Applied Soft Computing Journal 89 (2020) 106099

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning☆

Chengzhi Qu, Wendong Gai *, Maiying Zhong, Jing Zhang

Shandong University of Science and Technology, Qingdao 266590, China

ARTICLE INFO

Article history:
Received 21 February 2019
Received in revised form 20 November 2019
Accepted 14 January 2020
Available online 20 January 2020

Keywords:
Unmanned aerial vehicles (UAVs)
Three-dimensional path planning
Reinforcement learning
Grey wolf optimizer

ABSTRACT

Unmanned aerial vehicles (UAVs) have been used in wide range of areas, and a high-quality path planning method is needed for UAVs to satisfy their applications. However, many algorithms reported in the literature may not feasible or efficient, especially in the face of three-dimensional complex flight environment. In this paper, a novel reinforcement learning based grey wolf optimizer algorithm called RLGWO has been presented for solving this problem. In the proposed algorithm, the reinforcement learning is inserted that the individual is controlled to switch operations adaptively according to the accumulated performance. Considering that the proposed algorithm is designed to serve for UAVs path planning, four operations have been introduced for each individual: exploration, exploitation, geometric adjustment, and optimal adjustment. In addition, the cubic B-spline curve is used to smooth the generated flight route and make the planning path be suitable for the UAVs. The simulation experimental results show that the RLGWO algorithm can acquire a feasible and effective route successfully in complicated environment.

1. Introduction

In aerospace domain, unmanned aerial vehicles (UAVs) have been demonstrated as the representation of high potential and challenging technologies in recent years [1]. As a kind of modern aerial equipment, UAVs have been used in various areas, such as search, rescue, mapping and surveillance [2,3]. However, threats like buildings, trees, mountains, i.e., make it very difficult for UAVs to accomplish missions efficiently [4,5]. Therefore, a predefined path is needed for UAVs to satisfy mission requirement, and path planning methods are required to acquire feasible and efficient solutions. Traditional methods, such as A* algorithm [6], artificial potential field [7], linear programming [8] and random trees [9] are raised to solve the path planning problem. But most of these methods suffer from the high time complexity and local minima trapping when UAVs plan path with multiple constraints.

As a set of nature-inspired algorithms, meta-heuristic algorithms are originated from imitating biological interactive behaviors or physical phenomena [10,11]. The complicated real-world optimization problems for which traditional methods are not useful can be solved by the meta-heuristic algorithms [12]. Recently, series of meta-heuristic algorithms have been used to solve the UAVs path planning problem by considering the path planning problem as the optimization problem [13]. For example, Ref. [14] presented an improved bat algorithm (IBA) to solve the three-dimensional path planning problems for uninhabited combat air vehicles (UCAV). Ref. [15] used the multiverse optimizer (MVO) for solving the two-dimensional UAVs path planning problem. An enhanced discrete particle swarm optimization (DPSO) was developed by Ref. [16] to solve the path planning problem for UAVs vision-based surface inspection. Ref. [17] proposed an improved whale optimization algorithm (IWOA) to acquire a feasible route for solar-powered UAVs in urban environment.

Grey wolf optimizer (GWO) algorithm was a new meta-heuristic algorithm and introduced by Mirjalili in 2014. This algorithm mimics the social hunting behavior of grey wolf [18]. Superior to other metaheuristic algorithm, GWO is a simple swarm-based method which simulates the social behavior and leadership hierarchy of the grey wolf. The bionic structure of the GWO algorithm makes it has the advantages of implementation and flexibility [19]. The GWO algorithm has been applied to solve many engineering application and control problems, such as load frequency control [20], feature selection [21], and UAVs path planning [22]. Ref. [23] presented an inspired grey wolf optimizer (IGWO) to solve numerical optimization problems. Ref. [24] proposed an astrophysics-inspired grey wolf algorithm to solve engineering design problems. An exploration-enhanced grey wolf optimizer (EEGWO) was presented by Ref. [25] to solve high-dimensional numerical optimization. Although those modified

☆ This work is supported by National Natural Science Foundation under Grant 61603220, 61873149, 61733009; the Research Fund for the Taishan Scholar Project of Shandong Province of China; SDUST Young Teachers Teaching Talent Training Plan under Grant BJRC20180503.

* Corresponding author.
E-mail address: gwd2011@sdust.edu.cn (W. Gai).

https://doi.org/10.1016/j.asoc.2020.106099
1568-4946/

algorithms have been proved that various optimization problems are solved efficiently, there are still some drawbacks as follow:

- All the wolves are forced to perform same behavior during the search process. The characteristic of each individual is not considered, and the exploration or exploitation operations are performed uniformly for the whole population.
- When the dimensions of optimization problem increase, the performance of the GWO algorithm drops like other metaheuristic algorithms. However, the UAVs path planning problems usually own high dimensions in complex flight environment.

Reinforcement learning (RL) is a branch of machine learning [26]. The essence of RL is that an agent discovers an optimal policy autonomously by interacting with the environment to maximize a long-term reward [27]. The RL agent can receive evaluative reward from the environment after adopting an action in some state, and the performance of subsequent actions will be improved, which is closer to the human learning process [28]. Since the pioneering work of Ref. [29], reinforcement learning has attracted increasing attention. Ref. [30] presented a deterministic improved Q-learning method to solve the mobile robot path planning problems. Compared to the classical Q-learning, the proposed algorithm has a smaller time complexity. To solve the problem of unmanned ground vehicle (UGV) trajectory tracking, Ref. [31] developed a reinforcement learning based deterministic policy gradient (DPG) algorithm to model the controller for tracking the optimal path. A modified reinforcement learning algorithm was presented by Ref. [32] to solve the multi-agent systems path planning problem in unknown environment. The environment is estimated by the greedy actions using neural networks and kernel smoothing method. Ref. [33] used the deep reinforcement learning paradigm as a framework to achieve the adaptive control of autonomous underwater vehicles (AUVs). The experiment results show that the deep reinforcement learning approach owns strong applicability for the autonomous robot control problem.

To overcome the defects of the GWO algorithm, this paper proposes a novel reinforcement learning-based grey wolf optimizer algorithm (called RLGWO) to solve UAVs path planning problem. The optimal path is computed by the proposed algorithm off-line, and it is smoothed by the cubic B-spline curve. The proposed algorithm integrates the essential characteristics of RL and GWO. The individuals of the GWO are defined as agents in RL and the cost function values are regarded as the evaluative signals. Comparing with other improved GWO algorithm, this paper differs in the form that reinforcement learning method is inserted into RLGWO to choose the operation of each individual. Meanwhile, considering that the proposed algorithm is designed to serve for UAVs path planning, four operations have been developed for each individual: exploration, exploitation, geometric adjustment, and optimal adjustment. The geometric adjustment operation is developed to ensure the feasibility and the smoothness of the UAVs flight path. And the optimal adjustment operation is introduced to make the current trajectory jump out of local optimum. In addition, each operation will generate a positive or negative reward according to its result. The main contributions of this paper are summarized as follows:

(1) A novel reinforcement learning-based grey wolf optimizer algorithm called RLGWO is proposed to solve the UAVs three-dimensional path planning problem.

(2) The RLGWO includes four operations: exploration, exploitation, geometric adjustment and optimal adjustment. Each individual in RLGWO perform their operations independently.

(3) The geometric adjustment and optimal adjustment operations are developed to solve the problem of trapping in local optimization and unsmooth for UAVs path planning.

The structure of this paper is organized as follows. The mathematical model of UAV path planning is described in Section 2.

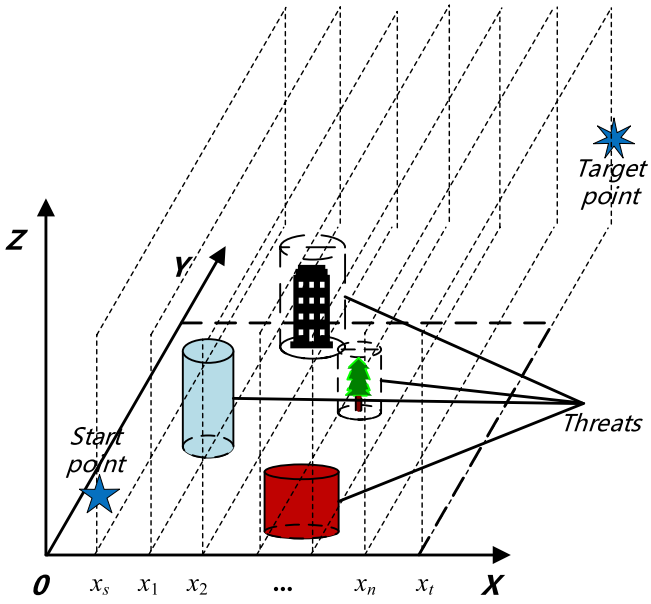


Fig. 1. Schematic diagram of planning space.

Section 3 explains the principles of the basic GWO algorithm and the reinforcement learning. The detailed implementation of the proposed RLGWO algorithm is described in Section 4. Section 5 describes the route smoothing method by the cubic B-spline curve. In Section 6, the comparison experiments are finished. Finally, Section 7 summarizes the conclusions.

2. Mathematical model in UAVs path planning

Path planning is a critical component of UAVs mission planning, and it is used to find the optimal flight route under the constraints such as mountains, buildings, and other threats. In this paper, we assume that the UAVs maintains constant speed during its mission, and the mathematical model is described as follows.

2.1. Threat resource model in UAVs path planning

The optimal route of UAVs path planning acquired from start point to target point need to consider all the threats and mission requirements. Motivated by the Ref. [34], the starting point has been defined as S and the target point has been defined as T, which is illustrated in Fig. 1. The threat obstacles are represented in the form of cylinders.

First, the coordinate of S is set as (x_s, y_s, z_s) and the coordinate of T is set as (x_t, y_t, z_t). The x-axis range has been divided into n+1 equal portions according to the point x_s and x_t, the corresponding split points are defined as $\{x_k, k = 1, 2, \ldots, n\}$. The waypoints (x_k, y_k, z_k) are distributed on the vertical plane corresponding to the split points. Therefore, the whole flight path is the connection of these discrete points. The y-axis and z-axis coordinates of the waypoints are defined in the variable $X_i = [y_{i1}, y_{i2}, \ldots, y_{in}, z_{i1}, z_{i2}, \ldots, z_{in}], i = 1, 2, \ldots, N$, in this way, the path planning problem is transformed to $2n$ dimensional optimization problem.

2.2. Cost function and performance constraints

It is known that the best flight route of UAVs path planning is often related to the straight path between S and T. Considering

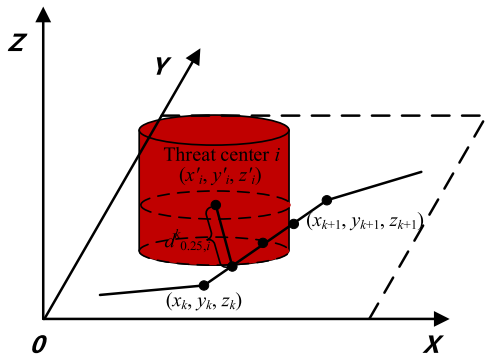


Fig. 2. Computation of threat cost.

this characteristic, the UAVs path planning cost function J_{cost} is defined as follows:

$$\begin{aligned} J_{\text{cost}} &= \mu_1 J_{fuel} + \mu_2 J_{threat} + \mu_3 J_{deviation} \\ &= \mu_1 \int_0^{length} W_{fuel}\mathrm{d}l + \mu_2 \int_0^{length} W_{threat}\,\mathrm{d}l \\ &\quad + \mu_3 \int_0^{length} W_{deviation}\mathrm{d}l \end{aligned} \tag{1}$$

where J_{fuel} is the fuel cost, J_{threat} is the threat cost, $J_{deviation}$ is the deviation cost, μ_i, $i = 1, 2, 3$ is a weighting parameter between 0 and 1, l presents the line segment of the whole route $\{l_k, k = 1,2, \ldots, n, n+1\}$. W_{fuel}, W_{threat} and $W_{deviation}$ represent the fuel cost, threat cost and the deviation cost on each path segment, and *length* denotes the length of the created flight path.

The deviation cost of the segment l_k is calculated as follows:

$$W_{deviation,l_k} = \sqrt{(y_k - y_{kl})^2 + (z_k - z_{kl})^2} \tag{2}$$

where (x_k, y_{kl}, z_{kl}) is the corresponding coordinate of the split point x_k projected onto the straight path between S and T.

The threat cost of the segment l_k is calculated at five points (include the start point and target point of l_k), as shown in Fig. 2.

If the path segment falls into a threat circle, the W_{threat} is calculated as follows:

$$\begin{aligned} W_{threat,l_k} = \frac{l_k}{5} \cdot \sum_{i=1}^{m} \frac{1}{5} \cdot \Big(& \left|d_{0,i}^k - r_i\right| \\ & + \left|d_{0.25,i}^k - r_i\right| + \left|d_{0.5,i}^k - r_i\right| \\ & + \left|d_{0.75,i}^k - r_i\right| + \left|d_{1,i}^k - r_i\right| \Big) \end{aligned} \tag{3}$$

where m denotes the number of threatening circles, $d_{0.25,i}^k$ denotes the distance between the ith threat center and the 0.25 point on the segment, r_i refers to the radius of the ith threat. It is assumed that the speed of the UAVs is a constant, therefore, the J_{fuel} can be considered as the length of the path.

Considering generating a suitable route for the UAVs, the yawing angle and the pitch angle constraints are introduced as follows:

$$\varphi_k = \left|\arctan\left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right)\right| \le \varphi^{\max},\ k = 1, 2, 3, \ldots, n-1 \tag{4}$$

$$\theta_k = \left|\arctan\left(\frac{z_{k+1} - z_k}{x_{k+1} - x_k}\right)\right| \le \theta^{\max},\ k = 1, 2, 3, \ldots, n-1 \tag{5}$$

where $\varphi^{\max}$ is the maximum yawing angle, $\theta^{\max}$ is the maximum pitch angle, φ_k and θ_k are the yawing angle and the pitch angle of the path point (x_k, y_k, z_k).

Therefore, the UAVs path planning problem can be turned to the optimization problem to minimize J_{cost} under the constraint conditions Eqs. (4) and (5).

3. Preliminary knowledge and basic idea

3.1. 1the GWO algorithm

In the GWO algorithm, the individual with the best fitness is named α wolf. The second and third individuals with better fitness are named β wolf and δ wolf. These three wolves form the leader group. The rest of the individuals in the population are considered as ω. To model the encircling behavior that the grey wolves hunt the prey, Eqs. (6) and (7) are designed as follows:

$$D_i = \left|C_i \cdot X_p(t) - X_i(t)\right| \tag{6}$$

$$X_i(t+1) = X_p(t) - A_i \cdot D_i \tag{7}$$

where t is the current iteration, X_i indicates the position of a grey wolf. X_p denotes the position of the prey. The coefficient parameters A_{i} and C_{i} are shown as:

$$\begin{cases} A_i = 2a \cdot r_1 - a_w \\ C_i = 2 \cdot r_2 \\ a_w = 2 - 2t/t_{\max} \end{cases} \tag{8}$$

where r_1 and r_2 are random parameters in [0,1], a_w is linearly decreased from 2 to 0.

Each ω wolf updates its position according to the leader group (α wolf, β wolf and δ wolf) as follows:

$$D_\alpha = |C_1 \cdot X_\alpha(t) - X_i(t)| \tag{9}$$

$$D_\beta = \left|C_2 \cdot X_\beta(t) - X_i(t)\right| \tag{10}$$

$$D_\delta = |C_3 \cdot X_\delta(t) - X_i(t)| \tag{11}$$

$$X_1 = X_\alpha(t) - A_1 \cdot D_\alpha \tag{12}$$

$$X_2 = X_\beta(t) - A_2 \cdot D_\beta \tag{13}$$

$$X_3 = X_\delta(t) - A_3 \cdot D_\beta \tag{14}$$

$$X_{\text{i}}(t+1) = (X_1 + X_2 + X_3)/3 \tag{15}$$

where X_α, X_β, and X_δ denote the position of the leader group, X_{i} is the current individual's position.

3.2. Reinforcement learning

Over the passage of time, many significant breakthroughs have been developed in reinforcement learning, and RL can be categorized into two groups: policy-based methods and value-based methods. Q-learning algorithm is a typical representative of value-based methods. During the learning, the agent performs action with the highest expected Q-values to estimate the optimal policy. The Q-table is updated based on the reward dynamically, and it is computed as follows:

$$\begin{aligned} Q_{t+1}(s_t, a_t) \leftarrow Q(s_t, a_t) + \lambda[r_{t+1} \\ + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \end{aligned} \tag{16}$$

where s_t is the current state, s_{t+1} is the next state, a_t is the current action, γ is a discount parameter, λ is the learning rate, r_{t+1} is the immediate reinforcement reward acquired from the execution of a_t at s_t, $Q(s_{t+1}, a)$ is the estimated Q-value when the action a is performed at state s_{t+1}. The procedural code of the Q-learning algorithm is shown follows:

The Q-learning algorithm pseudocode

Initialize
Set the state s and the action a
For each state s_i and action a_i
Set $Q(s_i, a_i) = 0$
End For
Randomly choose an initial state s_t
While the terminal condition is not reached **do**
Choose the best action a_t from the current state s_t from Q-table
Execute action a_t, then get the immediate reward
Find out the new state s_{t+1}
Acquire the corresponding maximum Q-value of s_{t+1}
Update the Q-table by Eq. (16)
Update the state $s_t \leftarrow s_{t+1}$
End While

3.3. Basic idea of the proposed algorithm

To solve complex optimization problems, all the metaheuristic algorithms have been designed to find a proper balance between global exploration ability and local exploitation ability. In GWO, the search operations (exploration exploitation) are selected by the parameter A_i. When the random values of A_i are in [-1,1], the wolves move with smaller distance uniformly, which means a process of local search. When $|A_i| > 1$, the wolves are forced to make a global search uniformly. However, the global minimum will not be guaranteed by the unified search behavior. The independent search operation will become a better indicator to ensure the final success, and the embedding of reinforcement learning can accomplish this task well. Meanwhile, the geometric adjustment operation and the optimal adjustment operation are introduced to smooth the path and jump out of local optimum. Therefore, the novel reinforcement learning based grey wolf optimizer algorithm called RLGWO algorithm has been presented.

4. The development of the RLGWO algorithm

4.1. The RLGWO structure

During the proceed of RLGWO, each individual is influenced by the leader group in the population on the base of their accumulated performance. The pseudocode of RLGWO is illustrated. This code will be repeated until meeting the maximum iteration. Fig. 3 shows the whole structure of the RLGWO algorithm. Individuals of GWO are regarded as the train agents of reinforcement learning. The search space is regarded as the interactive environment. Four operations represented the states have been developed for each individual: exploration, exploitation, geometric adjustment, and optimal adjustment. And the change of state is defined as the action. According to the accumulated performance, the individual switches operation (state) adaptively. If the execution of operation results in a better performance, a positive reward will be given, otherwise the negative reward is given to complete the punishment.

The RLGWO algorithm pseudocode

Initialize
Set the basic parameters
Set the state $s = \{s_1, s_2, s_3, s_4\}$ and action $a = \{a_1, a_2, a_3, a_4\}$
Set the initial Q-table: Q(s, a)=0
Initialize population position $X_i, i = 1,2,...,N$
Calculate the fitness of each individual $f(X_i)$
Find the initial leader group: $X_\alpha, X_\beta, X_\delta$
Set t=0
Optimize
While $t < t_{\max}$ **do**
Calculate a_w, A, C, λ
For each individual X_i
Choose the best a for the current s from Q-table
Switch action
Case 1: exploration
Update the position of X_i by the Eq.(15)
Case 2: exploitation
Update the position of X_i by the Eq.(19)
Case 3: geometric adjustment
For each dimension of X_i
Update the position of X_i by the Eq.(20) and Eq.(21)
End For
Case 4: optimal adjustment
For each dimension of X_i
Repeat T times
Update the position of X_i by the Eq.(27)
Compute fitness
End For
Set the reward to a negative cost value r_p by the Eq.(28)
Update the position of X_i
End Switch
If action is not optimal adjustment
Update fitness
Get the reward r by the Eq. (18)
End If
Update X_α, X_β, X_δ
Update the Q-table
t=t+1
End While
Return results
Terminate

The parameter setting of the learning rate λ requires careful consideration in the Q-learning algorithm. When the learning rate λ approaches 1, the newly gained information is more valuable for the update of the Q-value in Q-table. And the existing information will not be ignored when a small value of λ is given. To maximize learning from the search space, λ can be reduced adaptively from a high value during the iteration process.

$$\lambda = \frac{\lambda_{initial} + \lambda_{final}}{2} - \frac{\lambda_{initial} - \lambda_{final}}{2} \cdot \cos\left(\pi\left(1 - \frac{t}{t_{\max}}\right)\right) \quad (17)$$

where $\lambda_{initial}$ represents the initial value of λ, λ_{final} represents the final value of λ, t is defined as the current iteration number, $t_{\max}$ is defined as the maximum iteration number.

The reward r is set as follows:

$$r = \begin{cases} 1, & if\ the\ fitness\ improved \\ -1, & otherwise \end{cases} \quad (18)$$

Fig. 4 illustrates the update method of the Q-table. The Q-table is designed as a 4 × 4 matrix. The columns of the Q-table represent the action and the rows represent the state (exploration, exploitation, geometric adjustment, and optimal adjustment). Each individual has its own Q-table to ensure that the learning process is independent.

4.1.1. The exploration and exploitation operations

In the classical GWO algorithm, the exploration operation is performed at the beginning of the iteration process. Towards the end of the iteration, the exploitation operation is executed. However, motivated by the description in [35], individuals need to have the capacity that any operations can be executed at any time during the iteration process. Reinforcement learning will

Fig. 3. The flow chart of the RLGWO algorithm.

become a better indicator to ensure that the best operation will be performed for each individual according to the accumulated performance.

The individual X_i updates its position affected by the current leader group. Based on the descriptions of Section 3, the convergence parameter a_w controls the movement of the individual. So, in the exploration operation, the parameter a_w should be set a high value to force the individual make a global search. Moreover, the individual should be impacted by the current three best individuals uniformly in the exploration mode to weaken the influence of the α wolf. After applying Eq. (15), the result of the exploration operation is shown in Fig. 5.

In the exploitation operation, all individuals should move slowly towards the current best individual. Therefore, a_w should be set a low value low to ensure a local search around the α wolf. Moreover, the individual should be mainly impacted by α wolf in the exploitation mode. Fig. 6 shows the exploitation operation after applying Eq. (19).

$$X_i(t+1) = \sigma_1 X_1 + \sigma_2 X_2 + \sigma_3 X_3 \tag{19}$$

where σ_i, $i = 1, 2, 3$ is a weighting parameter between 0 and 1.

4.1.2. The geometric adjustment operation

The movements of individuals are affected by the leader group and the convergence parameter a_w. But the distribution of the individual's elements is scattered under the premise of proper fitness. When the individual is regarded as a group of waypoints, the feasibility and the smoothness of the UAVs flight path are substantially affected.

Therefore, the geometric adjustment operation has been proposed in this paper to solve this problem. The pseudocode of the geometric adjustment operation is given. As shown in Fig. 7, this operation can result in a more efficient and direct path from position A to E in the form of creating a middle waypoint. The middle waypoint is calculated as:

$$X_{my}(k) = (X_m(k-1) + X_m(k+1))/2 \tag{20}$$

$$X_{mz}(k) = (X_m(k+n+1) + X_m(k+n+3))/2 \tag{21}$$

The pseudocode of the geometric adjustment

For k=1 to n-1
  Set X_m=[y_s, X_i(1:n), y_t, z_s, X_i(n+1:2n), z_t]
  Compute the $X_{my}(k)$ using Eq. (20)
  Compute the $X_{mz}(k)$ using Eq. (21)
  Set $X_m(k)$=$X_{my}(k)$
  Set $X_m(k+n+2)$=$X_{mz}(k)$
  Set X_m=[X_m(2:n+1), X_m(n+4:2n+3)]
  Compute fitness $f(X_m)$
  If $f(X_m)$<$f(X_i)$
    X_i=X_m
  End If
End For

State \ Action	exploration	exploitation	geometric adjustment	optimal adjustment
exploration	-0.9	35.1	0	0
exploitation	-6.3	-6.3	-3.6	-8.8
geometric adjustment	-30.2	-33.3	-33.3	-35.3
optimal adjustment	-1.8	-1.8	-1.8	-7.2

(a) Before Q-table update

State \ Action	exploration	exploitation	geometric adjustment	optimal adjustment
exploration	-0.9	35.1	0	0
exploitation	-6.3	-6.3	-3.6	-8.8
geometric adjustment	-34.2	-33.3	-33.3	-35.3
optimal adjustment	-1.8	-1.8	-1.8	-7.2

(b) After Q-table update

Fig. 4. The Q-table update.

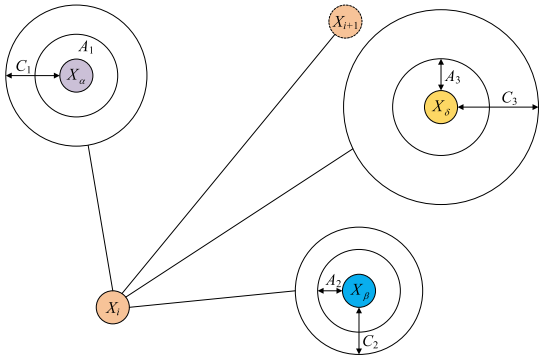


Fig. 5. The exploration operation.

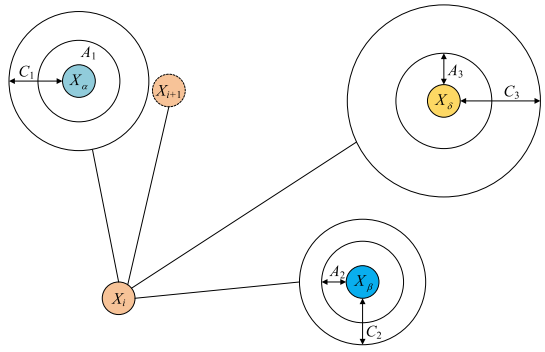
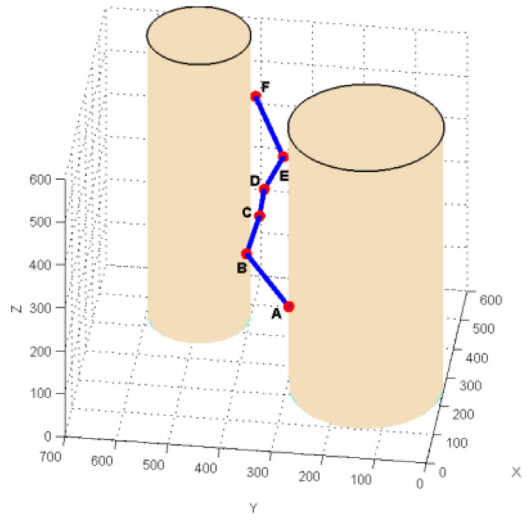
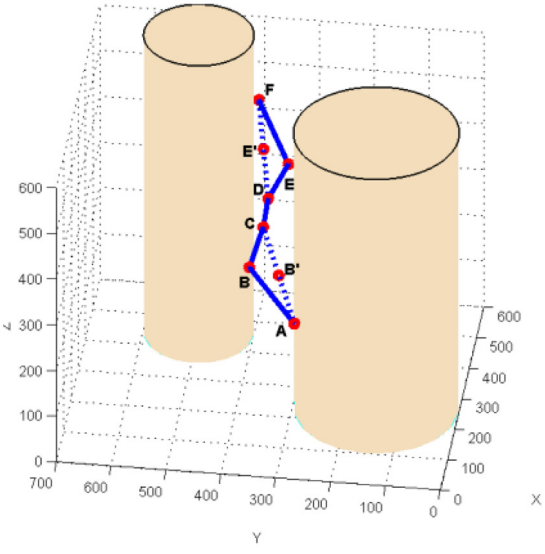


Fig. 6. The exploitation operation.



(a) Before the geometric adjustment operation



(b) After The geometric adjustment operation

Fig. 7. The geometric adjustment operation.

4.1.3. The optimal adjustment operation

For the three-dimensional path planning problem of UAVs, it is known that the best flight route is often related to the straight path between start point and target point. When there are obstacles existing, the optimal route is the feasible path close to the straight path. Therefore, the influence of the optimal adjustment operation is followed. This operation updates all dimensions of the current individual, and starting from the *y*-axis and *z*-axis elements corresponding to the first waypoint, moving it to the direction of the straight path. If a better result is obtained each time, the element is updated, otherwise the position is unchanged, and each waypoint is operated in turn.

The equation of the straight path has been acquired at the initialization stage of the algorithm. And the coordinates of the points on the line corresponding to the x-axis split points are defined as (x_k, y_{kl}, z_{kl}).

The detail procedure of the optimal adjustment operation is introduced as follows:

Step 1 Set

$$(P_{yk}, P_{zk}) = (X(k), X(k+n)) = (y_k, z_k) \tag{22}$$

Using (P_{yk}, P_{zk}) to represent the kth dimension of the current individual in y-axis and z-axis.

Step 2 The velocities of P_{yk} and P_{zk} are computed as follows:

$$\begin{cases} V_{\mathrm{yk}} = \dfrac{y_{kl} - P_{yk}}{2} + R_{yk} \\ V_{\mathrm{zk}} = \dfrac{z_{kl} - P_{zk}}{2} + R_{zk} \end{cases} \tag{23}$$

where

$$R_{yk} = \begin{cases} 0.2 \cdot V_{\mathrm{yk}}, & if\ fitness\ improved \\ -0.1 \cdot V_{\mathrm{yk}}, & otherwise \end{cases} \tag{24}$$

$$R_{zk} = \begin{cases} 0.2 \cdot V_{\mathrm{zk}}, & if\ fitness\ improved \\ -0.1 \cdot V_{\mathrm{zk}}, & otherwise \end{cases} \tag{25}$$

The position of (P_{yk}, P_{zk}) is updated as follows:

$$\begin{cases} P_{yk} = P_{yk} + V_{yk} \\ P_{zk} = P_{zk} + V_{zk} \end{cases} \tag{26}$$

Step 3 Set the point (x_k, P_{yk}, P_{zk}) as a provisional target point, compute the fitness of the path between the start point and the point (x_k, P_{yk}, P_{zk}), and the variate $(X(k), X(k+n))$ is updated as follows:

$$(X(k), X(k+n)) = \begin{cases} (P_{\mathrm{yk}}, P_{\mathrm{zk}}), & if\ fitness\ improved \\ (X(k), X(k+n)), & otherwise \end{cases} \tag{27}$$

Step 4 Repeat *Step* 2 and Step 3 T times ($T = t_{max} - t$), then set $k = k + 1$ and go to Step 1 until the termination condition $k > n$ is met.

The optimal adjustment operation is useful for UAVs path planning problem to escape the local optimization. However, this operation need to set a high cost at the beginning of the iteration, because much time consumed by executing this operation. Therefore, to prevent RLGWO from performing the optimal adjustment operation frequently at the beginning of the iteration process, an adaptive punishment r_p is defined as follows:

$$r_p = (r_{p,initial} - r_{p,final}) \cdot \left(1 - \frac{t}{t_{\max}}\right) + r_{p,final} \tag{28}$$

where $r_{p,initial}$ represents the initial punishment parameter, $r_{p,final}$ represents the final punishment parameter.

4.2. Computing complexity

As is shown in Fig. 3, the RLGWO algorithm can be divided into two phases. As the first phase of the program, the initialization phase is executed one time at the start, and the other phases are executed in each cycle. The individual's position in population is a vector with size of N. The dimension of each individual is N. The maximum iteration of the proposed algorithm is t_{max}. The current iteration is t. The computational complexity is mostly affected by the phase of the algorithm. The computing complexity of each phase is shown as follows:

Phase 1: Initialization

The population are initialized for the next work and the computing complexity of this phase is $O(N)$. Because the complexity of deciding on stopping criteria ended is $O(1)$, hence this phase runs in $O(N)$ complexity.

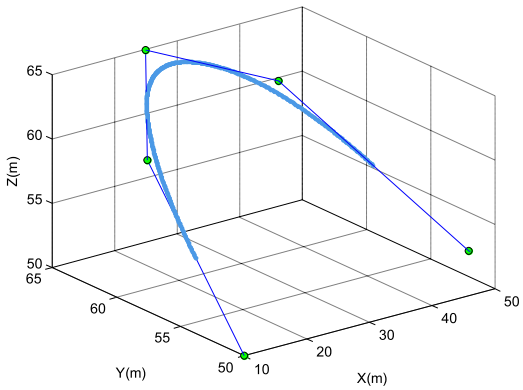


Fig. 8. The cubic B-spline curve.

Phase 2: Optimization

The optimization process of RLGWO includes four operations: exploration, exploitation, geometric adjustment, and optimal adjustment, Due to the character that each individual switches operation independently according to the accumulated performance, therefore, each operation is assumed to perform for all individuals to calculate the extreme computational complexity of each iteration.

In the exploration operation, all individuals should move fast to achieve global search impacted by the current three best individuals uniformly. In the exploitation operation, all individuals should move slowly towards the current best individual to ensure a local search. The computing complexity of exploration operation and exploitation operation are $O(N)$.

The geometric adjustment operation can ensure the feasibility and the smoothness of the planning path. The optimal adjustment operation updates all dimensions of the current individual to correct the planning path. The computing complexity of geometric adjustment operation is $O(N \cdot (D-1)/2)$. The computing complexity of optimal adjustment operation is $O(N \cdot D \cdot (t_{\max} - t)/2)$.

Therefore, the maximum computing complexity of the proposed algorithm is $O(N \cdot D \cdot (t_{\max} - t)/2)$ in each iteration, which proves that this algorithm owns fast execution speed.

5. Path smoothing

In most cases, the routes planning by meta-heuristic algorithms are usually unsmooth for UAVs. Therefore, the cubic B-spline curve method is introduced to smooth the generated path in this paper. As shown in Fig. 8, the technique of cubic B-spline is introduced to ensure the flight route flyable and smooth [36,37]. The B-spline curve has evolved from Bezier curves and inherits the advantages of geometrical invariability, convexity preserving and affine invariance.

The construction of the B-spline curves is based on blending functions, and the coordinates is defined as:

$$P(u) = \sum_{i=0}^{n} d_i N_{i,k}(u) \tag{29}$$

where $d_i (i = 0, 1, \ldots, n)$ are control points, $N_{i,k}(u)$ are the k-order normalized B-spline basic functions defined by the following

Cox–deBoor recursion formulas:

$$\begin{cases} N_{i,k}(u) = \begin{cases} 1, if\ u_i \le u \le u_{i+1} \\ 0,\ otherwise \end{cases} \\ N_{i,k}(u) = \dfrac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \dfrac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \\ define\ \dfrac{0}{0} = 0 \end{cases} \tag{30}$$

The basic functions are determined by a non-decreasing sequence of parameters called parametric knots $\{u_0 \le u_1 \le \cdots \le u_{n+k}\}$.

Compared with the Bezier curves, the B-spline curves method is particularly useful for smoothing path, and the degree of the polynomial will not be increased no matter how many points are added.

6. Experiment results

6.1. The three-dimension experiment comparison results

In this section, there are three simulation cases designed to evaluate the feasibility and effectiveness of the RLGWO algorithm. The size of the flight planning space is 2000 m * 2000 m * 1000 m. The start point is set to (0 m, 0 m, 0 m). The target point is set to (2000 m, 2000 m, 1000 m). The weighting parameters μ_1, μ_2, μ_3 are set as 0.2, 0.6, 0.2, respectively. To show the superiority of the proposed algorithm, the comparative results with the classical GWO IGWO [23], MGWO [24] and EEGWO [25] algorithms are also given. The maximum iteration number t_{max} is set as 1000, and the initial parameters of the five algorithms are list in Table 1.

The simulation experiments of all cases are repeated for 30 times independently. In the planning space, the threats are denoted by eight cylinders. The related information of the threats is shown in Table 2. The results of the simulation are demonstrated in Figs. 9–17. The comparison result is listed in Table 3. In this table, the mean, std, worst and optimal represent the mean fitness value, the standard deviation, worst fitness value and the optimal fitness value, respectively.

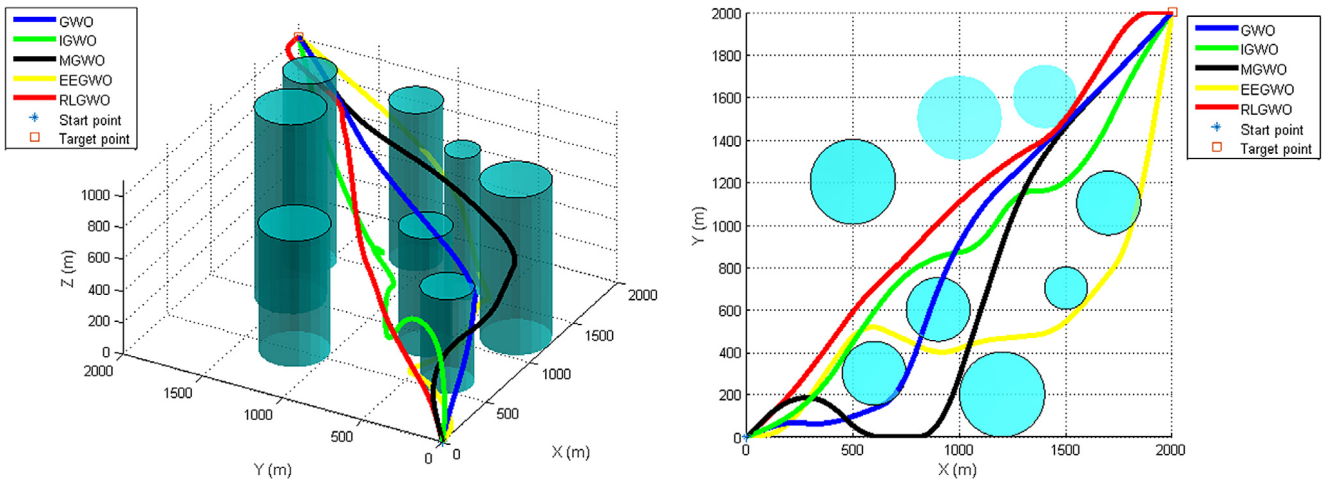
For the first case, Fig. 9 shows that the experimental results of the five algorithms have some differences. The trajectory planned by the GWO and EEGWO have touched the edge of the threats, the trajectory planned by the MGWO is trapped in local optimization, and the result of the IGWO is oscillatory. The result of the RLGWO maintains significant performance. The convergence cures of the five algorithms in Case 1 are illustrated in Fig. 10. It is obvious that the convergence effect of the proposed algorithm is better than the other algorithms. The RLGWO attains the global optimal value in iteration 60. The IGWO attains the local optimal value in iteration 160. The GWO and MGWO approach to their optimal value in iteration 680. The EEGWO finds its local optimal value in iteration 560. The statistical results are illustrated in Fig. 11 and Table 3. The optimal value of the GWO is 1474.8, but the worst value is 2303.3, and the standard deviation is 319.5. These data indicate that GWO algorithm has poor result for multiple independent runs. The optimal value of MGWO is 1486.4, but the worst value is 3365.1, the standard deviation is 731.3. These data indicate that MGWO algorithm has lower success rate. The IGWO and EEGWO algorithms have similar poor result. Compared with the other algorithms, the simulation result of the proposed algorithm has smaller optimal, worst, and mean values. Meanwhile, the standard deviation of the RLGWO is 80.3, which proves that RLGWO can search for the optimal path stably.

For the second case, we have increased the complexity of the plan space slightly. Fig. 12 shows that the RLGWO can find a feasible path to satisfy the path planning requirements with the smallest cost. The results of GWO, IGWO and EEGWO can satisfy the path planning requirements better. But the trajectory planned by the MGWO is trapped in local optimization. The convergence cures of the five algorithms in Case 2 are illustrated in Fig. 13. From these curves it can be seen that the best convergence rate belongs to the RLGWO algorithm. The proposed algorithm attains the global optimal value in iteration 35. The EEGWO and GWO attain their local optimal value in iteration 920 and 660. The MGWO approaches to its optimal value in iteration 650. The convergence effect of the IGWO algorithm is preferable but worse than the proposed algorithm. For the statistical results shown in Fig. 14 and Table 3, the RLGWO has the best performances for optimal, worst, mean and std values. The IGWO and EEGWO have close poor statistical results, and GWO has good results relatively. The MGWO has similar mean and optimal values with the IGWO and EEGWO, but the standard deviations of MGWO is larger. The standard deviation of the proposed algorithm is 80.1, which shows its superior performance.

In the third case, we have established a more complex flight environment to examine the performance of the five algorithms. Fig. 15 shows that only RLGWO can complete the path planning mission perfectly. The trajectory planned by GWO, IGWO and EEGWO algorithm cannot satisfy the path planning requirements. And the trajectory planned by the MGWO algorithm is trapped in local optimization. Fig. 16 shows the convergence cures of the five algorithms in Case 3. The RLGWO has the faster convergence rate. The IGWO has a relatively fast convergence rate but worst result. The other three algorithms have poor convergence rates. As the statistical results shown in Fig. 17 and Table 3, the proposed

Table 1
The information of algorithms.

Algorithm	Parameter	Value
GWO	Population size N	50
	Convergence parameter a_w	linearly decreased from 2 to 0
IGWO	Population size N	50
	Convergence parameter a_w	$1-\log(1+(e-1)\cdot t/t_{max})$
	Inertia weight w	$((t_{max}-t)\cdot 0.8/t_{max})+0.1$
MGWO	Population size N	50
	Convergence parameter a_w	linearly decreased from 2 to 0
	Control parameter C	$2\cdot rand-(a/2)$
EEGWO	Population size N	50
	Convergence parameter a_w	$2-2(t_{max}-t/t_{max})^{1.5}$
	Constant coefficient b_1	0.8
	Constant coefficient b_2	0.2
RLGWO	Population size N	10
	Convergence parameter a_w (exploration)	2
	Convergence parameter a_w (exploitation)	0.5
	Learning rate parameter $\lambda_{initial}$	0.9
	Weighting parameter σ_1	0.6
	Weighting parameter σ_2	0.3
	Weighting parameter σ_3	0.1
	Learning rate parameter λ_{final}	0.1
	Discount factor γ	0.5
	Punishment $r_{p,initial}$	−10
	Punishment $r_{p,final}$	−2



(a) The three dimensional comparative result

(b) The overlooking view of the comparison

Fig. 9. The comparative path planning result in Case 1.

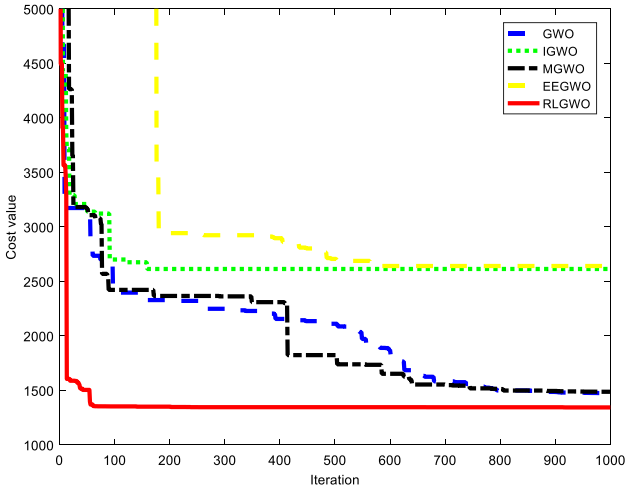


Fig. 10. The convergence cures of the five algorithms in Case 1.

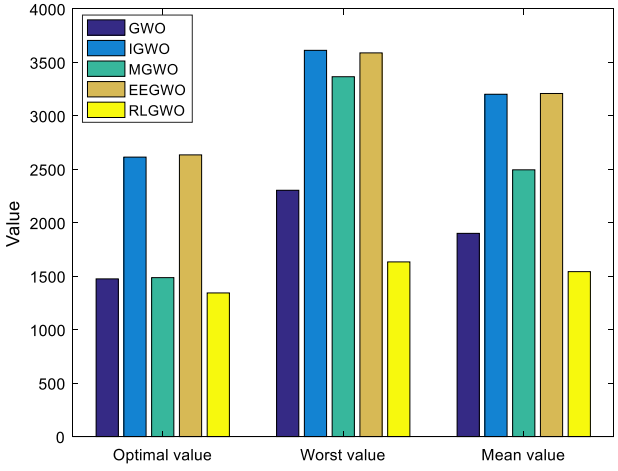


Fig. 11. The statistical results of the five algorithms in Case 1.

algorithm still provides the best results in optimal, worst, mean and std values. These results prove that RLGWO has excellent capability for UAV path planning in complex environment.

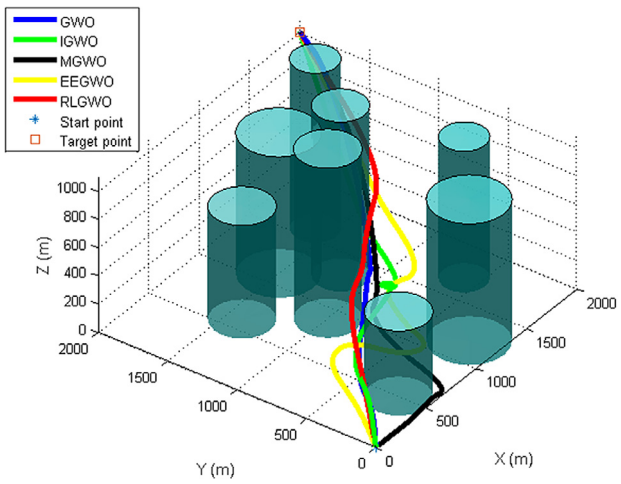
Table 2
The information of threat areas.

Case number	Threat center	Height	Threat radius
1	(600,300)	600	150
	(500,1200)	800	200
	(1200,200)	1000	200
	(1000,1500)	1200	200
	(1700,1100)	1000	150
	(900,60 0)	750	150
	(1500,700)	900	100
	(1400,1600)	1200	150
2	(500,200)	600	200
	(600,1400)	800	200
	(1200,200)	1000	250
	(1100,1500)	1000	250
	(1700,600)	1000	150
	(900,1000)	1200	200
	(1300,1200)	1200	170
	(1600,1600)	1200	150
3	(500,250)	600	250
	(760,1000)	1200	300
	(1100,200)	1000	200
	(500,1500)	1000	250
	(1500,900)	1000	200
	(1700,500)	750	250
	(900,1400)	900	200
	(1400,1600)	1200	300

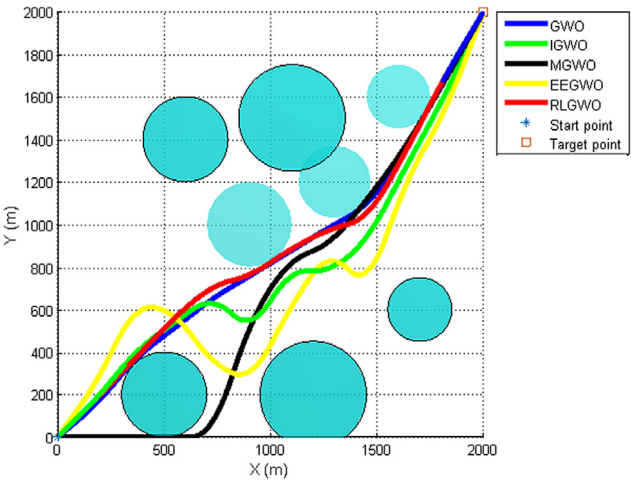
Table 3
Results comparison of the three cases.

		GWO	IGWO	MGWO	EEGWO	RLGWO
Case 1	Mean	1900.4	3201.1	2494.5	3208.3	1542.4
	Std	319.5	374.2	731.3	367.1	80.3
	Worst	2303.3	3611.8	3365.1	3587.9	1633.4
	Optimal	1474.8	2613.3	1486.4	2634.2	1343.2
Case 2	Mean	2542.3	2996.6	3064.2	3082.4	1841.9
	Std	438.8	734.3	1324.6	808.8	80.1
	Worst	2985.9	4805.9	5546.3	4501.2	1933.7
	Optimal	2098.5	2444.2	2380.6	2372.2	1670.9
Case 3	Mean	3137.9	5042.8	3458.7	4901.6	1971.6
	Std	872.5	2241.6	1042.9	1427.1	476.1
	Worst	5360.3	8347.8	5399.2	7617.2	3315.1
	Optimal	2335.7	3170.6	2013.1	2555.1	1664.3

In summary, from the above experimental results it can be seen that the RLGWO algorithm can search for a satisfactory path



(a) The three dimensional comparative result



(b) The overlooking view of the comparison

Fig. 12. The comparative path planning result in Case 2.

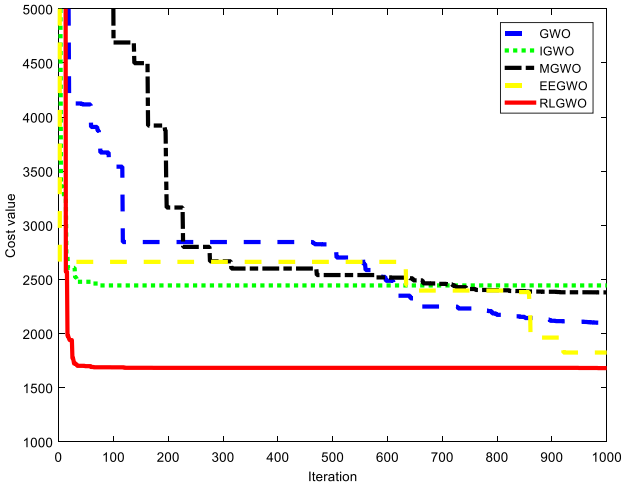


Fig. 13. The convergence cures of the five algorithms in Case 2.

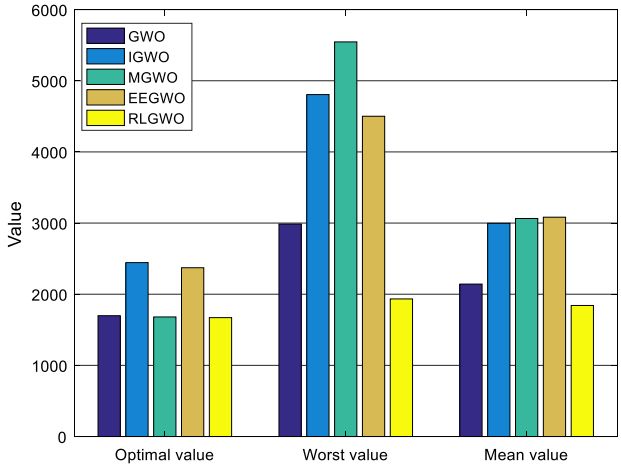


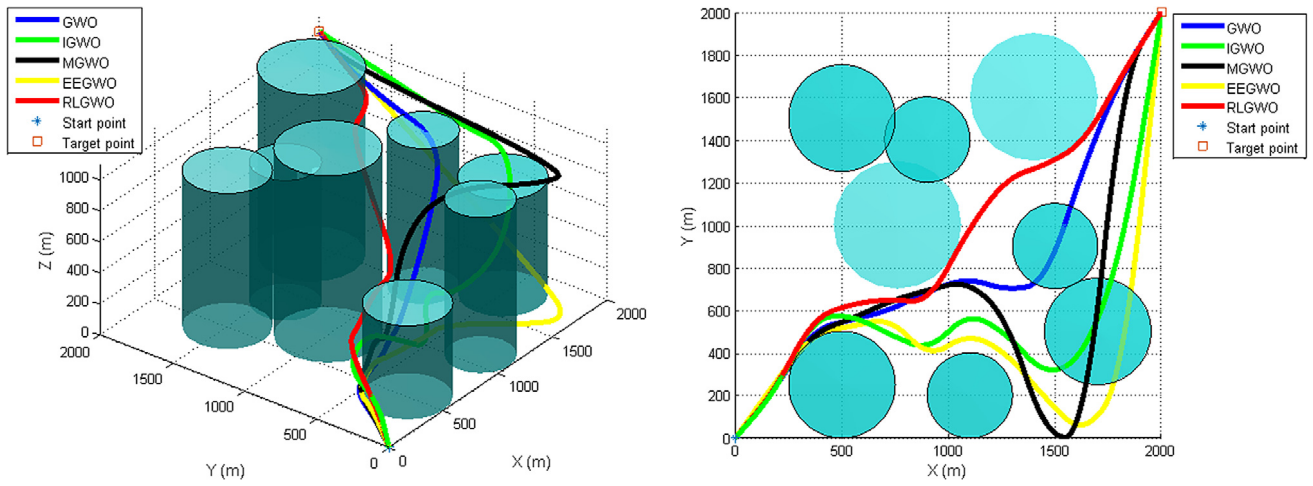
Fig. 14. The statistical results of the five algorithms in Case 2.

stably, especially in the complex flight environment. And the favorable performance of the proposed algorithm is verified. In the whole iterative process of RLGWO, all operations participate in the optimization process. Each individual's global search behavior is embodied by the exploration operation, and its local search behavior is affected by the exploitation operation. The geometric adjustment operation is performed to ensure the feasibility and the smoothness of the planning path. The optimal adjustment operation updates all dimensions of the current individual to correct the planning path. The switching of the four operations is coordinated by RL, which finally makes the RLGWO algorithm has excellent performance in the simulation experiment of UAVs three-dimensional path planning.

6.2. Analysis of the RLGWO control parameters

The key control parameters of RLGWO are shown in Table 1. Compared to other algorithms, because the optimal adjustment operation owns higher computing complexity in earlier iteration, the population size N of RLGWO need to be set as a smaller value. Meanwhile, smaller N will not decrease the search performance because each individual owns independent search behavior. The values of convergence parameter a_w and weighting parameter $\sigma_i, i = 1, 2, 3$ are set according to the characteristic of GWO. In GWO, the exploration and exploitation behaviors are selected by the parameter A_i. When the random values of A_i are in [-1,1], the wolves move with smaller distance, which means a process of exploitation. When $|A_i| > 1$, the wolves are forced to perform exploration operation. But the parameter A_i is mainly affected by the convergence parameter a_w. Thus, in the exploration operation of RLGWO, the parameter a_w should be set a high value to force the individual make a global search. In the exploitation operation, a_w should be set a low value to ensure a local search around the α wolf. Moreover, the decreasing weight $\sigma_i, i = 1, 2, 3$ represent that individuals should be mainly impacted by α wolf in the exploitation operation.

The parameter setting of the learning rate λ is affected by the Q-learning algorithm. When the learning rate λ approaches 1, the newly gained information is more valuable for the update of the Q-value in Q-table. And the existing information will not be ignored when a small value of λ is given. Thus, λ is reduced adaptively from a high value to maximize learning rate from the search space. Therefore, the $\lambda_{initial}$ is set to a high value, and λ_{final} is set to a low value. To prevent RLGWO from performing the optimal adjustment operation frequently at the beginning of the iteration process, an adaptive punishment r_p is defined, and $r_{p,initial}$ represents the initial high cost, $r_{p,final}$ represents the final low cost.



(a) The three dimensional comparative result

(b) The overlooking view of the comparison

Fig. 15. The comparative path planning result in Case 3.

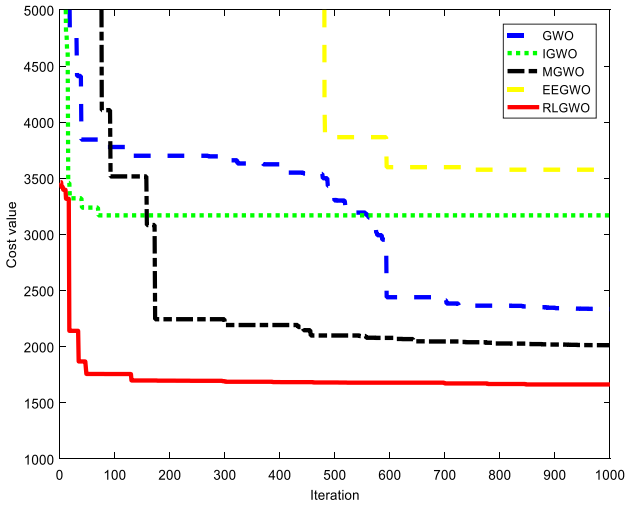


Fig. 16. The convergence cures of the five algorithms in Case 3.

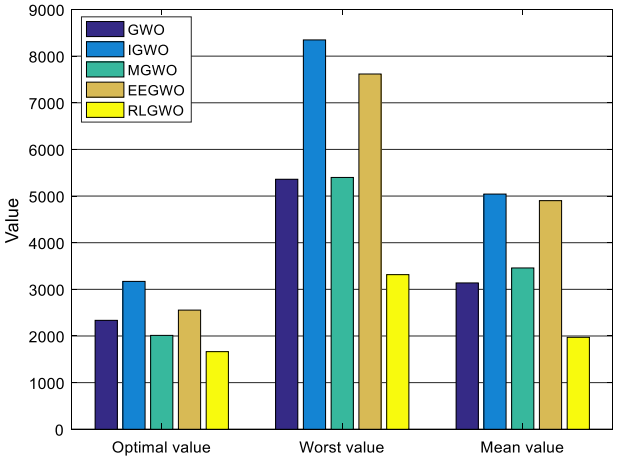


Fig. 17. The statistical results of the five algorithms in Case 3.

7. Conclusions

In this paper, a novel reinforcement learning based grey wolf optimizer algorithm called RLGWO has been presented for UAVs path planning problem in three-dimensional flight environment. Firstly, the path planning problem is transformed into the optimization problem to acquire the optimal path by minimizing the cost function. Secondly, considering that the proposed algorithm is designed to serve for UAVs path planning, four operations have been developed for each individual: exploration, exploitation, geometric adjustment, and optimal adjustment. According to the accumulated performance controlled by the reinforcement learning, the individual switches operation adaptively. The geometric adjustment and optimal adjustment operations are developed to solve the problem of trapping in local optimization and unsmooth for UAVs path planning. Meanwhile, the cubic B-spline curve is used to smooth the generated flight route. Finally, the performance of the proposed algorithm has been tested with the GWO, IGWO, MGWO and EEGWO algorithms to generate an optimal path for UAVs in three-dimensional space. The effectiveness and superiority of these algorithms are compared based on the statistical results and convergence cures. The comparative results show that the RLGWO algorithm is superior to other four algorithms, and the proposed algorithm can solve the three-dimensional UAVs path planning problem successfully.

In the area of meta-heuristic optimization algorithms, the No Free Lunch (NFL) theorem was proposed that a universally applicable optimal algorithm independent of the specific application is not exist. It means that if one algorithm can solve a kind of problem effectively, then it will not be effective to solve another kind of problem. Thus, due to the addition of geometric adjustment operation and optimal adjustment operation, the proposed algorithm is more effective in solving path planning problems of intelligent agents (such as robots, unmanned aerial vehicles, underwater vehicles, etc.), which is also the limitation of this method. We will further solve and improve this problem in the future work.

Author contribution

Chengzhi Qu carried out the investigation and wrote the original draft. Wendong Gai proposed the methodology and finished review. Maiying Zhong and Jing Zhang finished the validation.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2020.106099.

References

[1] D.F. Zhang, H.B. Duan, Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning, Neurocomputing 313 (2018) 229–246.
[2] P.B. Sujit, S. Saripalli, J.B. Sousa, Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless, IEEE Control Syst. 34 (1) (2014) 42–59.
[3] C.H. Yang, H.M. Tsai, C.S. Kang, et al., UAV path planning method for digital terrain model reconstruction - A debris fan example, Autom. Constr. 93 (2018) 214–230.
[4] W. Liu, Z. Zheng, K.Y. Cai, Bi-level programming based real-time path planning for unmanned aerial vehicles, Knowl.-Based Syst. 44 (2013) 34–47.
[5] Y. Lin, S. Saripalli, Sampling-based path planning for UAV collision avoidance, IEEE Trans. Intell. Transp. Syst. 18 (11) (2017) 3179–3192.
[6] I.S. Alshawi, L. Yan, W. Pan, et al., Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm, IEEE Sens. J. 12 (10) (2012) 3010–3018.
[7] Y.B. Chen, G.C. Luo, Y.S. Mei, et al., UAV path planning using artificial potential field method updated by optimal control theory, Internat. J. Systems Sci. 47 (6) (2016) 1407–1420.
[8] M. Radmanesh, M. Kumar, Flight formation of UAVs in presence of moving obstacles using fast-dynamic mixed integer linear programming, Aerosp. Sci. Technol. 50 (2016) 149–160.
[9] M. Kothari, I. Postlethwaite, A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees, J. Intell. Robot. Syst. 71 (2) (2013) 231–253.
[10] D. Manjarres, I. Landatorres, S. Gillopez, et al., A survey on applications of the harmony search algorithm, Eng. Appl. Artif. Intell. 26 (8) (2013) 1818–1831.
[11] W.A. Guo, M. Chen, L. Wang, et al., A survey of biogeography-based optimization, Neural Comput. Appl. 28 (8) (2016) 1909–1926.
[12] M.T. Younis, S. Yang, Hybrid meta-heuristic algorithms for independent job scheduling in grid computing, Appl. Soft Comput. 72 (2018) 498–517.
[13] Y. Zhao, Z. Zheng, Y. Liu, Survey on computational-intelligence-based UAV path planning, Knowl.-Based Syst. 158 (2018) 54–64.
[14] G.G. Wang, H.E. Chu, S. Mirjalili, Three-dimensional path planning for UCAV using an improved bat algorithm, Aerosp. Sci. Technol. 49 (2016) 231–238.
[15] P. Kumar, S. Garg, A. Singh, et al., MVO-based 2D path planning scheme for providing quality of service in UAV environment, IEEE Internet Things J. 5 (3) (2018) 1698–1707.
[16] M.D. Phung, H.Q. Cong, T.H. Dinh, et al., Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection, Autom. Constr. 81 (2017) 25–33.
[17] J. Wu, H. Wang, N. Li, et al., Path planning for solar-powered UAV in urban environment, Neurocomputing 275 (2017) 2055–2065.
[18] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (3) (2014) 46–61.
[19] E. Daniel, J. Anitha, J. Gnanaraj, Optimum laplacian wavelet mask based medical image using hybrid cuckoo search – grey wolf optimization algorithm, Knowl.-Based Syst. 131 (2017) 58–69.
[20] D. Guha, P.K. Roy, S. Banerjee, Load frequency control of interconnected power system using grey wolf optimization, Swarm Evol. Comput. 27 (2015) 97–115.
[21] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary gray wolf optimization approaches for feature selection, Neurocomputing 172 (2015) 371–381.
[22] M. Radmanesh, M. Kumar, M. Sarim, Grey wolf optimization based sense and avoid algorithm in a Bayesian framework for multiple UAV path planning in an uncertain environment, Aerosp. Sci. Technol. 77 (2018) 168–179.
[23] W. Long, J. Jiao, X. Liang, et al., Inspired grey wolf optimizer for solving large-scale function optimization problems, Appl. Math. Model. 60 (2018) 112–126.
[24] V. Kumar, D. Kumar, An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems, Adv. Eng. Softw. 112 (2017) 231–254.
[25] W. Long, J. Jiao, X. Liang, et al., An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization, Eng. Appl. Artif. Intell. 68 (2018) 63–80.
[26] J. Kober, J.A. Bagnell, J. Peters, Reinforcement learning in robotics: A survey, Int. J. Robot. Res. 32 (11) (2013) 1238–1274.
[27] I. Grondman, L. Busoniu, G. Lopes, et al., A survey of actor-critic reinforcement learning: Standard and natural policy gradients, IEEE Trans. Syst. Man Cybern. C 42 (6) (2012) 1291–1307.
[28] B. Kiumarsi, K.G. Vamvoudakis, H. Modares, et al., Optimal and autonomous control using reinforcement learning: A survey, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2018) 2042–2062.
[29] R.S. Sutton, Learning to predict by the methods of temporal differences, Mach. Learn. 3 (1) (1988) 9–14.
[30] A. Konar, I.G. Chakraborty, S.J. Singh, et al., A deterministic improved Q-learning for path planning of a mobile robot, IEEE Trans. Syst. Man Cybern.: Syst. 43 (5) (2013) 1141–1153.
[31] M.G. Wei, S. Wang, F.J. Zheng, et al., UGV navigation optimization aided by reinforcement learning-based path tracking, IEEE Access 6 (2018) 57814–57825.
[32] D.L. Cruz, W. Yu, Path planning of multi-agent systems in unknown environment with neural kernel smoothing and reinforcement learning, Neurocomputing 233 (2017) 34–42.
[33] I. Carlucho, M.D. Paula, S. Wang, et al., Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning, Robot. Auton. Syst. 107 (2018) 71–86.
[34] Y. Chen, J. Yu, Y. Mei, et al., Modified central force optimization (MCFO) algorithm for 3D UAV path planning, Neurocomputing 171 (2016) 878–888.
[35] Z.H. Zhan, J. Zhang, Y. Li, et al., Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybern. 39 (6) (2009) 1362–1381.
[36] X.Y. Zhang, Y.X. Lu, M.S. Jia, et al., A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning, Appl. Soft Comput. 70 (2018) 371–388.
[37] Y.B. Chen, Y.S. Mei, J.Q. Yu, et al., Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm, Neurocomputing 266 (2017) 445–457.