Soft Comput
DOI 10.1007/s00500-016-2138-6

METHODOLOGIES AND APPLICATION

Dynamic Adaptive Ant Lion Optimizer applied to route planning for unmanned aerial vehicle

Peng Yao[1,2] · **Honglun Wang**[1,2]

Abstract This paper proposes a novel Dynamic Adaptive Ant Lion Optimizer (DAALO) for route planning of unmanned aerial vehicle. Ant Lion Optimizer (ALO) is a new intelligent algorithm motivated by the phenomenon that antlions hunt ants in nature, showing the great potential to solve the optimization problems of engineering. In the proposed DAALO, the random walk of ants is replaced by Levy flight to make ALO escape from local optima more easily. Besides, by introducing the improvement rate of population as the feedback, the size of trap is adjusted dynamically based on the 1/5 Principle to improve the performance of ALO including convergence accuracy, convergence speed and stability. Compared to some other bio-inspired methods, the proposed algorithm are utilized to find the optimal route in two different environments such as mountain model and city model. The comparison results demonstrate the effectiveness, robustness and feasibility of DAALO.

Keywords Dynamic Adaptive Ant Lion Optimizer (DAALO) · Route planning · Unmanned aerial vehicle (UAV) · Ant Lion Optimizer (ALO) · Levy flight · 1/5 Principle

Communicated by V. Loia.

✉ Honglun Wang
hl_wang_2002@126.com

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

[2] Unmanned Aerial Vehicle Research Institute, Beihang University, Beijing 100191, China

1 Introduction

Depending on the advantages of cost-effectiveness and flexibility, unmanned aerial vehicle (UAV) has been widely used for some military or civilian purposes in the last decade and is trending to replace the traditional manned vehicle in some fields (Liu et al. 2013; Oz et al. 2013). To enhance the autonomous level of UAV, various technologies should be carried out, e.g., detecting and modeling the environment, mission planning, and the design of flight control systems. Route planning is one of the fundamental studies catching more and more attention (Goerzen et al. 2010). It aims to generate an optimal or sub-optimal route from the initial point to the target position under some performance criteria, e.g., path length and computational efficiency. Besides, the planned route should be under environment constraints and UAV physical constraints, i.e., the path should be collision-free to any obstacle and feasible to be executed by the UAV control system. Overall, the global route planning is a typical constrained optimization problem.

To solve the problem of route planning, researchers have proposed and developed many algorithms which can be classified into six kinds. The first group is called the roadmap-based or skeleton-based algorithm (Banerjee and Chandrasekaran 2013; Hwang et al. 2003), mainly including the Visibility Graph method and the Voronoi diagram search method. But the planned path may be unfeasible as UAV kinematic and dynamic constraints are not taken into account. Another kind of widely applicable approach is the heuristic search algorithm, where the planning space is discretized by cell decomposition and the least-cost route from the given initial node to the target node is then found by introducing the heuristic information. The Dijkstra method (Yershov and Lavalle 2011), the traditional A* algorithm (Wu and Qu 2013), the sparse A* search (SAS) method

Published online: 05 April 2016

(Szczerba et al. 2000), the D* and D* lite method (Hrabar 2008) are the typical representatives. This kind of method is simple in principle and has been used on the UAV platform successfully, but unfortunately the time spent on finding the optimal route will increase explosively if the planning space enlarges. The third group is based on principles of probabilistic programming, e.g., the probabilistic roadmaps (PRM) method (Adolf and Andert 2011) and the rapidly exploring random tree (RRT) method (Karaman et al. 2011). RRT method can avoid local minimum easily, and the planned route can fully consider the kinematic and dynamic constraints of UAV. But the obstacle avoidance behavior of the route by RRT is unsatisfactory or passive. The fourth way is the potential field-based method, including the traditional artificial potential field (APF) (Khaitib 1986; Jaradat et al. 2012), virtual force (VF) (Lam et al. 2009) and the interfered fluid dynamical system (IFDS) (Wang et al. 2015; Yao et al. 2015a, b) which we have proposed recently. This kind of method has little calculation time, but the path may trap into local minimum. The fifth group is optimization-based method (Zhang et al. 2013; Ma and Miller 2006; Ahmed and Deb 2013; Zhang et al. 2014; Zhu and Duan 2014; Karimi and Pourtakdoust 2013), as the route planning can be taken as one kind of constrained optimization problem or NP-hard problem as mentioned above. This kind of approach mainly includes mixed integer linear programming (MILP) method (Ma and Miller 2006), intelligent algorithms (Ahmed and Deb 2013; Zhang et al. 2014; Zhu and Duan 2014; Karimi and Pourtakdoust 2013), e.g., particle swarm optimization (PSO), genetic algorithm (GA), ant colony optimization (ACO), artificial bee colony (ABC) and differential evolution (DE). This group of method can deal with multiple constraints and specific applications flexibly, but the computational complexity is high. The sixth kind is the method based on optimal control, e.g., the pseudospectral method (Vera et al. 2014; Zhang et al. 2015; Bollino et al. 2007), which is utilized to solve the problem of obstacle or collision avoidance in various scenarios. This method is an intuitive method considering UAV dynamic constraints directly, but the calculation efficiency will be unsatisfactory when the environment is complicated.

Intelligent algorithm or evolutionary method, which belongs to the fifth group of method, is one kind of promising and advancing way to solve the route planning problem. The operating mechanism of intelligent algorithm imitates various natural phenomena. Compared to the deterministic algorithms, this kind of method has some stochastic operators. Hence the result in each run is different delicately, but it becomes much easier to avoid local optimum. Besides it has some intrinsic superiorities, e.g., problem independency, derivation independency and simple principle (Mirjalili 2015). Although the computational complexity is high and the local optimum entrapment still exists inevitably, this kind of method can be easily improved by introducing some mechanisms of other methods or even adopting some new intelligent algorithms. Hence intelligent algorithm is selected in this paper for UAV path planning. Fu et al. (2013) propose the phase angle-encoded and quantum-behaved particle swarm optimization (θ-QPSO), which is a variant of PSO, to plan a safe and feasible route for UAV. It is proved that the path quality, convergence speed and the stability are all improved. A memetic algorithm for global path planning (MAGPP) is presented in Zhu et al. (2015), which is the synergy of GA-based global path planning and local path refinement. By combing the traditional differential evolution algorithm and the level comparison method (mDELC), Zhang and Duan (2015) propose an improved constrained differential evolution algorithm to obtain an optimal route. Gravitational search algorithm (GSA) is firstly utilized for UAV path planning (Li and Duan 2012) based on the introduction of the nature of memory and social information of PSO, the weight-based adaption of agent mass, and the greedy strategy of EA. In addition to the above-mentioned single-objective problem, the path planning problem can be solved with the multi-objective optimization technology. In Besada-Portas et al. (2010) and de la Cruz et al. (2008), a generic multi-objective Pareto-evaluation function with goals and priorities is first utilized to achieve different types of missions, and the paths of all the UAVs are then obtained with the multiple coordinated agents coevolution EA (MCACEA), where EA is used per UAV that share their optimal solutions to coordinate the evolutions of the populations. The multi-objective optimization technology can solve the path planning problem with various contradictory optimization indexes, but the computation efficiency will decline significantly.

Recently a new bio-inspired algorithm called the Ant Lion Optimizer (ALO) is proposed by Mirjalili (2015), which mimics the behavior of antlions hunting ants in nature. Five main steps, i.e., the random walk of ants, antlions building traps, the entrapment of ants in traps, antlions catching ants and re-building traps, are implemented and formulated. By comparing it with some other popular intelligent methods, e.g., PSO, GA and cuckoo system (CS), this algorithm is benchmarked by some mathematical functions and engineering problems such as the design of ship propeller, to prove that ALO has better properties in convergence speed and accuracy, local optima avoidance and robustness (Mirjalili 2015). Instinctively we would like to utilize this new intelligent algorithm to solve the problem of 2D route planning in this paper, where the quality of optimization can improve significantly.

In this literature, some improvement strategies are introduced into ALO before it is employed for route planning, and this improved algorithm can be named the Dynamic Adaptive Ant Lion Optimizer (DAALO). Levy flight (Pavlyukevich 2007) is one kind of random movement by which the con-

tradiction between local search and global search can be reduced effectively. Hence, Levy flight (Pavlyukevich 2007) is first utilized to replace the standard random walk of ants of ALO, enabling the method explore the solution space more sufficiently and jump out of local optima with a larger probability. The second strategy is the dynamic adaption of trap size. Generally speaking, the parameters of bio-inspired methods are decisive for the process of optimization. However, most methods only determine parameters previously or adjust them online without the dynamic feedback of searching process (Li and Duan 2012; Valian et al. 2011), which is open-loop or even unstable sometimes from the perspective of control theory. Hence the improvement rate of population is taken as the feedback of ALO framework in this paper, and the trap size is then adjusted dynamically on the basis of 1/5 Principle by Rechenberg (Back 1996). Finally, to demonstrate the effectiveness of DAALO for route planning, it is compared with GA, PSO, ABC and ALO in two cases: one is the simple mountain model with circles, and the other the complicated city model with rectangles. It is concluded that DAALO generates the route with higher quality and better stability at a faster convergence speed, so it is applicable for UAV route planning.

The remainder of this paper is organized as follows. Section 2 specifies the modeling of route planning problem. In Sect. 3, the basic ALO is described in detail. In Sect. 4, the DAALO method is introduced and then utilized for UAV route planning. The comparative simulation results are given in Sect. 5. In Sect. 6, the conclusion and future work are provided.

2 Description of UAV route planning problem

2.1 Modeling of route

A simplified model (Zhu and Duan 2014) is utilized to represent the planned route in this paper. The start point and target point in the planning space o_xy are determined to be $P_s = (x_s, y_s)$ and $P_t = (x_t, y_t)$, respectively, as shown in Fig. 1. Firstly P_s and P_t are connected with a straight line, and P_sP_t is then divided into $D+1$ parts equally. At each segment point the line perpendicular to P_sP_t is drawn, and this set of lines can be denoted as $L_1 \cdots L_d \cdots L_D$. Then we choose a point from each line, composing the set of discrete points $C = \{P_s, P_1 \cdots P_d \cdots P_D, P_t\}$ where $P_d = (x_d, y_d)$. And the route is then obtained by connecting these points successively. Subsequently, the route planning problem can be taken as the optimization of coordinate series.

To further simplify this model, the initial coordinate system o_xy is transformed into the new one o′_$x'y'$, where P_s is taken as the origin and P_sP_t the x' axis. Then the set C in o′_$x'y'$ are redefined as $C' = \{P'_s, P'_1 \cdots P'_d \cdots P'_D, P'_t\}$,

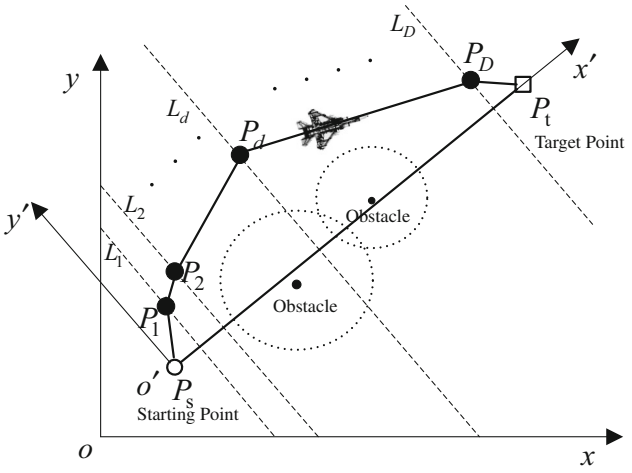


Fig. 1 The model of planned route

where $P'_s = (0, 0)$, $P'_1 = (x'_1, y'_1)$, $P'_d = (x'_d, y'_d)$, $P'_D = (x'_D, y'_D)$ and $P'_t = (|P_sP_t|, 0)$. In this way, the x-coordinate of any point P'_d can be obtained by the fixed expression $x'_d = \frac{d}{D+1} \cdot |P_sP_t|$. Hence only y-coordinates, i.e., $\{y'_1 \cdots y'_d \cdots y'_D\}$ need to be optimized, reducing the amount of calculation significantly.

2.2 Constraints of route planning

2.2.1 Environment constraint

To simplify the algorithm, the obstacles, e.g., mountains, buildings or radars in the flight environment are equivalent to some standard circles or rectangles. Hence a unified formulation can represent the obstacle (Wang et al. 2015):

$$\Gamma(P) = \left(\frac{x - x_0}{a}\right)^{2c} + \left(\frac{y - y_0}{b}\right)^{2d}, \tag{1}$$

where (x_0, y_0) is the obstacle center, a, b the axis lengths, and c, d the index parameters. a, b, c, d can determine the size and shape of obstacle: if $c = d = 1$ and $a = b$, the obstacle is a circle; if $c = d = 1$ and $a \neq b$, the object is a ellipse; if $c \gg 1, d \gg 1$, the obstacle can be regarded as a rectangle. The inequation $\Gamma(P) > 1$ expresses the region outside of the obstacle, and $\Gamma(P) < 1$ denotes the space inside of the obstacle, and $\Gamma(P) = 1$ is the surface of obstacle.

To satisfy the environment constraint, the route must avoid all the obstacles, i.e., the route should be in the safety region S_{free}:

$$S_{\text{free}} = \{P | \Gamma_k(P) > 1, \forall k = 1, ..., K\}, \tag{2}$$

where K is the number of obstacles in the planning space and $\Gamma_k(P)$ the formulation of the kth obstacle using Eq.(1).

Besides, the route should be inside of the planning region, and the definition of upper bound and lower bound can be found in Zhang and Duan (2015).

2.2.2 UAV physical constraint

UAV performance must be considered as another kind of constraint to ensure the feasibility of route. In this paper, the maximum turn rate of UAV $\omega_{\max}$ and the maximum length of route $l_{\max}$ are mainly considered. Suppose ω_{d} the UAV turn rate of the dth route segment and l the whole length of route. Then the following conditions should be satisfied:

$$\begin{cases} |\omega_{\mathrm{d}}| \le \omega_{\max}, & \forall d = 1, 2, \ldots, D \\ l \le l_{\max} \end{cases} \tag{3}$$

2.3 The fitness evaluation function of route optimization

The route cost J is regarded as the fitness function of optimization and defined as follows:

$$J = \lambda_1 J_1 + \lambda_2 J_2, \tag{4}$$

where J_1 is the length cost, J_2 the safety cost, $\lambda_1, \lambda_2 \in [0, 1]$ the weighs of J_1 and J_2, respectively, meeting $\lambda_1 + \lambda_2 = 1$. Hence when λ_1 is approximating closer to 1, a shorter route will be obtained with less consideration to obstacle avoidance. On the other hand, when the value of λ_1 is closer to 0, a safer route farther away from obstacles can be generated at the expense of path length.

J_1 is simply defined as follows:

$$J_1 = |P_1 - P_{\mathrm{s}}| + |P_{\mathrm{t}} - P_{\mathrm{D}}| + \sum_{d=2}^{D} |P_{\mathrm{d}} - P_{\mathrm{d}-1}|. \tag{5}$$

Define P_0 and $P_{\mathrm{D}+1}$ as the start point P_{s} and the target point P_{t}, respectively. To describe the whole safety cost J_2 more accurately, each path segment is divided into five portions equally and each segmentation point contributes to the calculation. Hence the safety cost of path segment $P_{\mathrm{d}-1}P_{\mathrm{d}}$ to the k th obstacle can be given:

$$T_k(d) = \frac{\mu}{6} \cdot \left(\frac{1}{|P_{\mathrm{d}-1} - O_k|} + \frac{1}{|P_{\mathrm{d}} - O_k|} + \sum_{i=1}^{4} \frac{1}{|P_{d,i} - O_k|} \right), \tag{6}$$

where O_k is the center of the kth obstacle, $P_{\mathrm{d,i}}$ the ith segmentation point, and μ the safety coefficient depending on the flight environment.

Fig. 2 Funnel-shaped sandpit by antlions

Considering all the path segments and all the obstacles, we can obtain the safety cost J_2:

$$J_2 = \frac{1}{K} \cdot \sum_{k=1}^{K} \sum_{d=1}^{D+1} T_k(d). \tag{7}$$

3 Introduction to the basic ALO

3.1 Inspiration from the phenomenon

Antlions are the larvae of myrmeleontinae in nature, whose name comes from the unique hunting behavior. The process is simplified as follows. Firstly, an antlion digs a funnel-shaped sandpit by rotating and drilling down with its strong jaw. The built trap is shown in Fig. 2. Then the antlion hides on the bottom of pit and waits for insects especially ants to be trapped. Once an ant falls into the trap unfortunately, it will be hard for the ant to escape, as the antlion throws sands out of pit with its mandible to force the victim to slide towards the bottom. Finally the ant will be dragged into the sand and eaten. After that, the antlion amends the trap and waits for the next hunt. Besides, it must be noticed that the size of pit is positively related to the strength of antlions. Usually the more ants the antlion has consumed, the stronger the antlion, the bigger the pit size, the larger the probability of catching more ants.

3.2 Mathematical model of ALO

From the above analysis, two kinds of species, i.e., antlions and ants should be modeled at the same time. Ants move in the searching space, and antlions hunt them with fitter traps. Suppose N ants combine the ant population $P_{\mathrm{ant}} = (P_{A,1}, \ldots, P_{A,n}, \ldots, P_{A,N})^{\mathrm{T}}$ in the D-dimensional searching space, and $P_{A,n} = (P_{A,n}^1, \ldots, P_{A,n}^d, \ldots, P_{A,n}^D)$ is the position

of the nth ant, where $P^d_{A,n}$ means the position of the dth variable of the nth ant. The fitness values of all ants can then be stored in the matrix $F_{\text{ant}} = (F_{A,1}, ..., F_{A,n}, ..., F_{A,N})^{\text{T}}$, where $F_{A,n} = f(P^1_{A,n}, ..., P^d_{A,n}, ..., P^D_{A,n})$ is the objective function value of nth ant. Similarly, N antlions combine the antlion population $P_{\text{antlion}}=(P_{AL,1}, ..., P_{AL,n}, ..., P_{AL,N})^{\text{T}}$, and $P_{AL,n} = (P^1_{AL,n}, ..., P^d_{AL,n}, ..., P^D_{AL,n})$ is the position of the nth antlion where $P^d_{AL,n}$ means the position of the dth variable of the nth antlion. The fitness values of all antlions will be stored in the matrix $F_{\text{antlion}} = (F_{AL,1}, ..., F_{AL,n}, ..., F_{AL,N})^{\text{T}}$, where $F_{AL,n} = f(P^1_{AL,n}, ..., P^d_{AL,n}, ..., P^D_{AL,n})$ is the objective function value of the nth antlion.

3.2.1 The random walks of ants

In this paper, the random walk is selected and utilized to all the dimensions of each ant. For the nth ant, we define $P^d_{A,n}(t)$ as the dth variable at the tth iteration, and the set of walks in the whole process can be modeled as follows:

$$\begin{aligned} C(P^d_{A,n}) = [&\text{cumsum}\,(2r(t_1) - 1)\,, \\ &\text{cumsum}\,(2r(t_2) - 1)\,, \ldots, \\ &\text{cumsum}\,(2r(t_{\max}) - 1)]\,, \end{aligned} \tag{8}$$

where cumsum denotes the cumulative sum, $t_{\max}$ is the maximum iteration, and $r(t)$ is defined as follows:

$$r(t) = \begin{cases} 0 & \text{if rand} > 0.5 \\ 1 & \text{otherwise} \end{cases}, \tag{9}$$

where rand is a random number between 0 to 1. By this description, ants would have three tendencies of motion, i.e., fluctuating around the initial position, the increasing behavior and the decreasing trend.

However, the walks by Eq. (8) cannot be utilized directly as it may be out of boundary of variable, so the following normalization is adopted:

$$\begin{aligned} P^d_{A,n}(t) = &\frac{\left(P^d_{A,n}(t) - \min C(P^d_{A,n})\right) \cdot \left(u^d(t) - l^d(t)\right)}{\max C(P^d_{A,n}) - \min C(P^d_{A,n})} \\ &+ l^d(t), \end{aligned} \tag{10}$$

where $\min C(P^d_{A,n})$ and $\max C(P^d_{A,n})$ represent the minimum and maximum of random walks for the dth variable of the nth ant, $l^d(t)$ and $u^d(t)$ the lower bound and upper bound of dth variable at tth iteration, respectively (the boundary of searching space for each ant at the current iteration is the same in this paper). The random walks of ants are around antlions, which will be discussed in the following.

3.2.2 Antlions building traps

As mentioned above, the stronger the antlion is (i.e., the smaller the fitness function is), the higher the probability of hunting ants. Therefore the principle of roulette wheel is adopted to select the antlion P_{sel} based on the fitness values of the antlion population. Besides, the elite antlion P_{elite}, i.e., the best solution obtained so far is saved, which will influence the motions of all the ants during iterations. Hence the traps by P_{sel} and P_{elite} should be considered at the same time, and the definition of trap size (i.e., the lower and upper bound) will be discussed in Sect. 3.2.3.

Hence the ant will walk randomly as follows:

$$P^d_{A,n}(t) = \frac{R_{\text{A}}(t) + R_{\text{E}}(t)}{2}, \tag{11}$$

where $R_{\text{A}}(t)$ is the random walk around the selected antlion P_{sel}, and $R_{\text{E}}(t)$ the random walk around the elite antlion P_{elite} by Eqs. (8)–(10).

3.2.3 The entrapment of ants

As mentioned above, the ant will slide towards the antlion when it is trapped in the pit. Hence the lower and upper bound should decrease with the increase of iteration:

$$u^d(t) = \frac{u^d(t)}{I}, \tag{12}$$

$$l^d(t) = \frac{l^d(t)}{I}, \tag{13}$$

where I is a ratio defined as $I = 10^w \cdot t/t_{\max}$, and w is a constant on the basis of the current iteration ($w = 1$ when $t \le 0.1t_{\max}$, $w = 2$ when $0.1t_{\max} < t \le 0.5t_{\max}$, $w = 3$ when $0.5t_{\max} < t \le 0.75t_{\max}$, $w = 4$ when $0.75t_{\max} < t \le 0.9t_{\max}$, $w = 5$ when $0.9t_{\max} < t \le 0.95t_{\max}$, $w = 6$ when $t > 0.9t_{\max}$).

The random walks of ants are affected by the traps of antlions, hence the lower and upper bound are then modeled by the following equation:

$$u^d(t) = \begin{cases} P^d_{AL}(t) + u^d(t) & \text{if rand} > 0.5 \\ P^d_{AL}(t) - u^d(t) & \text{otherwise} \end{cases}, \tag{14}$$

$$l^d(t) = \begin{cases} P^d_{AL}(t) + l^d(t) & \text{if rand} > 0.5 \\ P^d_{AL}(t) - l^d(t) & \text{otherwise} \end{cases}, \tag{15}$$

where P^d_{AL} is the dth variable of the antlion P_{sel} or P_{elite}. From the analysis of Eqs. (14) and (15), the ant will walk randomly in a hypersphere around antlion.

Table 1 Pseudo-codes of ALO

The process of ALO algorithm
1. Initialize the first population of ants and antlions by the random sampling
2. Calculate the fitness values of all ants and antlions
3. Determine the antlion with the minimum fitness value as the elite antlion P_{elite}
4. **while** ($t \le t_{\max}$)
5. **for** each ant (i.e. $n = 1 \cdots N$)
6. Determine the selected antlion P_{sel} based on the fitness values by the roulette wheel principle
7. **for** each dimension (i.e. $d = 1 \cdots D$)
8. Update the lower and upper bounds by Eq.(12-13)
9. Calculate the bounds around the selected antlion or the elite antlion respectively by Eq.(14-15)
10. Determine $R_A(t)$ or $R_E(t)$ i.e. the random walk around the selected antlion or the elite antlion respectively by Eq.(8-10)
11. Update the position of the ant using Eq.(11)
12. **end for**
13. **end for**
14. Calculate the fitness values of all ants
15. Fitness values of ants and antlions are concatenated and then sorted from the smallest to the largest
16. Based on Eq.(16), choose the first half part as the antlion population and the first one as the elite antlion P_{elite}
17. The next iteration i.e. t=t+1
18. **end while**

3.2.4 Catching ants and re-building traps

The behavior of catching prey occurs when the fitness value of ant is less than that of antlion. Then the antlion updates its position to that of the hunted ant, as shown by the following equation:

$$P_{AL,j}(t) = P_{A,i}(t) \text{ if } (P_{A,i}(t)) < f(P_{AL,j}(t)), \qquad (16)$$

where $P_{AL,j}(t)$ is the position of jth antlion at tth iteration, $P_{A,i}(t)$ the position of ith ant at tth iteration. To achieve this process, all the fitness values of ants and antlions, i.e., F_{ant} and F_{antlion} can be concatenated and then sorted from the smallest to the largest. Then we determine the first N rows as the updated F_{antlion} and the corresponding positions as P_{antlion}.

3.3 ALO algorithm

The pseudo-codes of ALO are shown in Table 1. First, the positions of antlions and ants are initialized randomly. Second, the positions of all the ants in the current iteration are calculated. The bounds of ant position are shrinking with the increase of iteration; the antlion P_{sel} is selected by the roulette wheel principle and P_{elite} is determined by the elitism; then the ant positions are updated on the basis of the random walks around P_{sel} and P_{elite}, respectively. Third, the new positions of all ants are evaluated by the fitness function and compared with those of antlions. If there are ants fitter than one or more antlion, their positions are taken as the new positions of antlions for the next iteration. Besides, the elite antlion will be updated if the best antlion obtained in this iteration becomes fitter than the elite. The above steps repeat until the end of iteration.

4 Description of the DAALO

4.1 The improvement strategies

The basic ALO algorithm has good performance of exploration and exploitation. The exploration, i.e., the global search is guaranteed by the random selection of antlions to some extent and the random walks of ants around antlions, so the probability of avoiding local optima is high. The exploitation, i.e., the local search is ensured by the shrinking size of antlions' traps and the promising search area saved by the elite antlion, so the convergence accuracy is good. However, the performance of this method may still be unsatisfactory, especially for some complicated engineering problems, e.g., route planning. In order to further improve the performances of ALO, including the adaptability to different problems, the robustness of optimization results, the convergence accuracy and speed, the DAALO is proposed by introducing some improvement strategies in the following section.

4.1.1 Levy flight

Levy flight, intermittent walk and correlative random walk are the widely used models describing the biological motion in nature (Viswanathan et al. 2000). Levy flight is a Markov process, whose step-lengths have a heavy-tailed probability distribution. The other two kinds of walks belong to Brown walks, so does the walk by Eq. (8) in Sect. 3.2.1. The relation between the variance of Levy flight and the current iteration is $\sigma^2(t) \sim t^{3-\lambda}$, $1 < \lambda < 3$, while there is linear relationship between the variance of Brown walks and the current iteration, i.e., $\sigma^2(t) \sim t$. Therefore, the efficiency of exploring the whole space by Levy flight is higher than that by Brown walks.

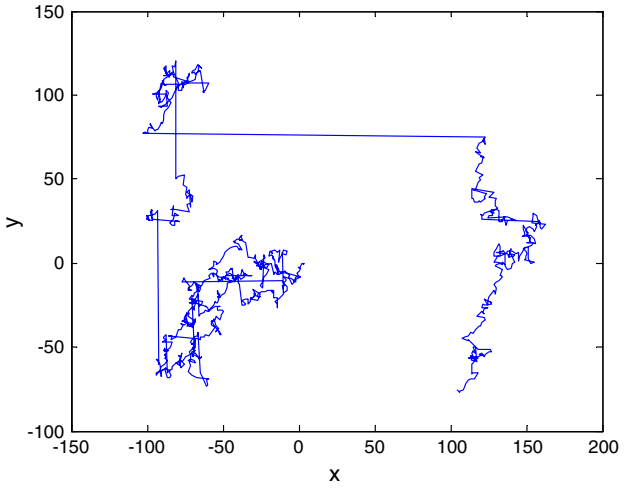


Fig. 3 The 2D Levy flight trajectory

Figure 3 shows the trajectory with 1000 steps by Levy flight in 2D plane. On the face of it, there will be one abrupt long-distance flight after multiple movements which look like Brown Walks. That is to say, both the usual short-range fight and occasional long-distance flight exist in the whole Levy flights. By utilizing Levy flight in the process of intelligent optimization, the contradiction between local search and global search can be resolved more effectively: on the one hand, some solutions search around the current optima to accelerate the process of local search; on the other hand, some solutions search the area which is far from the current optima to jump out of the local minimum when necessary. Therefore, the Levy flight is utilized to replace the random walk as the motion of ants. Hence Eq. (8) can be redefined as follows:

$$C(P^d_{A,n}) = [\text{cumsum}(L(t_1)), \text{cumsum}(L(t_2)), \ldots, \text{cumsum}(L(t_{\max}))], \quad (17)$$

where $L(t)$ is the Levy flight at the tth iteration, and the coefficient value of Levy flight can be found in Yang (2011).

4.1.2 The dynamic adaption of trap size

From Eqs. (12)–(15), the size of trap (or the ratio I) is the main parameter to be adjusted. Although the trap size is shrinking (i.e., I is increasing) with the increase of iteration, this mechanism is still passive or unsatisfactory, as the value of parameter is too dependent on intuition or trial-and-error and cannot react dynamically to the performance of the current generation. To react to the change of population more actively, it is necessary to introduce some strategy to control the evolution of population. Hence the dynamic adaption to the ratio I is proposed in this paper, which is similar to the closed-loop system of the control theory: the current population performance is the feedback, the expected population performance the reference, the adaption strategy the control algorithm.

In this paper, we define the improvement rate of population $R = \bar{N}_{AL}/N$ as the population performance, where $\bar{N}_{AL}$ is the number of antlions which are better than their last generation. The 1/5 Principle, proposed by Rechenberg (Back 1996) for the parameter self-tuning of evolution computation, is taken as the adaption strategy. The main theme of 1/5 Principle is that the parameters of algorithm should change dynamically to ensure the improvement rate of population to remain about 20 %. In this paper, the adjusted ratio I at current iteration is expressed by $I(t) = I_o(t) \cdot f(t)$, where $f(t)$ is the modulation coefficient with the initial value $f(t) = 1$ at the first iteration, and $I_o(t)$ is defined as $I_o(t) = 1000 \cdot t/t_{\max}$. Then $f(t)$ is adjusted as follows:

(1) When the improvement rate of population is less than 0.2, the search space is large enough to avoid local minima but the searching precision is low. To improve the exploitation ability, the search area, i.e., the trap size should further shrink by increasing $f(t)$.
(2) When the improvement rate of population is larger than 0.2, it is concluded that the solutions mainly search in the local area. To improve the exploration ability, the search area, i.e., the trap size should expand in turn by decreasing $f(t)$.
(3) When the improvement rate of population is equal to 0.2, both the exploration and exploitation are considered reasonably, so no adaption is needed.

However, the third case rarely appears in the actual optimization process, so $f(t)$ will change too frequently within a large scope. To enhance the stability of parameter self-tuning, the judging criterion is expanded from the single value 0.2 to the interval [0.2, 0.3]. Hence the adjustment of $f(t)$ can be redefined as follows:

$$f(t) = \begin{cases} f(t-1) \cdot f_I & R < 0.2 \\ f(t-1) & 0.2 \le R \le 0.3 \\ f(t-1)/f_I & R > 0.3 \end{cases}, \quad (18)$$

where R is the improvement rate of last generation, $f(t-1)$ is the modulation coefficient of last generation, and f_I is a constant value larger than 1, which is called the learning factor.

4.2 DAALO for route planning

In the process of DAALO, the random walk of ants are replaced by Levy flights, and the trap size is adapted dynamically based on the 1/5 Principle. Compared to the basic ALO, the searching region by DAALO switches more reasonably

between the global and local area. Therefore, there is a higher probability of resolving local optima, and the convergence speed and accuracy can improve as well. Then the proposed DAALO algorithm can be utilized for UAV route planning, and the procedures are as follows:

Step 1 Based on the environment information (e.g., the planning space, obstacle center, obstacle size and obstacle type) and UAV performances, determine the environment constraints and UAV dynamical constraints.

Step 2 Initialize the parameters of DAALO, including the number of ants or antlions N, the dimension of positions of ant or antlion D, the maximum iteration $t_{\max}$, the current iteration $t = 1$, the initial upper bound $u^d(t), d = 1 \cdots D$, the lower bound $l^d(t), d = 1 \cdots D$, and the learning factor f_I. As the waypoints are evenly spaced on x axis in the transformed coordinate system, only y coordinates are chosen randomly within the bounds to form the initial position of each ant or antlion $\{y'_1 \cdots y'_d \cdots y'_D\}$. Then, calculate the cost values of routes generated by the initial antlions based on Eqs. (3)–(7), and choose the selected antlion P_{sel} by the roulette wheel principle and the elite antlion P_{elite}.

Step 3 Adapt the ratio I dynamically and then update the bounds around P_{sel} and P_{elite}, respectively, by Eqs. (12)–(15) successively.

Step 4 Determine $R_A(t)$ or $R_E(t)$, i.e., the Levy flight of ants around P_{sel} or P_{elite}, respectively, by Eqs. (1) and (10).

Step 5 Obtain all the ants' position using Eq. (11) and calculate the cost values of the corresponding routes by Eqs. (3)–(7).

Step 6 Update all the antlions' position based on Eq. (16), and choose the selected antlion P_{sel} and the elite antlion P_{elite} for the next iteration ($t = t + 1$).

Step 7 If the termination condition holds, i.e., $t > t_{\max}$, then output the best solution and the corresponding route. Otherwise, go to Step 3.

The detailed process is shown in Fig. 4.

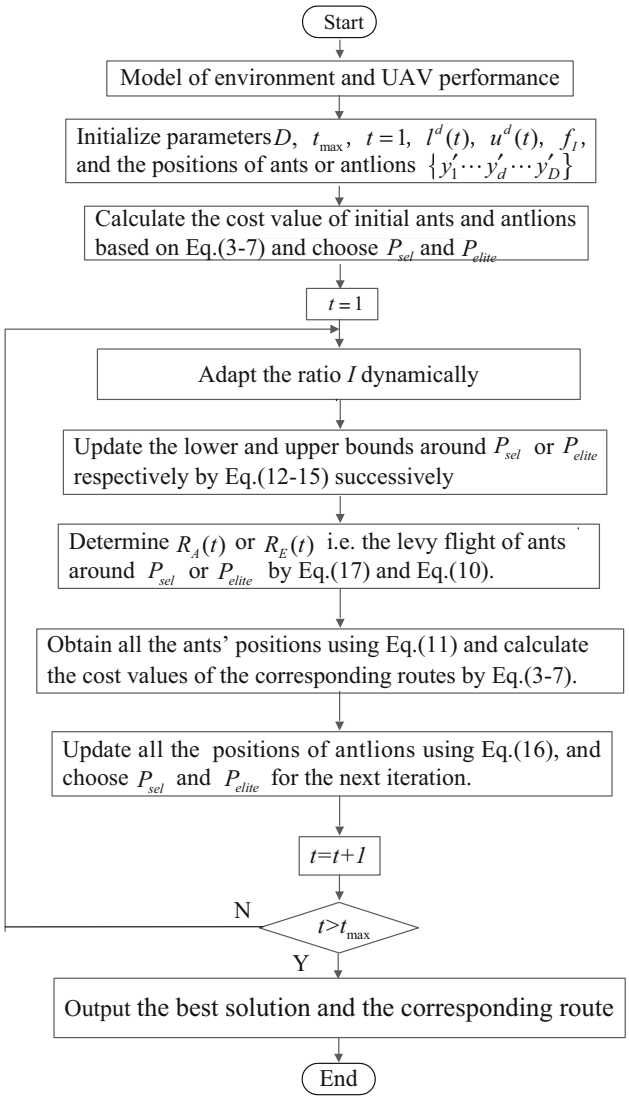


Fig. 4 The procedure of our proposed method

5 Simulation experiments

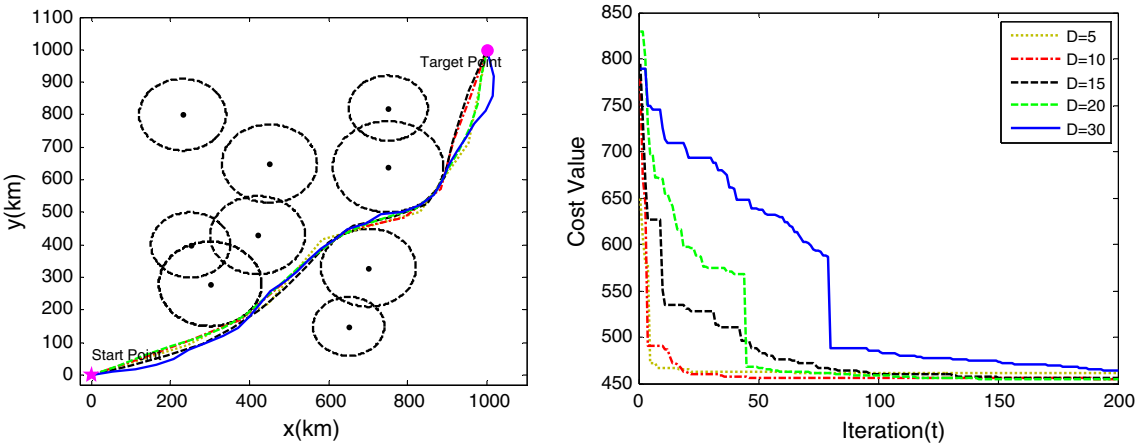
To evaluate the performances of DAALO to solve the route planning problem, series of experiments are implemented in MATLAB R2011a on a PC with the dominant frequency of 2.5 GHz running Windows 7. Each group of experiments should be executed 50 times independently and the statistical results are utilized for assessment analysis.

Unless otherwise specified, the parameters of DAALO are as follows: the population of ants or antlions $N = 30$, the dimension of positions of ant or antlion $D = 15$, the maximum iteration $t_{\max} = 200$, the learning factor $f_I = 1.2$. The physical performances of UAV are as follows: $\omega_{\max} = 30°$, $l_{\max} = 2000$ km. The safety coefficient is $\mu = 2000$ and the weigh coefficient of cost value is $\lambda_1 = 0.3$ and $\lambda_2 = 0.7$. The start point and destination are (0, 0) km and (1000, 1000) km, respectively. Two cases are tested here: case I is the simple mountain environment with nine circles, and case II the complicated city model with 13 rectangles as obstacles, as shown in Figs. 5a and 6a.

5.1 Comparison of D

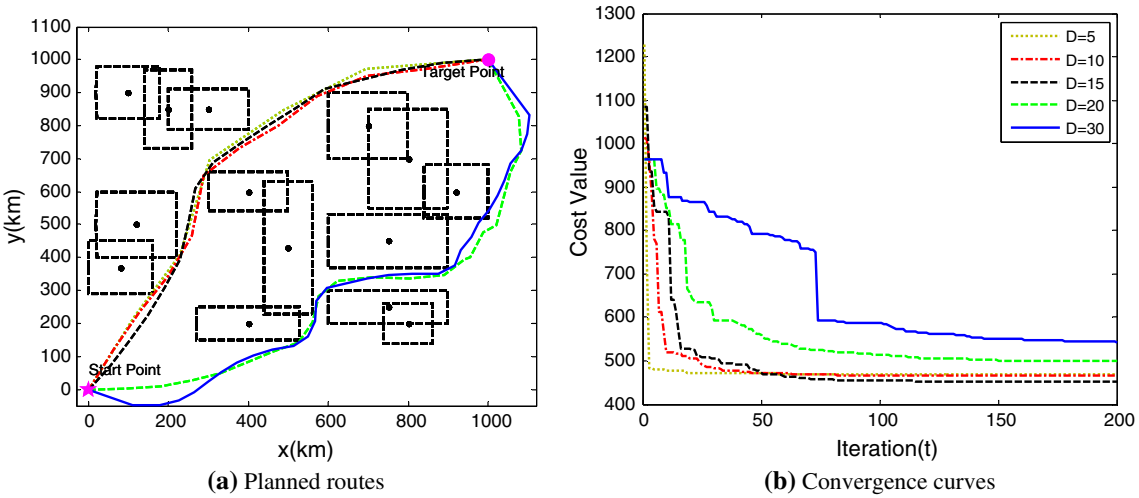
In this paper, D is the dimension of ant or antlion position, i.e., the number of discrete waypoints. The value of D is usually defined by seeking the compromise between the environment information, the computational efficiency and the route accuracy. To understand the influence of D on the routes, the DAALO method with different D in the above two cases is tested and analyzed by 50 times.

Figures 5a and 6a show the typical routes with different D in case I and case II during 50 runs. All the five routes



(a) Planned routes **(b)** Convergence curves

Fig. 5 Comparisons of results with different parameter D in case I



(a) Planned routes **(b)** Convergence curves

Fig. 6 Comparisons of results with different parameter D in case II

can avoid obstacles safely, but the path with $D = 15$ is of the best quality with short length and excellent smoothness. The convergence curves of the best fitness values are shown in Figs. 5b and 6b. Usually the smaller D is, the faster the convergence speed. But if D is too small, the optimal value after 200 iterations will be large. On the other hand, if D is too large, both the convergence speed and accuracy will be unsatisfactory.

The statistical results are listed in Table 2, including the best, mean, median, worst, standard deviation of the fitness value, as well as the percentage of feasible routes (i.e., FR) which satisfy the environment constraints and the UAV physical constraints at the same time. And the best results are marked with boldface. It should be emphasized that only the feasible routes, i.e., those successful runs are utilized for the statistics of column 3–7. In case I, the smaller the dimension D is, the less the standard deviation, but the feasible rate (i.e., FR) is the highest when $D = 15$. Besides, the smallest of the best, mean, median and worst cost value are obtained when $D = 20$, $D = 15$, $D = 15$ and $D = 10$, respectively. In case II, when $D = 15$, the highest FR and the smallest standard deviation are determined. Besides, the smallest of the best, mean, median and worst cost value are obtained when $D = 10$, $D = 15$, $D = 10$ and $D = 15$, respectively. When D is too large, the path planning problem will become very complex and sometimes it is hard for DAALO to converge to the optimal solution, in spite of the more accurate representation of route. If D is too small, the route cannot be described objectively or accurately. Overall, $D = 15$ is the most appropriate value, and the corresponding route is of the best quality.

5.2 Comparison between random walk and Levy flight

In our proposed DAALO method, the movements of ants are determined by the Levy flight instead of the random walk

Table 2 Comparisons of different parameter D

D	Best	Mean	Median	Worst	Standard deviation	FR (%)
Case I						
5	461.28	461.44	461.43	461.85	**0.13**	90
10	455.17	455.56	455.45	**456.76**	0.35	94
15	454.17	**455.10**	**454.93**	457.25	0.71	**100**
20	**453.70**	456.34	455.86	461.32	2.08	94
30	457.55	468.84	467.82	498.14	8.76	92
Case II						
5	462.21	470.24	462.28	572.05	22.71	78
10	**461.63**	467.43	**462.10**	509.43	11.79	82
15	462.29	**466.29**	464.59	**507.25**	**8.24**	**94**
20	462.32	476.17	469.67	517.85	15.28	84
30	463.31	502.46	493.05	573.78	28.54	56

Bold values indicate the best or optimal result of each column

utilized in the basic ALO. To verify the effectiveness of Levy flight in improving the performance of the proposed method, the above two kinds of movements are separately utilized in DAALO for case I and case II by 50 runs.

The statistical results are listed in Table 3. As mentioned in Sect. 4.1.1, by utilizing Levy flight, the contradiction between local search and global search is resolved more effectively and the solutions can jump out of the local minimum more easily. Therefore, for both case I and case II, the feasible rate (i.e., FR) by Levy flight increases significantly compared to that by random walk. In case I, the other statistical results by Levy flight is slightly worse than those by random walk. In case II, the mean, worst and standard deviation of cost value by Levy flight are much better than those by random walk.

In addition, if random walk is utilized, the fitness value of DAALO will converge rapidly to the optimal at the 85th iteration in case I and at the 60th iteration in case II. If Levy flight is utilized, the converging process will occur during the first 80 iterations in case I and the first 57 steps in case II, respectively. Hence the motion mode of ants has less influence on the convergence speed.

All things considered, the Levy flight is a better choice to represent the motion of ants.

5.3 Comparison of different learning factors

In our proposed DAALO method, the strategy of dynamic adaption of trap size is utilized. And the selection of learning factor f_I is crucial: if the value of f_I is appropriate, the performance of the algorithm will increase dramatically; if f_I is unreasonable, the planning results will be counterproductive. Therefore the comparisons between different learning factors $f_I = 0.8, 1.0, 1.2, 1.5, 1.8, 2.5$ are tested in two cases by 50 times. When the value of f_I is 1.0, it means that the strategy of dynamic adaption is not adopted.

The statistical results are shown in Table 4. It is evident that, nearly all the indicators of cost value are the best for both two cases when $f_I = 1.2$ (the median of cost value when $f_I = 1.2$ is slightly larger than that of $f_I = 1.5$ in case II). Hence $f_I = 1.2$ is the recommended value. Besides, the performance of DAALO with a too large or too small learning factor is even worse than the situation that no adaption strategy is utilized, i.e., $f_I = 1.0$. The convergence curves in case I and II with different learning factors are shown in Fig. 7a, b, respectively. In either case, by choosing an appropriate value $f_I = 1.2$, both the convergence speed and accuracy will be the best.

Figure 8a illustrates the varying ratio by $f_I = 1.0$ and $f_I = 1.2$ in case I. When $f_I = 1.0$, the ratio I will increase linearly with the iteration; when $f_I = 1.2$, the ratio will increase adaptively. Figure 8b illustrates the corresponding shrinking trap size, where both the lower and upper bounds are displayed. The adaptive shrinking size can further improve the convergence accuracy and convergence speed, and meanwhile it can still guarantee the exploitation, i.e., local search.

5.4 Comparison of different methods

In the two cases, the basic ALO method and three representative population-based optimization algorithms, i.e., GA, PSO and ABC, are also utilized for route planning. To guarantee a fair comparison, all the parameters of basic ALO are the same as those of ADDLO, and it should be noticed that the population size of ants and that of antlions are the same, which are equal to N. For GA, PSO and ABC, the maximum iteration $t_{\max}$ and the dimension of position D are the same as DAALO. For GA, the population size is $N = 30$; binary coding is utilized for each position and the chromosome length is 5; the principle of roulette wheel is used for

Table 3 Statistical results of two movements of ants

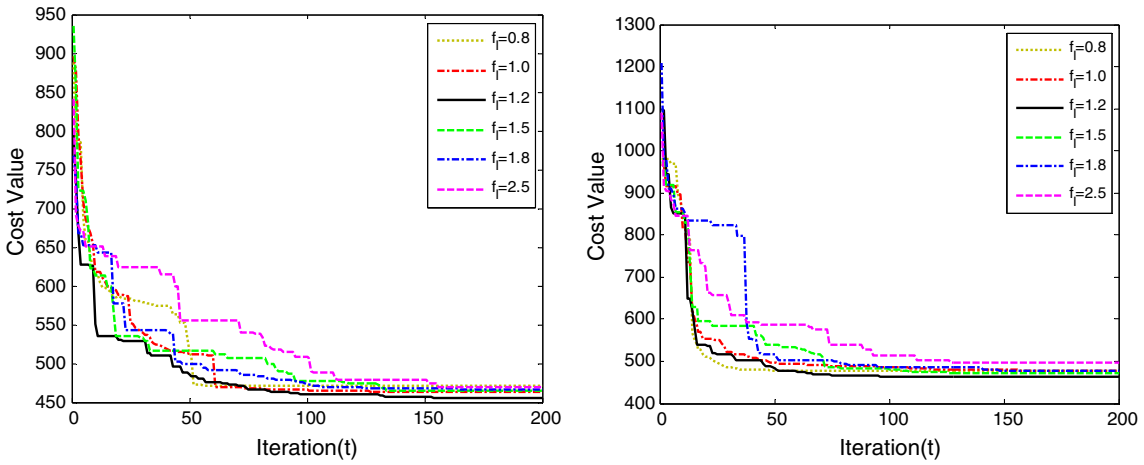
	Best	Mean	Median	Worst	Standard deviation	FR (%)
Case I						
Random walk	**454.14**	**454.62**	**454.50**	**456.65**	**0.55**	94
Levy flight	454.17	455.10	454.93	457.25	0.71	**100**
Case II						
Random walk	**462.20**	472.96	**462.96**	583.21	24.97	78
Levy flight	462.29	**466.29**	464.59	**507.25**	**8.24**	**94**

Bold values indicate the best or optimal result of each column

Table 4 Statistical results of different learning factor

f_I	Best	Mean	Median	Worst	Standard deviation	FR (%)
Case I						
0.8	454.85	461.67	461.72	470.00	4.54	92
1.0	454.26	455.78	455.34	461.27	1.50	94
1.2	**454.17**	**455.10**	**454.93**	**457.25**	**0.71**	**100**
1.5	454.21	455.58	455.01	461.16	1.54	98
1.8	454.46	458.46	455.78	481.28	6.50	96
2.5	464.54	465.11	461.48	492.84	10.46	96
Case II						
0.8	463.48	482.57	480.09	528.85	14.51	70
1.0	462.39	470.10	465.65	515.13	12.78	84
1.2	**462.29**	**466.29**	464.59	**507.25**	**8.24**	**94**
1.5	462.30	470.29	**464.31**	515.60	14.92	82
2.5	462.79	490.47	481.43	590.57	30.20	88

Bold values indicate the best or optimal result of each column

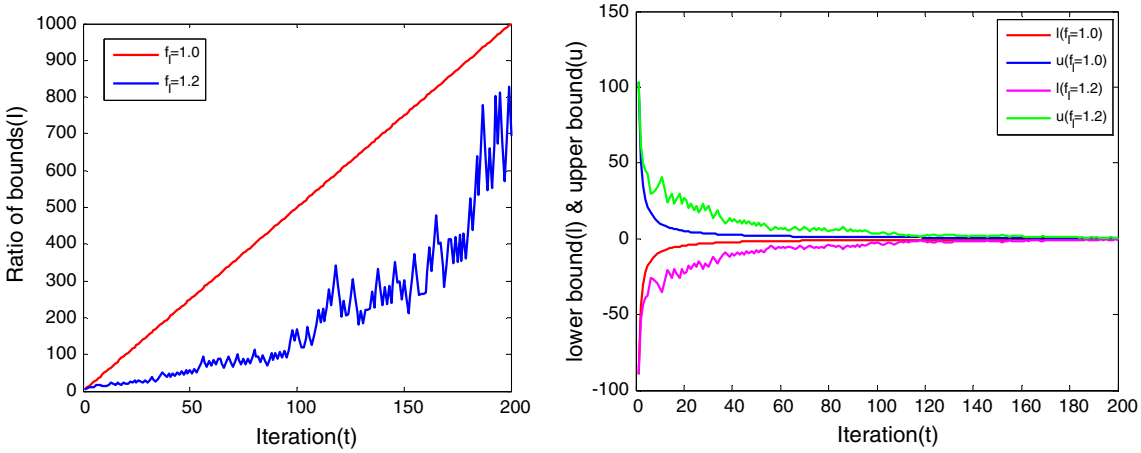


(a) Case I **(b)** Case II

Fig. 7 Convergence curves with different learning factors

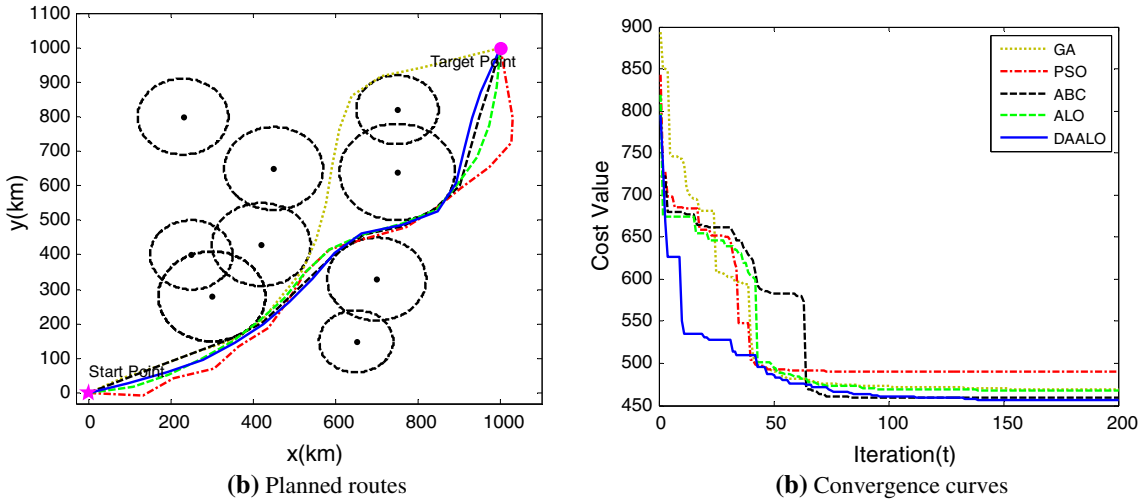
selection; the one-point crossover is adopted with the probability 0.7; the mutation probability is 0.05. For PSO, the population size is $N = 30$, and the learning factors are $c1 = c1 = 2.0$, and the weigh coefficient is defined by $w = w_{\max} - (w_{\max} - w_{\min})t/t_{\max}$ where $w_{\max} = 0.9$, $w_{\min} = 0.1$. For ABC, the numbers of employed or unemployed bees are equal to N, i.e., $N_e = N_u = N = 30$, so the population size is $2 \cdot N$, and the threshold limit of search times is Limit $= 40$.

Figure 9a shows the typical routes in case I by the five methods during 50 runs. All the five routes can avoid obstacles safely, but the path by DAALO is of the best quality with short length and excellent smoothness. The convergence curves of the best fitness values by various methods are shown



(a) Ratio *I* **(b)** Lower and upper bounds

Fig. 8 Change curves of adaptive trap size in case I



(b) Planned routes **(b)** Convergence curves

Fig. 9 Comparisons of results by various methods in case I

in Fig. 9b. The convergence speed of the basic ALO is faster than that of ABC, while the convergence accuracy of ALO is worse than that of ABC. And the comparison between ALO and PSO or GA is contrary to the above result. Compared to other methods, the cost value by DAALO drops to the smallest value with the fastest convergence speed in the first half of iterations.

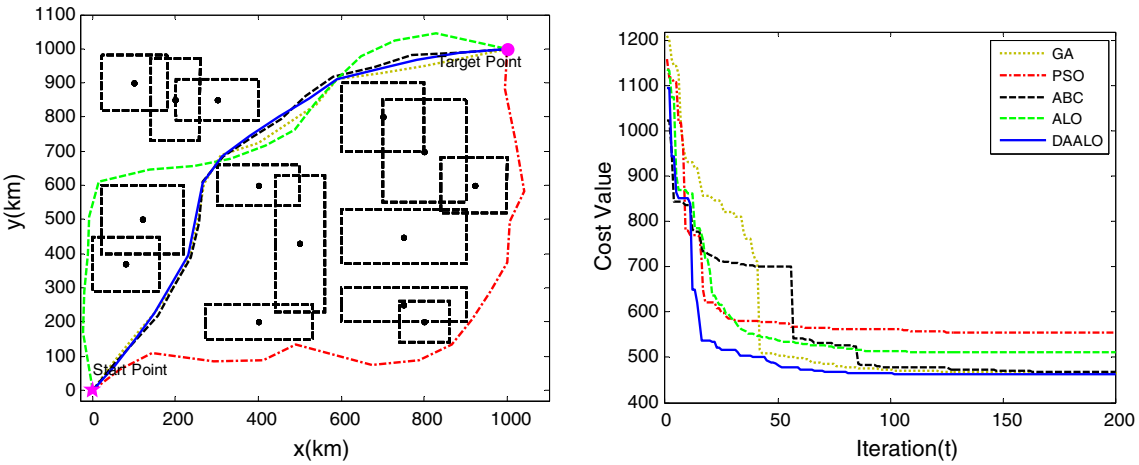
The statistical results of GA, PSO, ABC ALO and DAALO in case I during 50 runs are listed in Table 5. The FR of DAALO can achieve 100 %, while those of GA, PSO, ABC, and ALO are 92, 92, 94 and 90 %, respectively. For these feasible results, the standard deviation of the fitness value of DAALO is only 0.71, which is much smaller than other methods. And the best, mean, median and even worst cost value by DAALO are the minimum as well. Hence the performance of our proposed method is superior to that of other algorithms in terms of robustness and effectiveness.

Figure 10a shows the results of GA, PSO, ABC ALO and DAALO in case II. All the five paths are collision-free, but the path quality by DAALO is still the best. Figure 10b displays the corresponding convergence curves. Similar to case I, DAALO can find the optimal fitness value at 50 iterations, so the convergence speed is the fastest. Besides, the fitness value by DAALO after 200 iterations is the least, compared to other methods. The statistical results of case II are listed in Table 5. As case II is the city model where more intensive obstacles and local minima exist, the successful rate by five methods will reduce. But the FR of DAALO (94 %) is still much higher than that of GA (76 %), PSO (68 %), ABC (86 %) and ALO (70 %). Besides, the mean, median, worst and standard deviation of cost value by DAALO is still

Table 5 Performance comparisons of different methods in two cases

Algorithm	Best	Mean	Median	Worst	Standard deviation	FR (%)
Case I						
GA	455.50	469.33	468.58	505.87	20.47	92
PSO	467.60	513.79	508.33	591.54	32.40	92
ABC	454.25	461.96	457.90	490.11	9.36	94
ALO	455.15	473.00	463.39	536.39	24.39	90
DAALO	**454.17**	**455.10**	**454.93**	**457.25**	**0.71**	**100**
Case II						
GA	464.03	480.57	486.29	559.30	26.31	76
PSO	494.97	577.97	581.44	646.28	47.62	68
ABC	**462.27**	471.37	467.10	509.06	11.18	86
ALO	463.09	534.58	537.16	614.49	48.68	70
DAALO	462.29	**466.29**	**464.59**	**507.25**	**8.24**	**94**

Bold values indicate the best or optimal result of each column



(a) Planned routes **(b)** Convergence curves

Fig. 10 Comparisons of results by various methods in case II

smaller than others. Although the best cost value of DAALO is larger than that of ABC, both of them are very close (462.29 and 462.27). Therefore, regardless of the complexity of environment, DAALO can obtain more excellent results.

6 Conclusion

This work proposes a new nature-inspired method called DAALO to solve the UAV route planning problem. The basic ALO method mimics and models the hunting behavior of antlions and the entrapments of ants. To further improve the performance of ALO, DAALO is presented by introducing the Levy flight and the dynamic adaption of trap size. The experiment results in two typical cases show that DAALO is superior to other methods in terms of robustness, convergence speed and accuracy, local minima avoidance. Compared to other intelligent methods, DAALO can plan a smoother and shorter route, which is more suitable for route planning of UAVs under various constraints. The main contributions of this paper are the improvement to ALO and the first application of this newly proposed algorithm to plan feasible 2D UAV route.

In our future work, we will focus on the route planning problem in 3D environment, which is challenging to the computation efficiency and constraint handling of DAALO. Besides, we will extend our study to the cooperative route planning of multiple UAVs.

Acknowledgments This study was supported by the National Natural Science Foundation of China (No. 61175084) and Program for Changjiang Scholars and Innovative Research Team in University (No. IRT13004).

Compliance with ethical standards

Conflict of interest The authors declare that they do not have any conflicts of interest to this work.

References

Adolf FM, Andert F (2011) Rapid multi-query path planning for a vertical take-off and landing unmanned aerial vehicle. J Aeros Comp Inf Com 8(11):310–327

Ahmed F, Deb K (2013) Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. Soft Comput 17(7):1283–1299

Back T (1996) Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, New York, USA

Banerjee B, Chandrasekaran B (2013) A framework of Voronoi diagram for planning multiple paths in free space. J Exp Theor Artif In 25(4):457–475

Besada-Portas E, de la Torre L, de la Cruz JM, de Andres-Toro B (2010) Evolutionary trajectory planner for multiple UAVs in realistic scenarios. IEEE T Robot 26(4):619–634

Bollino K, Lewis LR, Sekhavat P, Ross IM, (2007) Pseudospectral Optimal Control: A Clear Road for Autonomous Intelligent Path Planning. In: AIAA Infotech@Aerospace, (2007) Conference and Exhibit. Rohnert Park, California, USA

de la Cruz JM, Besada-Portas E, de la Torre L, de Andres-Toro B, Lopez-Orozco JA (2008) Evolutionary path planner for UAVs in realistic environments. Genetic and Evolutionary Computation Conference. Atlanta, Georgia, USA, pp 1447–1484

Fu YG, Ding MY, Zhou CP (2013) Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV. IEEE T Syst Man Cy A 43(6):1451–4565

Goerzen C, Kong Z, Mettler B (2010) A survey of motion planning algorithms from the perspective of autonomous UAV guidance. J Intell Robot Syst 57(1–4):65–100

Hrabar S (2008) 3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs. Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice, FRANCE, pp 22–26

Hwang JY, Kim JS, Lim SS (2003) A fast path planning by path graph optimization. IEEE T Syst Man Cy A 33(1):121–128

Jaradat M, Garibeh MH, Feilat EA (2012) Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field. Soft Comput 16(1):153–164

Karaman S, Walter MR, Perez A (2011) Anytime motion planning using the RRT*. Proceedings of IEEE International Conference Robotics and Automation. Shanghai, China, pp 1478–1483

Karimi J, Pourtakdoust SH (2013) Optimal maneuver-based motion planning over terrain and threats using a dynamic hybrid PSO algorithm. Aerosp Sci Technol 26(1):60–71

Khaitib O (1986) Real-Time obstacle avoidance for manipulators and mobile robots. Int J Rob Res 5(1):90–98

Lam TM, Boschloo HW, Mulder M, van Paassen MM (2009) Artificial force field for haptic feedback in UAV teleoperation. IEEE T Syst Man Cy A 39(6):1316–1330

Li P, Duan HB (2012) Path planning of unmanned aerial vehicle based on improved gravitational search algorithm. Sci China Technol Sc 55(10):2712–2719

Liu W, Zheng Z, Cai KY (2013) Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval. Chinese J Aeronaut 26(3):646–660

Ma CS, Miller RH (2006) MILP optimal path planning for real-time applications. Proceedings of the American Control Conference. Minneapolis, MN, pp 4945–4950

Mirjalili S (2015) The Ant Lion Optimizer. Adv Eng Softw 83:80–98

Oz I, Topcuoglu HR, Ermis M (2013) A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles. Simul-T Soc Mod Sim 89(8):903–920

Pavlyukevich I (2007) Levy flights, non-local search and simulated annealing. J Comput Phys 226(2):1830–1844

Szczerba R, Galkowski P, Glicktein I, Ternullo N (2000) Robust algorithm for real-time route planning. IEEE T Aero Elec Sys 36(3):869–878

Valian E, Mohanna S, Tavakoli S (2011) Improved cuckoo search algorithm for feed forward neural network training. International Journal of Artificial Intelligence & Applications 2(3):36–43

Vera S, Cobano JA, Heredia G, Ollero A (2014) An hp-adaptative pseudospectral method for collision avoidance with multiple UAVs in real-time applications. IEEE International Conference on Robotics & Automation. Hongkong, China, pp 4717–4722

Viswanathan GM, Afanasyev V, Buldyrev SV et al (2000) Lévy flights in random searches. Physica A 282(1):1–12

Wang HL, Lyu WT, Yao P,liang X, Liu C,(2015) Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system. Chinese J Aeronaut 28(1):229–239

Wu Y, Qu XJ (2013) Path planning for taxi of carrier aircraft launching. Sci China Technol Sc 56(6):1561–1570

Yang XS (2011) Nature-inspired metaheuristic algorithms. Luniver Press, United Kingdom

Yao P, Wang HL, Su ZK (2015a) UAV feasible path planning based on disturbed fluid and trajectory propagation. Chinese J Aeronaut 28(4):1163–1177

Yao P, Wang HL, Su ZK (2015b) Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment. Aerosp Sci Technol 47:269–279

Yershov DS, Lavalle SM (2011) Simplicial Dijkstra and A* algorithms for optimal feedback planning. Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. CA, USA, San Francisco, pp 3862–3867

Zhang Y, Chen J, Shen LC (2013) Real-time trajectory planning for UCAV air-to-surface attack using inverse dynamics optimization method and receding horizon control. Chinese J Aeronaut 26(4):1038–1056

Zhang X, Chen J, Xin B, Peng ZH (2014) A memetic algorithm for path planning of curvature-constrained UAVs performing surveillance of multiple ground targets. Chinese J Aeronaut 27(3):622–633

Zhang SY, Yu JQ, Sun HD (2015) UAV path planning via Legendre pseudospectral method improved by differential flatness. the 27th Chinese Control and Decision Conference. Qingdao, China, pp 2580–2584

Zhang XY, Duan HB (2015) An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning. Appl Soft Comput 26(1):270–284

Zhu ZX, Wang FX, Shen H, Sun YW (2015) Global path planning of mobile robots using a memetic algorithm. Int J Syst Sci 46(11):1982–1993

Zhu WR, Duan HB (2014) Chaotic predator-prey biogeography-based optimization approach for UCAV path planning. Aerosp Sci Technol 32:153–161