EURASIP Journal on Advances
in Signal Processing

RESEARCH **Open Access**

Dual-game based UAV swarm obstacle avoidance algorithm in multi-narrow type obstacle scenarios

Ye Lin[1], Zhenyu Na[1,2*], Zilong Feng[1], Bin Lin[1] and Yun Lin[3]

*Correspondence:
nazhenyu@dlmu.edu.cn

[1] School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
[2] Guangdong Provincial Key Laboratory of Aerospace Communication and Networking Technology, Shenzhen 518052, China
[3] School of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Abstract

Due to the advantages of rapid deployment, flexible response and strong invulnerability, unmanned aerial vehicle (UAV) swarm has been widely applied in collaborative warfare and emergency communication. However, UAV swarm in complex environments is prone to chaotic collapse due to obstructions. A UAV swarm obstacle avoidance system model for multi-narrow type obstacles is established. Due to the fact that only one UAV is allowed to pass through each small hole at any given moment, addressing the issue of congestion caused by swarming effects becomes crucial in addition to managing the competitive allocation of multiple UAVs to multiple holes. Aiming at this problem, a dual-game real-time obstacle avoidance scheme is proposed for UAV swarm with multi-narrow type obstacle scenarios, which divides the flight process of the UAV swarm into two stages: maintaining the flight state of the UAV swarm unchanged when no obstacles are encountered, and implementing matching separation and motion state switching by means of dual-game strategy when facing multi-narrow type obstacles, ultimately achieving orderly passage after multiple rounds of games. For the proposed scheme, a dual-game based Flocking (DGF) obstacle avoidance algorithm is proposed. Specifically, the motion state of each UAV obtained from the game is parameterized and integrated with the Flocking algorithm to calculate the motion control input for each UAV. The solution is iteratively obtained until the UAV swarm completes the obstacle avoidance. Simulation results demonstrate that the proposed DGF algorithm not only enables smooth obstacle avoidance for the UAV swarm in multi-narrow type obstacle scenarios, but also effectively resolves the internal chaos problem in the UAV swarm, thereby preventing rigid collisions.

Keywords: UAV swarm, Multi-narrow type obstacles, Dual-game, Flocking obstacle avoidance algorithm

1 Introduction

With the rapid development of current wireless communication technology and the increasing intelligence of mobile devices, the application of unmanned aerial vehicle (UAV) has evolved from single UAV to UAV swarm [1–4]. Compared with the single UAV system, UAV swarm has the advantages of decentralization and high environmental adaptability [5, 6]. Additionally, UAV swarm can fully leverage inter-UAV collaboration and efficiently accomplish tasks that cannot be completed solely by UAV quantity accumulation [7–9]. However, due to the complexity of the environment, UAV swarm inevitably encounters obstacles during task execution, so formation avoidance has become the hot topic in the field of UAV swarm in recent years [10–12].

The flight environment of UAV swarm under multi-level task requirements is highly complex, often encountering threats from various types of obstacles, which brings significant uncertainty to the flight safety of UAV swarm [13–16]. Currently, the commonly used obstacle avoidance strategies for UAV swarm include leader-follower, virtual structure and artificial potential field methods [17–19]. Mata-Machuca et al. [17] simulated the control performance of a leader-follower approach for differential configuration two-wheeled robots. Under any conditions, the follower is able to estimate the trajectory at the next instant based on the information received from the leader and then flexibly avoid obstacles. However, the leader-follower approach has poor robustness because the entire swarm will be completely paralyzed when the master node fails. Tan et al. [18] proposed the virtual structure method, which forms a rigid body composed of a central leader and surrounding UAVs to achieve formation maintenance. Although the virtual structure method improves the weak robustness of the leader-follower approach, the skyrocketing communication and computation requirements make it unsuitable for large-scale swarm formations. Tian et al. [19] introduced an overall configuration planning method based on an improved artificial potential field approach, which avoids complex kinematic calculations and exhibits good environmental adaptability. Although the artificial potential field approach has advantages in smooth control, it easily falls into local optima and leads to deadlock. It can be revealed that traditional UAV swarm obstacle avoidance methods have reached relative maturity, but their deficiencies in flexibility, robustness and computational complexity make them inadequate for meeting the obstacle avoidance requirements of large-scale UAV swarm.

Swarm intelligence-based UAV swarm obstacle avoidance has excellent performance in environmental adaptability and system self-healing ability and has been widely used to solve the obstacle avoidance problem of large-scale complex obstacles in recent years [20–22]. The earliest research on swarm intelligence for obstacle avoidance was conducted by Olfati-Saber, who proposed the distributed Flocking obstacle avoidance algorithm based on three principles observed in biological swarms: separation, cohesion and alignment [23]. The main idea is to introduce virtual agents on obstacles and utilize the separation rule among agents to achieve obstacle avoidance. As the Flocking model has been gradually improved, numerous scholars have proposed Flocking-based obstacle avoidance algorithms, aiming to adapt to various scenarios by adjusting parameters or introducing additional control variables [24–26]. Hu et al. [24] extended the two-dimensional (2D) Flocking obstacle avoidance to the 3D curved surface, enabling the application of the Flocking algorithm on a three-dimensional conical surface. Huang et al. [25] proposed a novel Flocking algorithm based on decentralized model predictive control, which ensures all agents to form a stable α-lattice, ultimately enabling safe passage through obstacle areas. Wei et al. [26] proposed a Flocking algorithm with multiple virtual leaders and validated its stability using the Lyapunov stability theorem and LaSalle's invariance principle.

It can be summarized that the current works on the Flocking-based obstacle avoidance algorithms have primarily focused on convex obstacles. However, the types of obstacles in practical scenarios are highly complex. Therefore, it is necessary to redesign obstacle avoidance algorithms for non-convex obstacles to accord with complex scenarios [27–29]. Chang et al. [27] proposed a rotational force that allows the agents to rotate their direction to avoid concave obstacles. Based on Bersani's concave obstacle avoidance model, Olcay et al. [28] designed an additional term to generate a rotational force field and guided the agents to move along the inner wall of the concave obstacles by local information exchange. Mikhail et al. [29] introduced a swarm control strategy based on Aristotelian mechanics, ensuring that the swarm can bypass obstacles without collisions.

However, non-convex obstacles not only include concave obstacles but also encompass more extreme narrow obstacles, such as small holes. To address the issue of blockage faced by UAV swarm in front of narrow type obstacles, several methods have been proposed, including formation, multi-UAV path planning and control-based methods [30–35]. Xu et al. [30] proposed a biomimetic formation scheme based on the characteristics of biomimetic transformation to guide the agents as a collective entity in a distributed manner, enabling them to pass through narrow type obstacles. However, the scalability and adaptability of this method are limited since each UAV in the formation needs to maintain a predefined posture and perform transformation operations. Zhou et al. [32] introduced a lightweight topology-based trajectory generation method and utilized an unreliable trajectory sharing network to generate safe, smooth and dynamically feasible trajectories within milliseconds to avoid narrow type obstacles. However, with the expansion of UAV swarm, trajectory planning may become infeasible due to the increasing computational complexity and communication load. Since the control-based methods can directly guide the motion of UAVs based on global paths and local information, they are more suitable for the large-scale UAV swarm. Gao et al. [34] proposed a distributed vector field controller to guide the UAV swarm within a trapezoidal virtual tube through a narrow opening. However, the aforementioned works only consider single narrow type obstacle scenarios. We note that the above works on narrow type obstacle avoidance either lack scalability or do not adapt to the large-scale UAV swarm. Moreover, most of the existing methods only focus on single narrow type obstacle. In multi-narrow type obstacle scenarios, UAV swarm is prone to blockage due to the swarming effect and competitive allocation. Therefore, a real-time decision scheme is needed to determine the motion states of UAVs at each instant. Fortunately, game theory [36–38] provides a new perspective for obstacle avoidance algorithms in multi-narrow type obstacle scenarios, because it allows for state prediction and payoff balance. Cen et al. [36] proposed Nash equilibrium model and cooperative game model to find better paths by the cooperation and competition between the players in the game. Mohammad et al. [37] introduced a game theory-based bio-inspired distributed intelligent control method that exhibits good learning capability and low computational complexity.

In summary, the current works on UAV swarm obstacle avoidance primarily focuses on convex obstacles, with a lack of investigation on narrow type obstacle avoidance in complex environments. Furthermore, future tasks will impose stricter requirements on timeliness, which necessitates UAV swarm to perform ordered obstacle avoidance while maintaining consistent motion. In light of these considerations, this paper proposes a dual-game based Flocking (DGF) algorithm to efficiently perform UAV swarm obstacle avoidance in multi-narrow type obstacle scenarios. The main contributions of this paper are summarized as follows:

- A UAV swarm obstacle avoidance system model for multi-narrow type obstacles is established. The system consists of the UAV swarm and multiple small-hole type obstacles. Due to the extremely narrow aperture of each hole, only one UAV can pass through at a time. Therefore, the system not only considers the competitive allocation of multiple UAVs to multiple small holes, but also addresses the blockage problem caused by the swarming effect.
- A dual-game real-time obstacle avoidance scheme is designed. When the UAV swarm reaches the decision distance, the first-level game is initiated to divide the UAV swarm into two sub-swarms. This division aims to prevent one small hole from being congested by a large number of UAVs while leaving the other hole unused. Upon reaching the front of small holes, the sub-swarms trigger the second-level game to compete for the priority to pass through that hole, thereby achieving smooth obstacle avoidance for the UAV swarm.
- An iterative algorithm is proposed to implement the above scheme. The algorithm assigns different motion states based on corresponding game strategies. The UAV swarm is endowed with two distinct motion states by the first-level game. As the algorithm iterates, the sub-swarms gradually gather in front of small holes and update the motion states of UAVs by the second-level game. This process continues until all UAVs pass through the small holes in an orderly manner, achieving efficient obstacle avoidance for the UAV swarm.

The rest of the paper is organized as follows. Section 2 presents the proposed system model. Section 3 introduces the principles of the Flocking obstacle avoidance algorithm. The proposed DGF algorithm is detailed in Sect. 4. A comparison of simulations is conducted in Sect. 5, and our concluding remarks are drawn in Sect. 6.

2 System model

In this section, we first model the small-hole type obstacle as shown in Fig. 1, where R_2 represents the maximum width of the UAV, R_1 represents the diameter of the small hole, and we assume $2R_2 > R_1 > R_2$. Therefore, at any given instant, only one UAV is allowed to pass through each small hole.

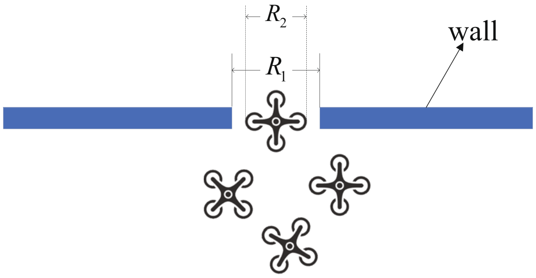


Fig. 1 Modeling of small-hole type obstacle

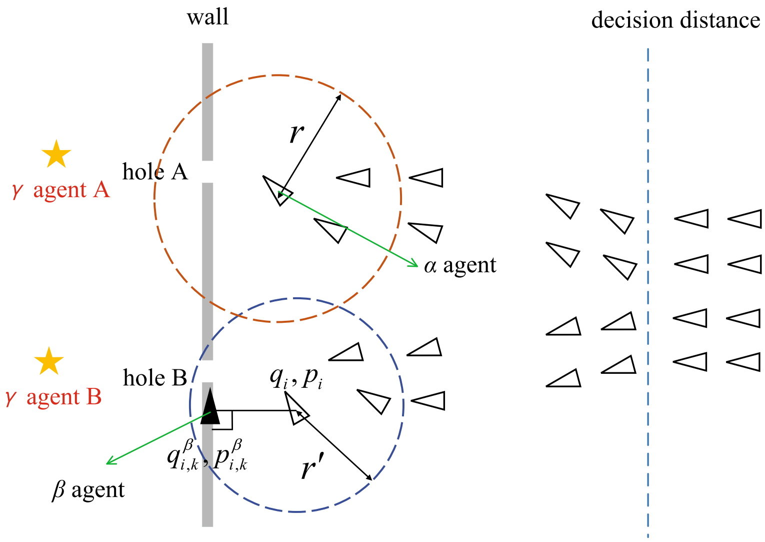


Fig. 2 UAV swarm obstacle avoidance system model for multi-narrow type obstacle

Table 1 Notational conventions

Notation	Description
N	Number of UAVs
q_i	Position of UAV i
p_i	Speed of UAV i
$q_{i,k}^{\beta}$	Position of obstacle k formed by the projection of UAV i
$p_{i,k}^{\beta}$	Speed of obstacle k formed by the projection of UAV i
N_i^{α}	Neighboring set of UAV i
r	Interaction range of the UAV
r'	Interaction range of the UAV to obstacles

As shown in Fig. 2, this paper focuses on a multi-narrow type obstacle avoidance system for the UAV swarm consisting of N UAVs and two holes. The UAV swarm gathers around hole A, and each UAV is represented as an α agent with position and speed denoted as q_i and p_i, respectively. The interaction range between α agents is r. All α agents within a circle with agent i as the center and r as the radius form the neighboring set N_i^{α}. The UAV swarm needs to move toward target points, so virtual γ agents are introduced to attract the α agents. During the flight to the target, the α agents project on the surface of the wall to form virtual β agents, which use the separation rule of the swarm to avoid collisions. $q_{i,k}^{\beta}$ and $p_{i,k}^{\beta}$ represent the position and speed of the β agents, respectively, and the interaction range between α agents and β agents is r'. Upon reaching the decision distance, the swarm initiates the first-level game to achieve dispersion, mitigating the congestion in front of one small hole while leaving the other hole underutilized. Subsequently, when the sub-swarms approach the front of the small holes, they trigger the second-level game to compete for the privileged passage through the small holes. The main notations used in this paper are listed in Table 1.

3 Overview of classic flocking algorithm

3.1 Flocking algorithm principle

The dynamics of a group of N agents are given by the following system of differential equations

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i \end{cases} \quad q_i, p_i, u_i \in \mathbb{R}^m \tag{1}$$

where u_i is the control input.

Each agent possesses an individual communication range within which other agents are considered as neighbors. The agent i can communicate only with its neighbors, the set of neighboring agents of agent i at instant t is defined as

$$N_i^\alpha = \left\{ j \in v_\alpha : \left\| q_j - q_i \right\| < r \right\} \tag{2}$$

where $\|\cdot\|$ represents the Euclidean norm of the vector, and v_α represents the set of α agents.

The geometry is considered to be in a stable state when each agent keeps a distance d from its neighboring agents. Hence, we define a mathematical boundary condition as follows:

$$\left\| q_j - q_i \right\| < d \quad \forall j \in N_i^\alpha \tag{3}$$

At this point, the lengths of the edges in the network formed by the agents are all equal, and this structure is referred to as an α-lattice. However, the network structure formed by the UAV swarm in this paper does not conform to a standard α-lattice configuration in most cases. Instead, a quasi α-lattice is considered, which satisfies the following inequality constraints

$$-\delta < \left\| q_j - q_i \right\| - d < \delta \tag{4}$$

where δ represents the uncertainty factor.

We create the deviation energy as a function of the distance between two agents. In order to ensure the differentiability of the function at $z = 0$, a mapping called σ-norm is introduced as follows:

$$\|z\|_\sigma = \frac{1}{\xi}\left[\sqrt{1 + \xi\,\|z\|^2} - 1\right] \tag{5}$$

where $\xi > 0$. In order to have a smooth potential function, the following bump function is defined

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h) \\ \frac{1}{2}\left[1 + \cos\left(\pi \frac{(z-h)}{(1-h)}\right)\right], & z \in [h, 1) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $h \in (0, 1)$. The bump function can create the adjacency matrix $A[q] = \left[a_{ij}(q)\right]$, whose element $a_{ij}(q)$ can be expressed as

$$a_{ij}(q) = \rho_h\left(\frac{\left\|q_j - q_i\right\|_\sigma}{r_\alpha}\right) \in [0,1], j \neq i \tag{7}$$

where $r_\alpha = \|r\|_\sigma$. If the distance between agent i and j is greater than r_α, then $\left\|a_{ij}\right\| = 0$.

For an obstacle with a hyperplane boundary that has a unit normal $\mathbf{a}_k$ and passes through the point y_k, the position and velocity of the β agent are determined by

$$\hat{q}_{i,k} = Pq_i + (I + P)y_k \tag{8}$$

$$\hat{p}_{i,k} = Pp_i \tag{9}$$

where $P = I - \mathbf{a}_k\mathbf{a}_k^{\mathrm{T}}$ is a projection matrix. If the velocity direction of the β agent is parallel to the surface of the obstacle, then $\mathbf{a}_k^{\mathrm{T}}\hat{q}_{i,k} = 0$.

$u_i^\alpha, u_i^\beta, u_i^\gamma$ are generated in the Flocking cooperative control algorithm so that the expression for the control quantity can be calculated for each of the α agents as follows:

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma \tag{10}$$

where u_i^α denotes the rules of movement between α agents that satisfy the three basic rules of aggregation behavior proposed by Reynolds: separation, alignment and speed cohesion. u_i^α can be expressed as

$$u_i^\alpha = c_q^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha\left(\left\|q_j - q_i\right\|_\sigma\right)\mathbf{n}_{i,j} + c_p^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)\left(p_j - p_i\right) \tag{11}$$

where c_q^α and c_p^α are positive constant parameters. $\mathbf{n}_{i,j}$ is defined as

$$\mathbf{n}_{i,j} = \frac{q_j - q_i}{\sqrt{1 + \xi\left\|q_j - q_i\right\|^2}} \tag{12}$$

The potential functions needed for the gradient-based term are defined as

$$\phi_\alpha(z) = \rho_h\left(\frac{z}{r_\alpha}\right)\phi(z - d_\alpha) \tag{13}$$

$$\phi(z) = \frac{1}{2}[(a + b)\sigma_1(z + c) + (a - b)] \tag{14}$$

where $\sigma_1 = \frac{z}{\sqrt{1+z^2}}$ and $0 < a \leq b, c = \frac{|a-b|}{\sqrt{4ab}}$, which guarantee $\phi(0) = 0$.

u_i^β denotes the repulsive force relationship between α agents and virtual obstacle β agents, which can be expressed as

$$u_i^\beta = c_q^\beta \sum_{k \in N_i^\beta} \phi_\beta\left(\left\|\hat{q}_{i,k} - q_i\right\|_\sigma\right)\mathbf{n}_{i,j}^\beta + c_p^\beta \sum_{k \in N_i^\beta} b_{i,k}(q)\left(\hat{p}_{i,k} - p_i\right) \tag{15}$$

where c_q^β and c_p^β are positive constant parameters. $\mathbf{n}_{i,j}^\beta$ is defined as

$$\mathbf{n}_{i,j}^{\beta} = \frac{\hat{q}_{i,k} - q_i}{\sqrt{1 + \xi \left\| \hat{q}_{i,k} - q_i \right\|^2}} \tag{16}$$

The individuals in the swarm have to move toward the target point, so the virtual target is introduced which is called the γ agent. u_i^{γ} denotes the attraction relationship between α and γ agents, which can be expressed as

$$u_i^{\gamma} = -c_q^{\gamma} \sigma_1 (q_i - q_r) - c_p^{\gamma} (p_i - p_r) \tag{17}$$

where $\sigma_1 = \frac{z}{\sqrt{1+z^2}}$, c_q^{γ} and c_p^{γ} are positive constant parameters.

3.2 Flocking algorithm steps

In summary, the Flocking obstacle avoidance algorithm is shown in Algorithm 1. First, if the distance between agent i and its neighboring agents is less than r, the neighboring set N_i^{α} is formed, and then the spatial adjacency matrix element $a_{ij}(q)$ is obtained. Secondly, the movement rules between α agents are obtained based on the spatial adjacency matrix. A virtual β agent is projected onto the obstacle when agent i encounters the obstacle. If the distance between agent i and the obstacle is less than r', the repulsion relationship u_i^{β} between α agent and the virtual β agent is calculated. Next, the acting force u_i^{γ} between α agent and the target agent γ is calculated. Finally, u_i, q_i and p_i are obtained, respectively.

```
Input: $q_i^{(0)}$, $p_i$, $N$
for $i = 1$ to $N$ do
    if $\|q_j - q_i\| < r$ then
        Calculate $a_{ij}(q)$ according to (7)
        Calculate $u_i^{\alpha}$ according to (12)
    end
    for all $k$ do
        Calculate $\hat{q}_{i,k}$ according to (11)
        if $\|q_j - q_i\| < r'$ then
            Calculate $u_i^{\beta}$ according to (8)
        end
     end
    Calculate $u_i^{\gamma}$ according to (17)
        $u_i \leftarrow u_i + u_i^{\alpha} + u_i^{\beta} + u_i^{\gamma}$
        $q_i \leftarrow q_i + v_i \Delta t + 0.5 u_i \Delta t^2$
        $p_i \leftarrow p_i + p_i \Delta t$
```

Algorithm 1 Flocking obstacle avoidance algorithm.

Table 2 Game matrix when UAV *i* is closer to hole A than UAV *j*

	UAV *j* flying to hole A	**UAV *j* flying to hole B**
UAV *i* flying to hole A	(0,−2)	(0,0)
UAV *i* flying to hole B	(−2,−2)	(−2,−1)

Table 3 Game matrix when UAV *i* is closer to hole B than UAV *j*

	UAV *j* flying to hole A	**UAV *j* flying to hole B**
UAV *i* flying to hole A	(−2,−1)	(−2,−2)
UAV *i* flying to hole B	(0,0)	(0,−2)

4 Dual-game based flocking obstacle avoidance algorithm

4.1 Dual-game real-time obstacle avoidance scheme

The proposed dual-game real-time obstacle avoidance scheme aims to maintain the integrity of the swarm structure and divides the formation flight process into two stages. First, the motion state of the UAV swarm remains unchanged when no obstacle is encountered. Then, when facing multiple small holes, the UAV swarm utilizes the dual-game to determine the next motion state, ensuring successful obstacle avoidance.

At the decision distance, the UAV swarm triggers the first-level game. Firstly, the swarm obtains the relative position information of the two obstacles via a fully connected network. Then, the two UAVs closest to the obstacles are selected as participants in the game. During the game, the payoffs for passing through the obstacles are generated, and the game matrix for the UAV swarm is obtained. Subsequently, according to the Nash equilibrium derived from the game matrix, the motion states of each UAV are adjusted, and then, the UAV gets the attraction from different targets. The purpose of this game is to disperse the UAV swarm, thereby avoiding the situation where one hole is congested with a large number of UAVs while the other hole remains idle. Assuming that UAV $i(i=1,2,\cdots,N)$ is closer to hole A than UAV $\left(j=1,2,\cdots N;j\neq i\right)$ when reaching the decision distance, the game matrix for UAV i is shown in Table 2.

From Table 2, it can be observed that when UAV i reaches the decision distance and is closer to hole A than UAV *j*, the Nash equilibrium is achieved if UAV i flies toward hole A and UAV *j* flies toward hole B. Similarly, when UAV i is closer to hole B than UAV *j* upon reaching the decision distance, the corresponding game matrix is shown in Table 3.

From Table 3, it can be observed that when UAV i reaches the decision distance and is closer to hole B than UAV *j*, the Nash equilibrium is achieved if UAV i flies toward hole B and UAV *j* flies toward hole A. Therefore, after multiple rounds of the game at the decision distance, the UAV swarm can ultimately disperse in front of the two holes and prepare for the next level of the game.

Table 4 Game matrix when UAV *i* is closest to the hole

	UAV *j* passes	**UAV *j* waits**
UAV *i* passes	(−2,−2)	(1,0)
UAV *i* waits	(−1,−1)	(0,0)

Table 5 Game matrix when UAV *i* is closest to the hole

	UAV *j* passes	**UAV *j* waits**
UAV *i* passes	(−2,−2)	(−1,−1)
UAV *i* waits	(0,1)	(0,0)

To avoid simultaneous gathering of the UAV swarm toward a single small hole, a matching separation strategy is employed when reaching the decision distance based on the relative distances between individual UAVs and obstacles. By establishing an α -lattice structure, the UAV swarm forms a fully connected network, enabling information exchange on position and speed among UAVs. Consequently, the first two UAVs reaching the decision distance gain the attraction from different targets by the first-level game. By multiple rounds of the game, the UAV swarm gradually converges toward the stable Nash equilibrium state, ultimately forming two cohesive and stable sub-swarms.

After the first-level game, the UAV swarm is dispersed around the two small holes. In special cases, the UAV swarm will pass through the small holes orderly in a straight line, but in general, the UAV swarm will be blocked in front of the small holes and cannot pass through them. At this point, each UAV is not only influenced by the attraction from its target but also affected by the repulsion from other UAVs, thus causing the UAVs to block each other from moving forward. To avoid this problem, the second-level game is introduced into the obstacle avoidance strategy, where UAVs need to compete for the right of passage through the respective small holes.

Specifically, assuming that UAV i is closest to small hole $x\left(x \in \left\{\mathrm{A}, \mathrm{B}\right\}\right)$ at this instant, the game matrix with UAV j is shown in Table 4.

From Table 4, it can be observed that when UAV i passes through the small hole while UAV j waits for the next round of the game, it reaches the Nash equilibrium state. Furthermore, considering the practical context, it is reasonable for UAV i to pass through the small hole in this scenario. Similarly, if UAV j is the closest to the small hole x $\left(x \in \left\{\mathrm{A}, \mathrm{B}\right\}\right)$ at this instant, the corresponding payoff matrix between UAV i and UAV j is shown in Table 5.

From Table 5, it can be observed that when UAV j passes through the hole while UAV i waits, it reaches the Nash equilibrium state. Therefore, after multiple rounds of the second-level game, all UAVs can converge to the Nash equilibrium state, resulting in a sequential passage through small holes from the nearest to the farthest UAV.

For the UAV blocked in front of small holes that cannot advance, it is known from the above game matrix that it is inevitable that the UAV close to the small hole gets the priority to pass through the hole. At this point, the UAV close to the small hole gets the attraction from the target and then cuts off the interaction force with other UAVs and passes through the hole by changing the control input. Meanwhile, other UAVs lose the attraction from the target and wait for the next round of the game. After multiple rounds of the game, all UAVs can obtain the right to pass through the small holes and finally pass through the small holes in an orderly manner.

According to the above game results, UAVs have different motion states under each layer of the game. Therefore, it is necessary to define different motion state expressions for each layer of the game in order to obtain the actual speed and position information of each UAV. In the first-level game, where UAVs need to fly toward different targets, we define motion state S_1 as the attraction from the target corresponding to hole A, indicating that the UAV flies toward hole A. Similarly, motion state S_2 is defined as the attraction from the target corresponding to hole B, indicating that the UAV flies toward hole B. Therefore, UAV motion state expressions corresponding to different motion states are defined as follows:

$$S_1 : u_i = c_q^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha\left(\left\|q_j - q_i\right\|\right)\mathbf{n}_{i,j} + u_i^\beta + u_i^{\gamma_{\mathrm{A}}} \tag{18}$$

$$S_2 : u_i = c_q^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha\left(\left\|q_j - q_i\right\|\right)\mathbf{n}_{i,j} + u_i^\beta + u_i^{\gamma_{\mathrm{B}}} \tag{19}$$

Similarly, it is necessary to control the order of UAVs passing through the small holes in the second-level game. The UAV closest to the target obtains the attraction from the target and cuts off the interaction force with other UAVs. At this instant, we define the motion state of this UAV as S_3. UAVs that have not acquired the attraction from the target remain stationary, waiting for the next round of the game, and their motion state is defined as S_4. The motion state expressions are, respectively, defined as follows:

$$S_3 : u_i = c_q^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha\left(\left\|q_j - q_i\right\|\right)\mathbf{n}_{i,j} + u_i^\beta + u_i^{\gamma_x}, \left(x \in \left\{\mathrm{A}, \mathrm{B}\right\}\right) \tag{20}$$

$$S_4 : u_i = u_i^\alpha + u_i^\beta \tag{21}$$

```
first-level game:
  while $l < L$ do
    for $i = 1$ to $N$ do
      if $\|q_i - q_{\gamma_{\mathrm{A}}}\| < \|q_j - q_{\gamma_{\mathrm{A}}}\|$ && $\|q_j - q_{\gamma_{\mathrm{B}}}\| < \|q_i - q_{\gamma_{\mathrm{B}}}\|$ then
        $S_1 : u_i \leftarrow u_i + c_q^{\alpha} \sum_{j \in N_i^{\alpha}} \phi_{\alpha}\left(\|q_j - q_i\|_{\sigma}\right) \mathbf{n}_{i,j} + u_i^{\beta} + u_i^{\gamma_{\mathrm{A}}}$
        $S_2 : u_i \leftarrow u_i + c_q^{\alpha} \sum_{j \in N_i^{\alpha}} \phi_{\alpha}\left(\|q_j - q_i\|_{\sigma}\right) \mathbf{n}_{i,j} + u_i^{\beta} + u_i^{\gamma_{\mathrm{B}}}$
      end
    end
    $l = l + 1$
  end
second-level game:
  while $l < L$ do
    while $N \neq 0$ do
      for $i = 1$ to $N$ do
        $q_i = \min\left(\|q_i - q_{\gamma_{\mathrm{A}}}\|\right)$
      end
      for $i = 1$ to $N$ do
        $q_j = \min\left(\|q_j - q_{\gamma_{\mathrm{A}}}\|\right)$
      end
      if $\|q_i - q_{\gamma_x}\| < \|q_j - q_{\gamma_x}\|$ then
        $S_3 : u_i \leftarrow u_i + c_q^{\alpha} \sum_{j \in N_i^{\alpha}} \phi_{\alpha}\left(\|q_j - q_i\|_{\sigma}\right) \mathbf{n}_{i,j} + u_i^{\beta} + u_i^{\gamma_x}$
        $S_4 : u_i \leftarrow u_i + u_i^{\beta}$
      end
      $N = N - 1$
    end
    $l = l + 1$
  end
```

Algorithm 2 Dual-game real-time obstacle avoidance scheme flow

4.2 Dual-game based Flocking obstacle avoidance algorithm

Combined with Algorithm 1 and Algorithm 2, the proposed DGF algorithm is presented in Algorithm 3. First, according to Algorithm 1, the motion control inputs u_i^{α}, u_i^{β} and u_i^{γ} are calculated for each α agent. Then, in the first-level game, the UAV swarm is divided into two sub-swarms, each influenced by attraction from different targets and the motion states S_1 and S_2 are obtained. Subsequently, in the second-level game, the two UAVs that are closest to the target are selected for the game, and the motion states S_3

and S_4 are obtained. Finally, the actual motion parameters of the UAVs are derived from the motion states, which further determine the UAV trajectories.

```
Algorithm 1: Calculate $u_i^\alpha$, $u_i^\beta$, $u_i^\gamma$
    $u_i \leftarrow u_i + u_i^\alpha + u_i^\beta + u_i^\gamma$
    while $l < L$
        Algorithm 2:
            first-level game: Calculate $S_1$ and $S_2$ according to (18)-(19)
                $q_i \leftarrow S_1$ and $S_2$
            second-level game: Calculate $S_3$ and $S_4$ according to (20)-(21)
                $q_i \leftarrow S_3$ and $S_4$
    end
```

Algorithm 3 Dual-game based Flocking obstacle avoidance algorithm

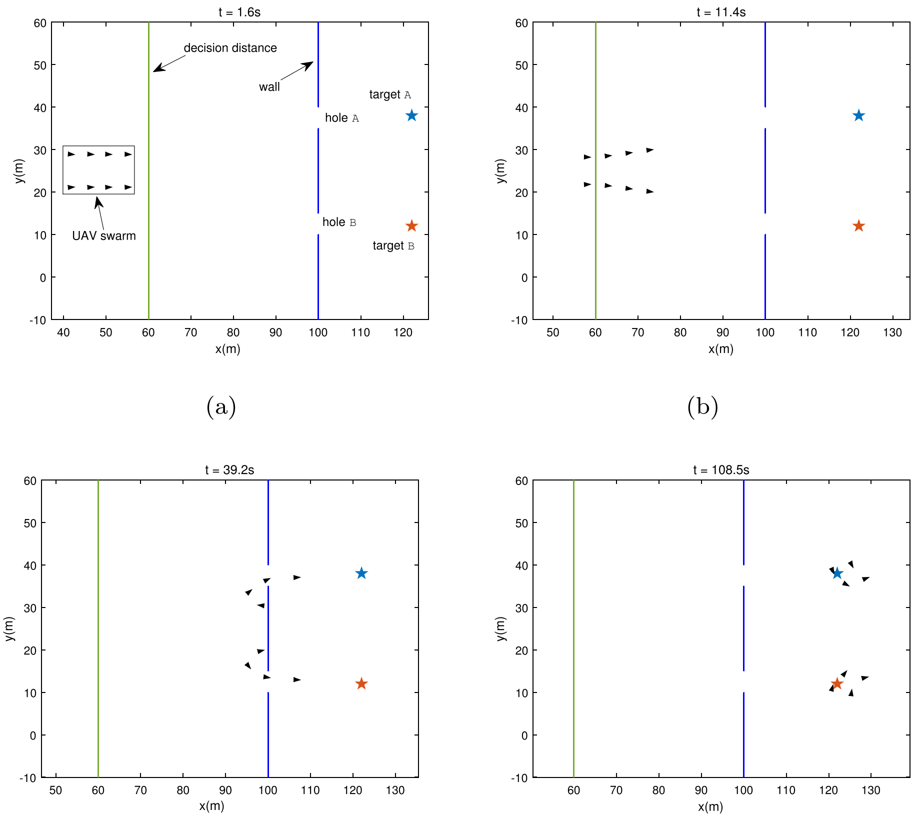
4.3 Complexity analysis

In this subsection, we will briefly analyze the computational complexity of the DGF algorithm. The entire algorithm is controlled by the number of iterations, which is denoted as l. Firstly, in the first-level game, the algorithm calculates the squared Euclidean distance matrix twice, which has a time complexity of $\mathcal{O}(N^2)$, where N represents the number of UAVs. Secondly, in the second-level game, the algorithm updates the motion states based on the interaction results of the agents, which has a time complexity of $\mathcal{O}(N^2)$. Therefore, the overall time complexity of the algorithm is $\mathcal{O}(l \times N^2)$.

5 Numerical results and discussion

In this section, we will present the numerical results of the UAV swarm passing through the small holes. The simulation parameters are set based on the classical Flocking algorithm [23]. We set the safe distance between the UAVs to $d = 5$ m and the interaction range between UAVs to $r = 6$ m. Similarly, the safe distance and interaction range between the UAVs and obstacles are set to $d' = 1.5$ m and $r' = 1.8$ m, respectively. To prevent collisions between UAVs, the interaction force coefficient need to satisfy $c_q^\alpha > c_q^\gamma$. Hence, we set $c_q^\alpha = 5$, $c_q^\gamma = 1$ and $c_q^\beta = 12$. To enhance the speed of passing through the small holes, we set the step $\delta_t = 0.1$ s and the number of iterations $l = 1600$. Additionally, in the σ mapping function, we set $\varepsilon = 0.1$, $h_\alpha = 0.2$, $h_\beta = 0.9$ and $a = b = 5$. In order to consider both algorithm performance and visualization simultaneously, we evaluate the algorithm based on both small-scale ($N = 8$) and large-scale ($N = 28$) UAV swarm scenarios.

Figure 3 presents the simulation results of the small-scale UAV swarm with $N = 8$ passing through the small holes. From Fig. 3a and b, it can be observed that the UAV swarm adopts an α-lattice structure and flies toward the targets before encountering the small holes. Upon reaching the decision distance, the swarm undergoes matching separation, with sub-swarms continuing to fly toward the two targets. As seen from Fig. 3c, the sub-swarms initiate a sequential and orderly passage as they approach the



(a) (b)

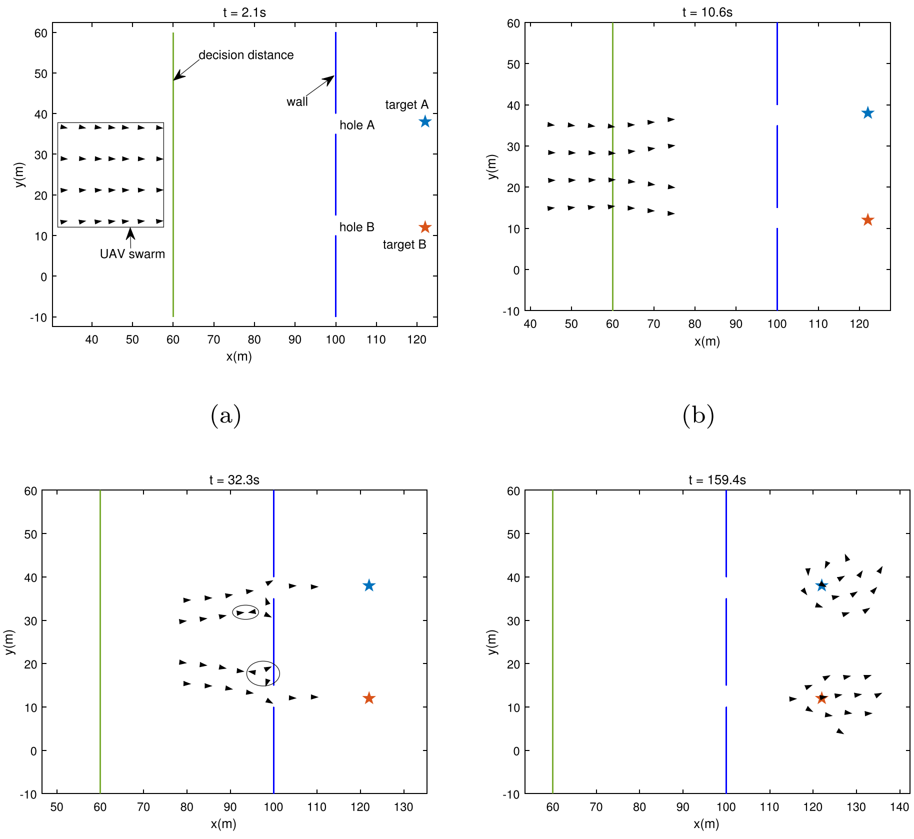
(c) (d)

Fig. 3 The perforation process of UAV swarm for $N = 8$

small holes. This is achieved by the dynamically adjusting UAVs' motion states based on the second-level game. The UAV closest to the small hole obtains the right to pass through it and maintains the motion state S_3, while the remaining UAVs switch to the motion state S_4, disconnect from the target attraction and wait for the next round of the game. From Fig. 3d, it is evident that the UAV swarm successfully pass through the small holes and reach target points after multiple iterations.

Figure 4 presents the simulation results of the large-scale UAV swarm with $N = 28$ passing through the small holes. Similar to the small-scale case, the UAV swarm steadily passes through the small holes over time. From Fig. 4c, it can be observed that the UAVs within the ellipse exhibit a tendency to move away from the targets in the opposite direction. This phenomenon is attributed to the fact that the UAVs are not only influenced by the attraction from the targets but also affected by repulsion from other neighboring UAVs. As a result, they adjust their acceleration in the opposite direction to maintain a safe distance from neighboring UAVs. From Fig. 4d, it is evident that the UAV swarm successfully pass through the small holes, ultimately reaching the target points. This illustrates the effectiveness and universality of the proposed DGF algorithm.

Figure 5 illustrates the relationship between the passage time of UAVs and the decision distance with $N = 8$. As seen from Fig. 5, the passage time of UAVs decreases as the decision distance increases, which indicates that UAV swarm can pass through



(a)

(b)

(c)

(d)

Fig. 4 The perforation process of UAV swarm for $N = 28$

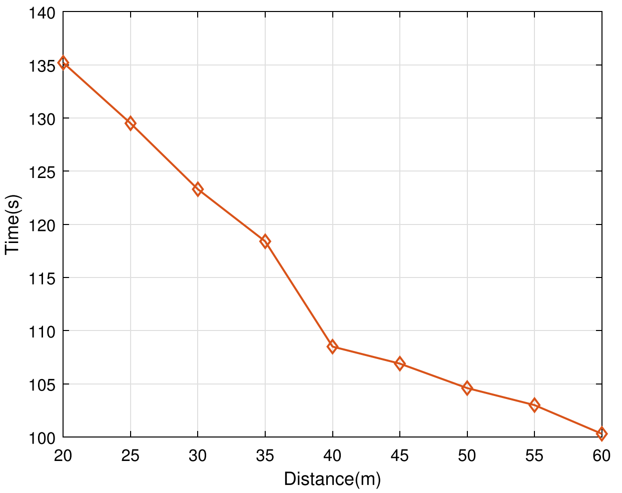
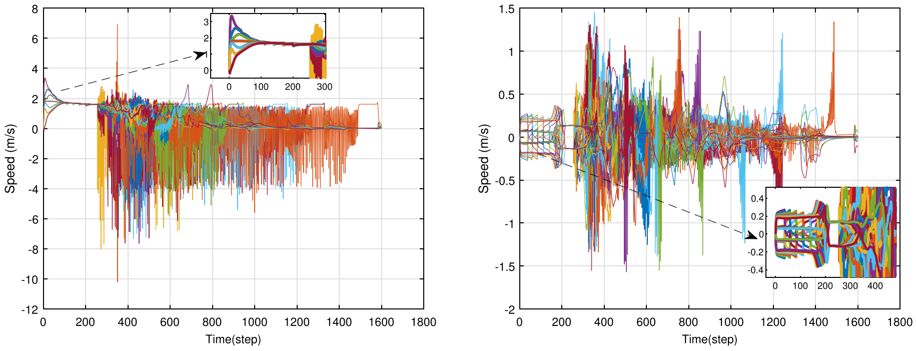


Fig. 5 Perforation time versus decision distance

the small holes more quickly by increasing the decision distance. However, a longer decision distance implies a stronger UAV perception capability. Notably, a significant change in passage time occurs when the decision distance is 40 m. Moreover, as the decision distances increase, the decreasing trend in passage time becomes less pronounced. Therefore, using 40 m as the decision distance can provide the UAV swarm with shorter passage time while imposing lower perception requirements. This is also why a 40-m decision distance is selected in the previous and following simulations.

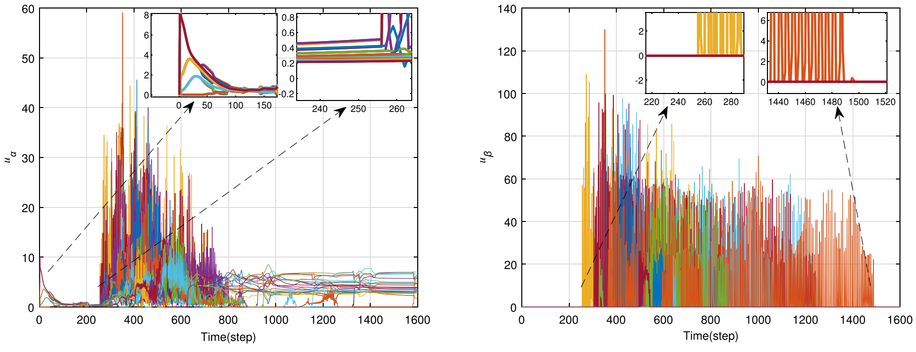
Figure 6 shows the speed components of each UAV in the x-axis and y-axis with $N = 28$. Firstly, from Fig. 6a, it can be observed that the speed component in the x-axis gradually increases from 0 m/s to 1.8m/s and stabilizes within the 0–250 time steps. This time period represents the convergence process where the UAV swarm self-organizes into an α-lattice structure and achieves convergence based on the speed consensus principle of the Flocking algorithm before encountering obstacles. Secondly, it can be seen from Fig. 6a and b that when the UAV swarm flies during the 250–1500 time steps, the speed components in both x-axis and y-axis oscillate remarkably. The reason for this phenomenon is that obstacles appear within the perception range of the UAVs during this time period, and the UAVs dynamically adjust their speed according to the positions of the obstacles and other UAVs in the neighboring set, thus avoiding collisions. Lastly, when all UAVs successfully pass through the small holes, it can be observed from Fig. 6a and b that the speed components in the x-axis and y-axis converge again and eventually approach 0. This demonstrates that the proposed DGF algorithm can dynamically adjust the motion states of individual UAVs according to the environment, ensuring the safe flight of the UAV swarm.

Figure 7 presents the variation curves of the motion control inputs of the UAV swarm with $N = 28$. Figure 7a shows the variation curve of the interaction term u^{α} between UAVs, while Fig. 7b shows the variation curve of the interaction term u^{β} between UAVs and obstacles. It can be observed from Fig. 7a that the value of u^{α} gradually increases from 0 and then decreases until it tends to 0 and remains stable. The reason for this phenomenon is that during this process, the UAV swarm transitions from the initial formation to the α-lattice structure. Once a stable formation is achieved, the potential energy function in the algorithm approaches 0 and remains stable without external forces acting on it. From Fig. 7a and b, it can be seen that the values of u^{α} and u^{β} start to increase abruptly around the time step of 255 and oscillation phenomenon appears. This is due to the fact that the obstacle starts to appear within the perception range of the UAV



(a) Speed component in the x-axis. (b) Speed component in the y-axis.

Fig. 6 Speed component with step



(a) u^α

(b) u^β

Fig. 7 UAV motion control input with step

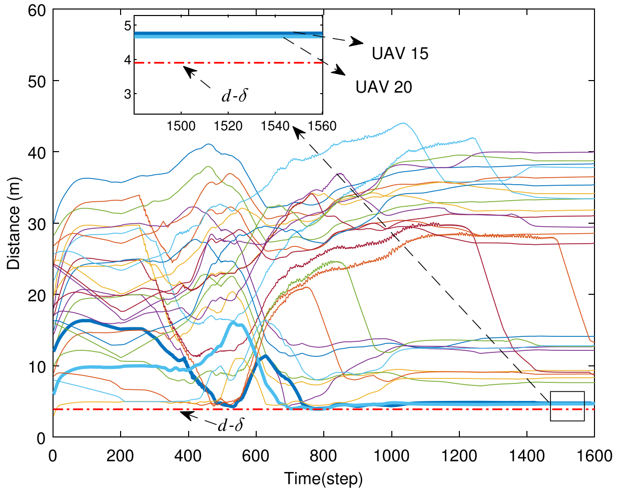


Fig. 8 Curve of distance change between UAV 28 and other UAVs

swarm at this instant, and each UAV starts to dynamically adjust the control input to move away from the surface of the obstacle due to the strong repulsion. An interesting phenomenon is observed in Fig. 7b, where after 1200 time steps, only one UAV exhibits repetitive oscillations in the u^β value, eventually converging to 0. This indicates that at this instant, only that UAV remains in front of the small hole and successfully passes through it after a certain period of time. Furthermore, from Fig. 7a, it can be seen that the value of u^α for the UAV swarm eventually stabilizes, indicating that the UAV swarm reestablishes a stable α-lattice structure. This further demonstrates the effectiveness of the proposed DGF algorithm.

Figure 8 shows the distance variations between UAV 28 and the other UAVs in the swarm with $N = 28$. It can be observed from Fig. 8 that the distance between UAV 28 and the other UAVs does not exactly equal the lattice spacing d. This is because the proposed DGF algorithm utilizes a quasi α-lattice structure, where there exists an elastic variable δ for the avoidance distance between UAVs. Furthermore, from Fig. 8, it can be noticed that after passing through the small holes, the distances between UAV 28 and

UAV 15 and UAV 20 return to $d - \delta$. This indicates that after passing through the small hole, UAV 28, along with UAV 15 and UAV 20, reestablishes a quasi α-lattice structure and then continues flying toward the target. Also, this proves that the proposed DGF algorithm effectively avoids the issue of structural fragmentation that may occur in the original Flocking algorithm during the obstacle avoidance process.

Figure 9 shows the minimum distance between UAV 28 and the other UAVs at each time step. It can be observed from Fig. 9 that although the minimum distance fluctuates at each time step, it never falls below the quasi lattice spacing of $d - \delta$. This indicates that the UAVs consistently maintain a safe distance during flight, thereby demonstrating the effectiveness of the proposed DGF algorithm in avoiding hard collisions between the UAVs.

Figure 10 illustrates the trajectories of the UAV swarm under different control algorithms with $N = 8$. Figure 10a shows the trajectory of the UAV swarm for the original Flocking algorithm, while Fig. 10b shows the trajectory of the UAV swarm for the proposed DGF algorithm. From Fig. 10a, we can see that in the case of original Flocking algorithm, a phenomenon occurs that the UAV swarm gets blocked in the front of a small hole because only one UAV can pass through it. Moreover, the UAV swarm is simultaneously subjected to the repulsive force of the obstacle wall and the attraction of the target agent, thus resulting in potential function of 0, so the UAV swarm remains in a stable state outside the wall. In addition, the UAVs break the lattice distance constraint in front of the obstacle wall and inter-UAV collision occurs. This is because the attraction from the targets to the UAV far exceeds the repulsion from the obstacle as well as other UAVs, and the UAVs eventually collide when the safe distance is gradually compressed. From Fig. 10b, we can see that with the DGF algorithm, the UAVs can get the optimal sequence of hole crossing after the dual-game and finally reach the target points smoothly. Thus, it can be seen that the proposed DGF algorithm can guide the UAV swarm to pass through the small holes smoothly while avoiding the collision between UAVs, which also confirms its reliability.

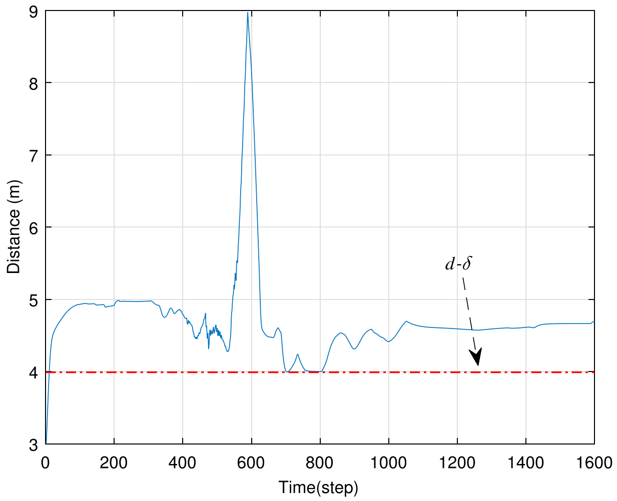
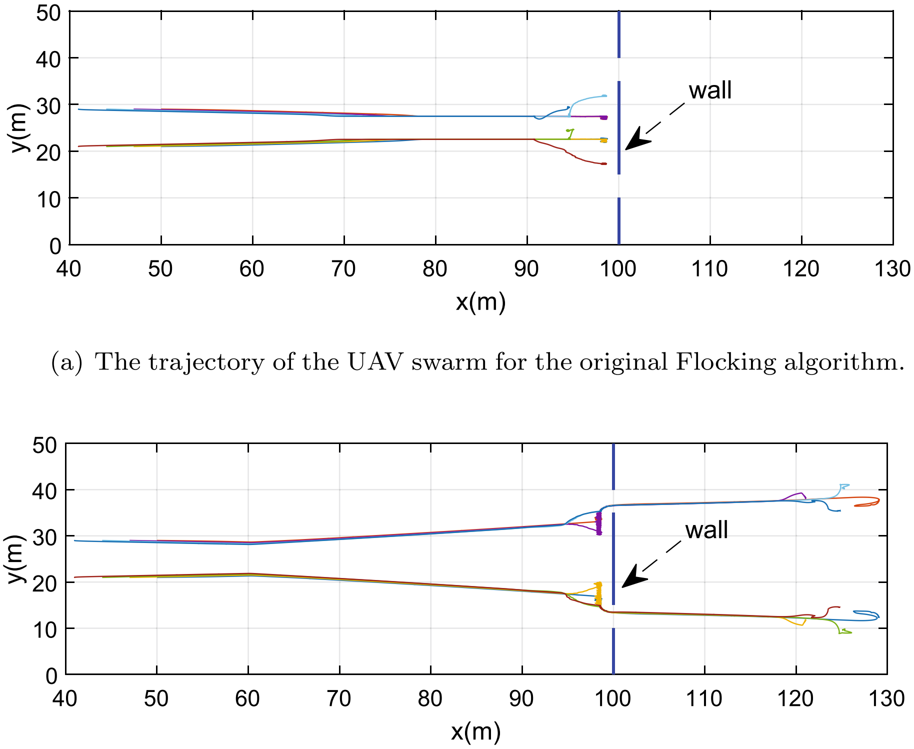


Fig. 9 Curve of minimum distance between UAV 28 and other UAVs



(a) The trajectory of the UAV swarm for the original Flocking algorithm.

(b) The trajectory of the UAV swarm for the proposed DGF algorithm.

Fig. 10 Trajectory of UAV swarm under different algorithms

6 Conclusions

In order to enable UAV swarm to pass through multi-narrow type obstacles in an orderly manner, a dual-game real-time obstacle avoidance scheme is proposed in this paper, aiming to realize the matching separation and obstacle avoidance control for UAV swarm. For the proposed scheme, a dual-game based Flocking obstacle avoidance algorithm is proposed. Specifically, the obtained motion states of each UAV from the game are transformed into actual motion control inputs, which are then integrated with the Flocking algorithm to determine the actual motion parameters of each UAV. Finally, the obstacle avoidance process is successfully accomplished after multiple iterations. Simulation results demonstrate that the proposed DGF algorithm enables the UAV swarm to pass through multi-narrow type obstacle scenarios successfully. Moreover, the algorithm effectively solves the hard collision problem between UAVs and notably improves the safety of the system.

Acknowledgements
We appreciate the editors and reviewers who processed and reviewed our manuscript to provide the detailed professional comments on the technical contributions, logical structure and content presentation of this paper.

Author contributions

YL was involved in conceptualization, data curation, validation and writing—original draft preparation. ZN was involved in funding acquisition, resources, supervision and writing review and editing. ZF was involved in methodology, software, visualization and writing—original draft preparation. BL was involved in resources, supervision and writing review and editing. YL was involved in supervision and writing review and editing.

Funding

This work was supported in part by the National Key Research and Development Program of China (2019YFE0111600), in part by the Natural Science Foundation of Liaoning Province (2023-MS-124), in part by the National Natural Science Foundation of China (61971081, 51939001 and 62371085), in part by the Liaoning Revitalization Talents Program (XLYC2002078), and in part by the Fundamental Research Funds for the Central Universities (3132023514).

Availability of data and materials

Parts of the models, data and codes that support the study are available from the corresponding author upon reasonable request.

Declarations

Competing Interests

The authors declare that they have no competing interests.

Received: 25 August 2023 Accepted: 6 November 2023
Published online: 16 November 2023

References

1. Y. Jiang, Y. Gao, W. Song, Y. Li, Q. Quan, Bibliometric analysis of UAV swarms. J. Syst. Eng. Electron. **33**(2), 406–425 (2022)
2. H. Pan, Y. Liu, G. Sun, J. Fan, S. Liang, C. Yuen, Joint power and 3D trajectory optimization for UAV-enabled wireless powered communication networks with obstacles. IEEE Trans. Commun. **71**(4), 2364–2380 (2023)
3. J. Wang, Y. Liu, S. Niu, H. Song, Extensive throughput enhancement for 5G-enabled UAV swarm networking. IEEE J. Miniat. Air Space Syst. **2**(4), 199–208 (2021)
4. Z. Na, B. Li, X. Liu, J. Wang, M. Zhang, Y. Liu, B. Mao, UAV-based wide-area Internet of Things: an integrated deployment architecture. IEEE Network **35**(5), 122–128 (2021)
5. N. Gao, L. Liang, D. Cai, X. Li, S. Jin, Coverage control for UAV swarm communication networks: a distributed learning approach. IEEE Internet Things J. **9**(20), 19854–19867 (2022)
6. R. Dong, B. Wang, K. Cao, T. Cheng, Securing transmission for UAV swarm-enabled communication network. IEEE Syst. J. **16**(4), 5200–5211 (2022)
7. W. Hu Jinqiang, Z.R. Husheng, M. Rafik, Z. Xuanwu, Self-organized search-attack mission planning for UAV swarm based on wolf pack hunting behavior. J. Syst. Eng. Electron. **32**(6), 1463–1476 (2021)
8. X. Shi, N. Deng, Modeling and analysis of mmwave UAV swarm networks: A stochastic geometry approach. IEEE Trans. Wireless Commun. **21**(11), 9447–9459 (2022)
9. X. Liu, B. Lai, B. Lin, V.C.M. Leung, Joint communication and trajectory optimization for multi-UAV enabled mobile internet of vehicles. IEEE Trans. Intell. Transp. Syst. **23**(9), 15354–15366 (2022)
10. J. Wang, Z. Na, X. Liu, Collaborative design of multi-UAV trajectory and resource scheduling for 6G-enabled Internet of Things. IEEE Internet Things J. **8**(20), 15096–15106 (2021)
11. Z. Na, Y. Liu, J. Shi, C. Liu, Z. Gao, UAV-supported clustered NOMA for 6G-enabled Internet of Things: trajectory planning and resource allocation. IEEE Internet Things J. **8**(20), 15041–15048 (2021)
12. Y. Wu, S. Tang, L. Zhang, Resilient machine learning based semantic-aware MEC networks for sustainable next-G consumer electronics. IEEE Trans. Consumer Electron. **99**, 1–11 (2023)
13. J. Tang, X. Chen, X. Zhu, F. Zhu, Dynamic reallocation model of multiple unmanned aerial vehicle tasks in emergent adjustment scenarios. IEEE Trans. Aerosp. Electron. Syst. **59**(2), 1139–1155 (2023)
14. W. Jia, C. Luo, Y. Luo, K. Li, Distributed UAV swarm formation and collision avoidance strategies over fixed and switching topologies. IEEE Trans. Cybern. **52**(10), 10969–10979 (2022)
15. L. Zhou, S. Leng, Q. Liu, Q. Wang, Intelligent UAV swarm cooperation for multiple targets tracking. IEEE Internet Things J. **9**(1), 743–754 (2022)
16. Q. Quan, F. Rao, K.-Y. Cai, Practical control for multicopters to avoid non-cooperative moving obstacles. IEEE Trans. Intell. Transp. Syst. **23**(8), 10839–10857 (2022)
17. J.L. Mata-Machuca, L.F. Zarazua, R. Aguilar-López, Experimental verification of the leader-follower formation control of two wheeled mobile robots with obstacle avoidance. IEEE Lat. Am. Trans. **19**(8), 1417–1424 (2021)
18. Tan, K.-H., Lewis, M.A., Virtual structures for high-precision cooperative mobile robotic control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, volume 1, pages 132–139. IEEE, (1996)
19. Yu. Tian, X. Zhu, D. Meng, X. Wang, B. Liang, An overall configuration planning method of continuum hyper-redundant manipulators based on improved artificial potential field method. IEEE Robot. Autom. Lett. **6**(3), 4867–4874 (2021)
20. S. Tang, Q. Yang, L. Fan, Contrastive learning based semantic communications. IEEE Trans. Commun. **99**, 1–5 (2023)
21. Z. Na, C. Ji, B. Lin, N. Zhang, Joint optimization of trajectory and resource allocation in secure UAV relaying communications for Internet of Things. IEEE Internet Things J. **9**(17), 16284–16296 (2022)
22. Y. Liu, Z. Na, Y. Zhang, F. Qin, B. Lin, Multi-UAV-assisted covert communications for secure content delivery in Internet of Things. Comput. Commun. **210**, 138–146 (2023)
23. R. Olfati-Saber, Flocking for multi-agent dynamic systems: algorithms and theory. IEEE Trans. Autom. Control **51**(3), 401–420 (2006)
24. Hu, N., Wu, S., Flocking of multi-agents with conic obstacle avoidance. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, volume 1, pp 126–132. IEEE, (2012)

25. D. Huang, Q. Yuan, X. Li. Decentralized Flocking of multi-agent system based on MPC with obstacle/collision avoidance. In *2019 Chinese Control Conference (CCC)*, pp 5587–5592. IEEE (2019)
26. H. Wei, X.-B. Chen, Flocking for multiple subgroups of multi-agents with different social distancing. IEEE Access **8**, 164705–164716 (2020)
27. Chang, D.E., Marsden, J.E. Gyroscopic forces and collision avoidance with convex obstacles. In *New trends in nonlinear dynamics and control and their applications*, pages 145–159. Springer, (2004)
28. E. Olcay, B. Lohmann, M.R. Akella, An information-driven algorithm in Flocking systems for an improved obstacle avoidance. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 298–304. IEEE (2019)
29. M.M. Khrustalev, Synthesis of adaptive control strategies for flocks of mobile robots using aristotle's mechanics. IFAC-PapersOnLine **51**(32), 525–529 (2018)
30. X. Yang, S. Zhao, D. Luo, Y. You, Affine formation maneuver control of high-order multi-agent systems over directed networks. Automatica **118**, 109004 (2020)
31. S. Zhao, D. Zelazo, Bearing rigidity theory and its applications for control and estimation of network systems: life beyond distance rigidity. IEEE Control Syst. Mag. **39**(2), 66–83 (2019)
32. X. Zhou, J. Zhu, H. Zhou, C. Xu, F. Gao, EGO-swarm: a fully autonomous and decentralized quadrotor swarm system in cluttered environments. In *2021 IEEE international conference on robotics and automation (ICRA)*, pp 4101–4107. IEEE (2021)
33. J. Park, D. Kim, G.C. Kim, O. Dahyun, H. Jin Kim, Online distributed trajectory planning for quadrotor swarm with feasibility guarantee using linear safe corridor. IEEE Robot. Autom. Lett. 7(2), 4869–4876 (2022)
34. Y. Gao, C. Bai, Q. Quan, Distributed control for a multi-agent system to pass through a connected quadrangle virtual tube. IEEE Trans. Control Netw. Syst. (2022)
35. D. Panagou, D.M. Stipanović, P.G. Voulgaris, Distributed coordination control for multi-robot networks using lyapunov-like barrier functions. IEEE Trans. Autom. Control **61**(3), 617–632 (2015)
36. Y. Cen, Y. Ye, N. Xie, J. Bao, C. Song, Flocking task research for multiple mobile robots based on game theory. In *2008 3rd IEEE Conference on Industrial Electronics and Applications*, pp 46–49. IEEE, (2008)
37. M. Jafari, H. Xu, A game theoretic based biologically-inspired distributed intelligent Flocking control for multi-UAV systems with network imperfections. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp 1138–1144. IEEE (2018)
38. Y. Lin, Z. Na, J. Liu, Y. Lin, Multi-narrow type obstacle avoidance algorithm for UAV swarm based on game theory. Adv. Control Appl. e168 (2023)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.