to better regions and shortens the path search time [8]. Article 9 proposes a DP-DDPG path planning algorithm applied to UAVs in complex environments. The convergence speed is improved by refining the state and action space of the deep reinforcement learning task and designing the reward function in combination with the artificial potential field method [9].

The Dung Beetle Algorithm (DBO) [10], as a new intelligent algorithm, is mainly inspired by the dung beetle's rolling, foraging, dancing, reproduction and greedy behaviour. Because of its expandability, several scholars have based on this algorithm for engineering applications, such as trajectory planning, photovoltaic cell parameter identification, aerial generator noise reduction, etc., which fully reflects the advantages of this algorithm in engineering applications such as strong applicability and high expansion rate.

In this paper, the dung beetle algorithm is improved in the following ways to address the shortcomings of the quadrotor UAV trajectory planning, such as long flight distance, high energy consumption, and easy to fall into the local optimum: the introduction of the Tent mapping initialisation population reduces the probability of local optimum, and enhances the algorithm's global searching ability, and the introduction of the Levy flight improves the greedy dung beetle position formula to improve the convergence accuracy. The superiority and effectiveness of the improved dung beetle algorithm are verified by simulation experiments.

2. UAV Trajectory Modelling

2.1. Environmental Modelling

The trajectory environment modelling is divided into baseline mountainous terrain, threat areas, and no-fly areas. The flight area is set as a 100m*100m*200m three-dimensional space area, and the terrain function and the mountain peak environment formula is shown in equation 1:

$$Z_b(x,y) = \sin(y+q) + w\cdot\sin(x) + e\cdot\cos(r\cdot\sqrt{(x^2+y^2)} + t\cdot\cos(y) + u\cdot\sin(o\cdot\sqrt{(x^2+y^2)}) \tag{1}$$

Formula 1, x and y are the mapped coordinates of the model in the horizontal datum, $Z_b(x,y)$ is the height value of the mapped coordinate. q,w,e,r,t,u,o are constant coefficients, different constant coefficients respond to different baseline terrain.

The threat area is the area where there are threat objects such as bird flocks, power lines, etc. while flying in the air, and the modelling formula is as in equation 2:

$$Y_i(x,y,z) = \begin{cases} \sum_i (x-x_i)^2 + (y-y_i)^2 = \text{thread}_{i(x,y)}{}^2 \\ z = \text{thread}_h = N^+ \end{cases} \tag{2}$$

Formula 2, $Y_i(x,y,z)$ indicates the coordinates of the location of the obstacle, $\text{thread}_{i(x,y)}$ and

$thread_h$ indicates the radius and height of the threat.

No-fly zones are areas where drones are restricted in the air, such as airports, state organ departments, and important military facilities. The modelling formula is shown in equation 3:

$$C_k(x,y,z) = \begin{cases} \sum_k (x-x_k)^2 + 1/4(y-y_k)^2 = no-fly_{k(x,y)}{}^2 \\ z = no-fly_h \ge 0 \end{cases} \tag{3}$$

Formula 3, $C_k(x,y,z)$ indicates the coordinates of the location of the no-fly zone, $no-fly_{k(x,y)}$ and $no-fly_h$ indicates the coordinates and altitude of the centre of the no-fly zone.

In summary, the modelling results are shown in Figure 1 Where (x,y,z) represents the size and height values of the map, respectively.

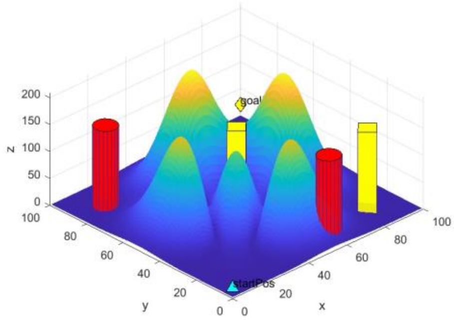

Figure 1: Environmental modelling.

2.2. Trajectory Constraints

UAVs often need to consider relevant flight constraints during flight, mainly including the height above ground, turning angle and the maximum value of climb/dive angle.

(1) Height above ground

The UAV needs to fly at an appropriate altitude, if the flight altitude is too low the UAV is easy to collide with the valley and the ground small size target; if the flight altitude is too high, the UAV will be out of control due to the low barometric pressure or high wind speed. Therefore, the height above ground is constrained as in equation 4:

$$H_{min} \le H_k \le H_{max} \tag{4}$$

Formula 4, H_{min} and H_{max} is the minimum and maximum altitude of flight.

(2) Turning angle

When the UAV is about to encounter an obstacle or near a no-fly zone, it needs to make a certain turn to avoid obstacles, but the size of the turn angle value often depends on the energy consumption and the threat of secondary collision. Therefore, the turn angle is constrained as in equation 5:

$$\alpha = \arccos\left(\frac{\overrightarrow{AB} * \overrightarrow{BC}}{\left\|\overrightarrow{AB}\right\| * \left\|\overrightarrow{BC}\right\|}\right)$$

$$\theta = \begin{cases} -\theta_{max}, \text{if} \alpha < -\theta_{max} \\ \alpha, \text{if } -\theta_{max} \le \alpha < \theta_{max} \\ \theta_{max}, \text{if} \alpha > \theta_{max} \end{cases} \tag{5}$$

Formula 5, $\overrightarrow{AB}$ and $\overrightarrow{BC}$ indicates two flight trajectory lines at corners, $\alpha \in [-\theta_{max}, \theta_{max}]$ is the angle of the trajectory line.

(3) Maximum value of climb/dive angle

Since the UAV has the ability to take off and land vertically, it is only necessary to consider the flight process due to the change of the angle value to produce jitter, resulting in unnecessary losses. The constraint formula is shown in equation (6):

$$\frac{|Z_k - Z_{k-1}|}{|\alpha_k|} \le \tan\theta_{max} \tag{6}$$

Formula 6, θ_{max} is the maximum value of the climb/dive angle; $|Z_k - Z_{k-1}|$ is the difference between heights; $|\alpha_k|$ is the horizontal projection of the kth track.

2.3. Fitness Function

The UAV trajectory planning problem is bounded by a large number of criteria and needs to consider multiple factors, mainly its trajectory flight cost, flight environment constraints and dynamic collision cost, and the fitness function is the sum of the three, as in Equation 7:

$$C = L_c + E_c + B_c \tag{7}$$

Formula 7, L_c is the trajectory flight cost; E_c is the dynamic collision cost; B_c is the flight environment constraint.

3. Dung Beetle Algorithm

3.1. Initial Dung Beetle Algorithm

The Dung Beetle Algorithm (DBO) serves as an intelligent algorithm that simulates the process of rolling, foraging, dancing, reproduction and greedy behaviour of dung beetles . The Dung Beetle Algorithm consists of four different strategies to update the position according to the function of each strategy, the update strategies are as follows:

(1) Ball rolling dung beetle

Dung Beetle encounters two scenarios with and without obstacles while performing the ball rolling behaviour:

1) When in an obstacle-free region, the dung beetle rolling path is affected by the intensity of the light source, and the position update formula for the rolling dung beetle during ball rolling is given in Equation 8:

$$x_k(t+1) = x_k(t) + \beta \cdot i \cdot x_k(t-1) + s \cdot \triangle x$$
$$\triangle x = |x_k(t) - x_{bad}| \tag{8}$$

Formula 8, $x_k(t+1)$ denotes the position of the k dung beetle at $t+1$ iteration, β is a natural number of 1 or -1, $i \in (0, 0.2]$ denotes a random deflection constant, s takes a random value in (0,1), $\triangle x$ is the intensity of the simulated light source, and x_{bad} is the global worst dung beetle position.

2) When in an area with obstacles, the ball-rolling dung beetle uses a tangent function to simulate the dancing behaviour of the dung beetle to obtain a new rolling direction, and its position update formula is shown in Equation 9:

$$x_k(t+1) = x_k(t) + \tan(\theta)|x_k(t) - x_k(t-1)| \tag{9}$$

Formula 9, θ between $[0, \pi]$.

(2) Breeding dung beetles

In nature dung beetles roll their dung balls to a place where they can safely reproduce their offspring and store them, so a boundary selection strategy was used to simulate female dung beetles searching for a safe territory to reproduce their offspring, and the selection formula is shown in equation 10:

$$Lb^* = \max\left(x_{local} \cdot (1-R), Lb\right)$$
$$Ub^* = \min\left(x_{local} \cdot (1+R), Ub\right) \tag{10}$$

Formula 10, Lb^* and Ub^* denote the lower and upper limits of the breeding region, x_{local} is the local optimal position, $R = 1 - t / T_{max}$, T_{max} denotes the maximum number of iterations set, Lb and Ub are the lower and upper limits, respectively.

As shown in Figure 2, the current local optimal position x_{local} is indicated in orange, the black circles near x_{local} indicate the breeding balls, each of which has a dung beetle pup, and the ellipses are the upper and lower limits of the region. If the dung beetle determines the breeding region, the female dung beetle will choose to breed in this region. For the dung beetle algorithm only one brood ball will hatch per breeding, so the breeding dung beetle position is dynamically updated during each iteration as in equation 11:

$$x_k(t+1) = x_{local} + b_1 \times (x_k(t) - Lb^*) + b_2 \times (x_k(t) - Ub^*) \tag{11}$$

Formula 11, $x_k(t)$ denotes the position of the k female dung beetle at the t iteration, b_1 and b_2 are denote two D dimensional random numbers.

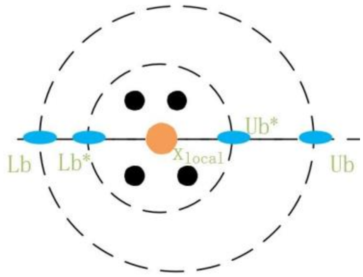

Figure 2: Breeder's position.

(3) Foraging dung beetle

The newly grown baby dung beetles need to forage in the foraging area, then the optimal foraging area needs to be defined, the definition formula is as in equation 12 and the position update formula is as in equation 13.

$$\begin{aligned} Lb^* &= \max(x_{global} \cdot (1-R), Lb) \\ Ub^* &= \min(x_{global} \cdot (1+R), Ub) \end{aligned} \tag{12}$$

$$x_k(t+1) = x_k(t) + C_1 \times (x_k(t) - Lb^*) + C_2 \times (x_k(t) - Ub^*) \tag{13}$$

Formula 12 and 13, x_{global} is the global best position, C_1 is denoted as a random value obeying a normal distribution, and C_2 belongs to a random value of $[0,1]$.

(4) Greedy dung beetles

Some dung beetles steal dung balls from other dung beetles for their own food, and from equation 12, it can be concluded that x_{global} is the global best food location, therefore, under each iteration, the greedy dung beetle location is updated as in equation 14.

$$x_k(t+1) = x_{global} + C \times V \times \left(\left| x_k(t) - x_{local} \right| + \left| x_k(t) - x_{global} \right| \right) \tag{14}$$

Formula 14, V is a constant value and C is a D-dimensional random value obeying a normal distribution.

3.2. Improving the Dung Beetle Algorithm

The traditional dung beetle algorithm, which uses a random distribution method for population initialisation, this treatment brings uncertainty to the performance of the algorithm; although the dung beetle algorithm has the performance of quickly arriving at an optimal solution, it also suffers from the defect of local optimality. Various strategies will be introduced to improve the above deficiencies.

3.2.1. Tent Chaos Mapping

Chaotic mapping is a kind of nonlinear dynamical system, which is characterised by deterministic and stochastic properties compared to the traditional random distribution. In population initialisation, chaotic mapping generates a larger range of solutions with more uniform distribution. In this paper, Tent chaotic mapping is used, Tent chaotic mapping, as a kind of chaotic mapping species, has the characteristics of collapsibility, randomness and simplicity, which improves the convergence speed of the algorithm and the accuracy of the solution. The distribution of its chaotic values is shown in Fig.3 and the frequency map of chaotic values is shown in Fig.4, and the formula of Tent chaotic mapping is shown in equation 15.

$$x_{k+1} = \begin{cases} \frac{x_k}{\alpha} & ,0 < x_k \le \alpha \\ \frac{(1-x_k)}{(1-\alpha)} & ,\alpha < x_k \le 1 \end{cases} \tag{15}$$

Formula 15, α takes the value 0.49.

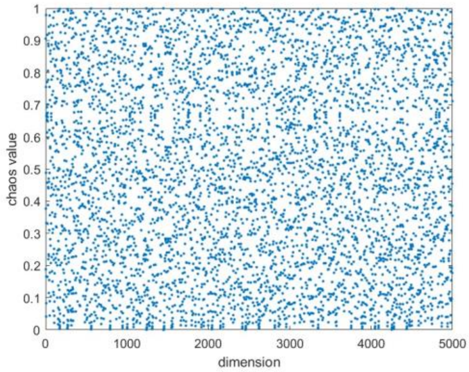

Figure 3: Chaos map.

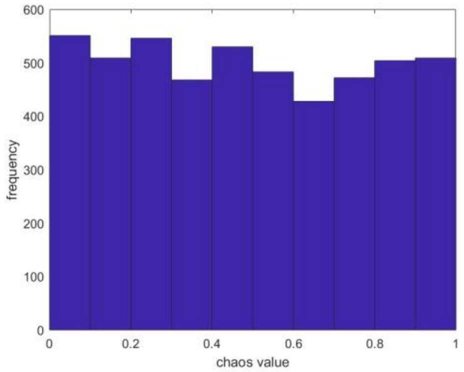

Figure 4: Frequency diagram of chaotic values.

3.2.2. Levy Flight

Levy Flight is a stochastic search method with a step size that conforms to the Levy distribution. It allows for a better global search capability by mixing the search between short and longer distances. In greedy dung beetle position updating, the use of Levy flight to generate random step sizes allows for a wider search range, improves the diversity of population search, and helps to increase the speed of convergence. The position update formula is as follows:

$$x_k(t+1) = levy \cdot x_{global} + c \cdot v \cdot \left(|x_k(t) - x_{local}| + |x_k(t) - x_{global}| \right) \tag{16}$$

3.3. XDBO Flowchart

The XDBO flowchart is shown in the figure 5:

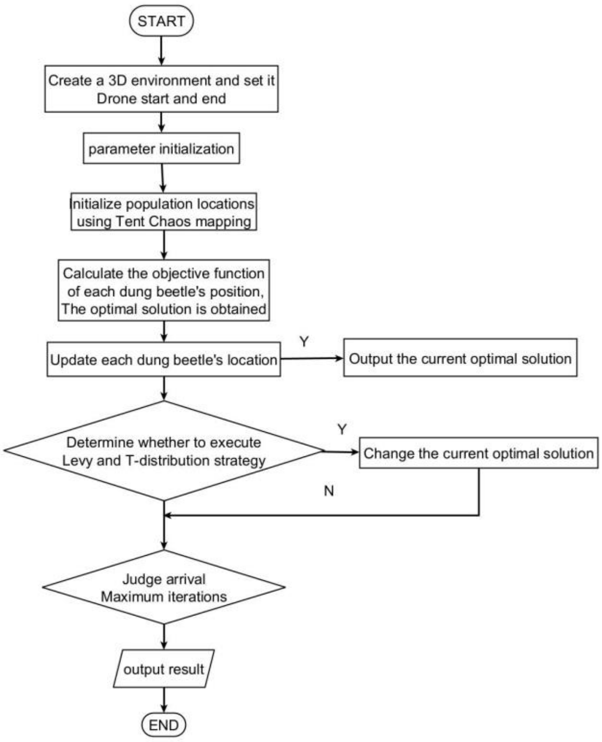

Figure 5: XDBO Flowchart.

4. Simulation Experiments and Results

In this section, the XDBO algorithm is applied to UAV trajectory planning in a mountainous environment. Constraints are established and the environment is modelled using MATLAB. The UAV in mountainous environment will be compared for trajectory planning under four algorithms, KOA, LO, DBO and XDBO, to verify the performance enhancement of XDBO in trajectory planning. The starting and ending points of UAVs in mountainous environments are (0,0,20) and (100,100,20), and the aircrafts are set to be consistent in the size settings of the no-fly zone or threat zone, and the two differ in the number of peaks and slopes of the mountains. All algorithms have a population size of 70 and an iteration number of 100.

The results and convergence curves of trajectory planning for mountainous areas using the above parameters are shown in Figures 6 and 7. In Fig. 6, it can be seen that all four algorithms reach the target by avoiding no-fly zones, threat zones and obstacle peaks. However, in the mountainous environment, the DBO flight trajectory is too large, the LO algorithm and the KOA algorithm do not fluctuate much, on the contrary, the XDBO flight trajectory is moderate and keeps a safe distance from the peaks and threatening areas. Figure 7 shows that except for the XDBO algorithm, the other algorithms tend to fall into the local optimum in the early stage, while the XDBO algorithm quickly searches for the optimal value and finds the best route at the beginning of the iteration.

In summary, the algorithm in this paper has a faster convergence speed, can avoid the problem of local optimum, and can also find the best route in complex environments.

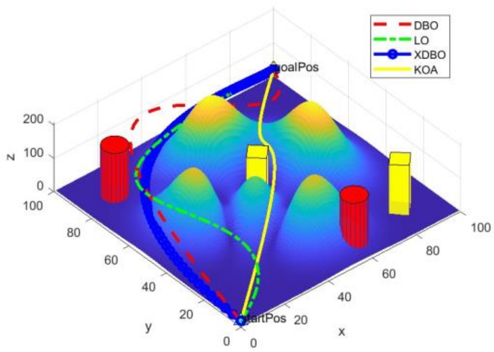

Figure 6: Route planning.

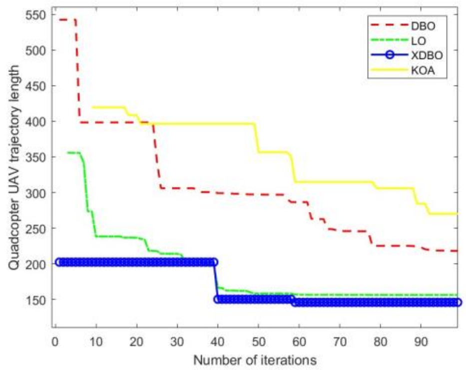

Figure 7: Convergence curve.

5. Conclusion

Aiming at the UAV trajectory planning problem, this paper improves the Dung Beetle algorithm and applies it to the trajectory planning of a quadrotor UAV in a three-dimensional environment. Tent chaotic mapping is introduced into the DBO algorithm, which changes the diversity of the initial population; Lévy flight is introduced to enhance the convergence speed of the algorithm. By comparing the algorithm's trajectory planning in mountainous environments, the experimental results show that the improved Dung Beetle algorithm's performance in trajectory planning is substantially improved. The cooperative flight of multiple UAVs will be further investigated in the future.

References

[1] Gugan G, Haque A. Path planning for autonomous drones: Challenges and future directions[J]. Drones, 2023, 7(3): 169.

[2] Huang X, Liu Y, Huang L, et al. BIM-supported drone path planning for building exterior surface inspection[J]. Computers in Industry, 2023, 153: 104019.

[3] Xiong T, Liu F, Liu H, et al. Multi-drone optimal mission assignment and 3D path planning for disaster rescue[J]. Drones, 2023, 7(6): 394.

[4] Hao G, Lv Q, Huang Z, et al. Uav path planning based on improved artificial potential field method[J]. Aerospace, 2023, 10(6): 562.

[5] Xiong T, Li H, Ding K, et al. A Hybrid Improved Symbiotic Organisms Search and Sine–Cosine Particle Swarm Optimization Method for Drone 3D Path Planning [J]. Drones, 2023, 7(10): 633.

[6] Liu H. A novel path planning method for aerial UAV based on improved genetic algorithm[C]//2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS). IEEE, 2023: 1126-1130.

[7] Chen Z, Yu J, Zhao Z, et al. A path-planning method considering environmental disturbance based on VPF-RRT[J]. Drones, 2023, 7(2): 145.
[8] Huang T, Fan K, Sun W, et al. Potential-Field-RRT: A Path-Planning Algorithm for UAVs Based on Potential-Field-Oriented Greedy Strategy to Extend Random Tree [J]. Drones, 2023, 7(5): 331.
[9] Wu X, Zhao Z, Ge Q, et al. Unmanned Aerial Vehicle Path Planning Based on DP-DDPG Algorithm[C]//2023 IEEE International Conference on Unmanned Systems (ICUS). IEEE, 2023: 251-256.
[10] Xue J, Shen B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization [J]. The Journal of Supercomputing, 2023, 79(7): 7305-7336.