Information Sciences 509 (2020) 515–529

Contents lists available at ScienceDirect

Information Sciences

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

A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles☆

Huaxin Qiu[a,b], Haibin Duan[a,*]

[a] State Key Laboratory of Virtual Reality Technology and Systems, School of Automation Science and Electrical Engineering, Beihang University (BUAA), Beijing 100083, PR China
[b] Shen Yuan Honors College, Beihang University (BUAA), Beijing 100083, PR China

ARTICLE INFO

Article history:
Received 24 January 2018
Revised 22 June 2018
Accepted 29 June 2018
Available online 2 July 2018

Keywords:
Unmanned aerial vehicle
Flocking control
Obstacle avoidance
Many-objective optimization
Pigeon-inspired optimization

ABSTRACT

Unmanned aerial vehicle (UAV) flocking control with obstacle avoidance is a many-objective optimization problem for centralized algorithms. A UAV flocking distributed optimization control frame is designed to render the many-objective optimization problem into a multi-objective optimization solved by a single UAV. For different objectives, two kinds of criteria are raised to guarantee flight safety: the hard constraints that must be satisfied and the soft ones that will be optimized. Considering the restrictions of onboard computing resources, multi-objective pigeon-inspired optimization (MPIO) is modified based on the hierarchical learning behavior in pigeon flocks. On such a basis, a UAV distributed flocking control algorithm based on the modified MPIO is proposed to coordinate UAVs to fly in a stable formation under complex environments. Comparison experiments with basic MPIO and a modified non-dominated sorting genetic algorithm (NSGA-II) are carried out to show the feasibility, validity, and superiority of the proposed algorithm.

1. Introduction

The unmanned aerial vehicle (UAV), an aircraft designed for missions that are too dull, dirty or dangerous for humans, has attracted the widespread attention of researchers [3,20,25]. The future UAV will possess distributed autonomous airborne abilities to complete a four-step loop independently [10]: observe, orient, decide, and act (OODA). The improvement of distributed airborne abilities will trigger changes in future mission modes where small and smart UAVs will execute a task in a group or a swarm [7,22,24]. A great number of problems exist in the course of the realization of the future mission mode, such as how to coordinate a group of agents to move in a formation under complex obstacle environments [11,15].

In the literature, Wang et al. [28] presented a new dual-mode control strategy for navigation problems of UAVs flying in a formation in an obstacle environment, and the time taken by the computations is a function of the number of UAVs on collision paths. Saska [26] proposed a bio-inspired stabilization approach for control and navigation of large teams of UAVs along a predefined path through a complex environment, and the UAV flocking is unsteady for lack of the strict velocity alignment. Alonso-Mora et al. [1] realized multi-robot formation control among obstacles by a sophisticated algorithm based on lots of local interaction information. In our previous work, a flocking control algorithm based on pigeon interaction mode

☆ This work was partially supported by National Natural Science Foundation of China (NSFC) # 61425008, # 61333004 and # 91648205, and Aeronautical Science Foundation of China # 2015ZA51013.

* Corresponding author.
E-mail address: hbduan@buaa.edu.cn (H. Duan).

https://doi.org/10.1016/j.ins.2018.06.061
0020-0255/

switch behavior, which relies on the prior information of obstacles, is proposed to coordinate a heterogeneous UAV swarm to fly through obstacle environments [23]. This paper aims to put forward a UAV flocking control algorithm, which can form a stable formation among obstacles without depending on the number of UAVs, other information of neighbors except positions and velocities, or prior information of environments.

To implement the stable and collision-free collective motion of a UAVs swarm under obstacle environments, optimal control decisions need to be made in the presence of many conflicting criteria [2,14]. These criteria depict trade-offs not only between individual flocking control and obstacle avoidance control but also between individual control and other individual control. Therefore, the UAV flocking centralized control problem with obstacle avoidance is a many-objective optimization problem ($n \cdot m$ objectives) where n is the number of UAVs and m is the number of objectives of each UAV. In this paper, a distributed UAV flocking control optimization frame is designed to render the many-objective optimization problem into a multi-objective optimization problem (m objectives) to be solved by each UAV. The objectives of each UAV are summarized as follows: 1) If obstacles are detected, pass through obstacles as soon as possible; If not, keep expected states; 2) Maintain a presupposed distance with neighbors steadily; 3) Keep a safe distance with obstacles; 4) Avoid collisions with neighbors. The above objectives are divided into soft constrains (1 and 2) and hard ones (3 and 4). Each UAV aims to minimize the soft and satisfy the hard. However, considering the restrictions of onboard computing, the time requirement for the multi-objective optimization algorithm computations is very strict.

Pigeon-inspired optimization (PIO), a novel bio-inspired computing algorithm, was proposed by Duan and Qiao [8]. Whether in image restoration [9] or UAV path planning [30], PIO has proven its worth in mono-objective optimization problems. However, the superiority of multi-objective pigeon-inspired optimization (MPIO) [21] is not apparent in multi-objective optimization problems compared with the modified non-dominated sorting genetic algorithm (NSGA-II) [6,13,19]. In this paper, a hierarchical learning behavior discovered in pigeon flocks [17,18] is adopted to modify the basic MPIO. By the designed distributed UAV flocking control optimization frame and modified MPIO, a UAV distributed flocking control algorithm is proposed to coordinate UAVs to fly in a stable formation under complex obstacle environments.

The rest of the paper is organized as follows. Section 2 gives a brief review of the principle of basic PIO. After building the UAV model, self-propelled flocking model, and obstacle avoidance model, Section 3 formulates the multi-objective optimization problem to be solved in this paper. Section 4 presents modified MPIO based on the hierarchical learning behavior in pigeon flocks. Section 5 proposes a distributed UAV flocking control algorithm with obstacle avoidance based on the modified MPIO in Section 4. Comparative simulation validations are elaborated in Section 6, and our concluding remarks are drawn in Section 7.

2. Pigeon-inspired optimization

A pigeon can return to its loft over great distances by using three navigation tools: the sun [29], the familiar visual landmark [4], and the earth's magnetic field [16]. As the pigeon gradually moves to its loft, the effect of the sun and the magnetic field on its navigation will decline progressively [12]. The pigeon will correct its route by known landmark image messages or will follow other pigeons which are familiar with nearby landmarks. Inspired by the above pigeon navigation behavior, PIO is proposed to provide a new approach to optimization problems [8]. In PIO, the position of each pigeon and the loft represent the potential solution and the optimal solution to an optimization problem respectively. In other words, the behavior of pigeon homing represents the convergence process of solutions to the global optimum. The map and compass operator is presented to mimic the navigation effect of the sun and the magnetic field on pigeons, while the landmark operator is raised to imitate the impact of familiar visual landmarks on the pigeon homing. PIO employs the two independent operators to optimize feasible solutions to optimization problems.

Consider N pigeons flying to their loft in a D-dimensional search space. The position $\boldsymbol{X}_i^{Nc}$ and velocity $\boldsymbol{V}_i^{Nc}$ of pigeon i at iteration Nc is updated by the map and compass operator expressed in the following equation when iteration Nc is not greater than the maximum iteration of the map and compass operator $Nc_{\max}^1$:

$$\begin{cases} \boldsymbol{V}_i^{Nc} = e^{-R \cdot Nc} \cdot \boldsymbol{V}_i^{Nc-1} + rand \cdot (\boldsymbol{X}_{\boldsymbol{g}} - \boldsymbol{X}_i^{Nc-1}) \\ \boldsymbol{X}_i^{Nc} = \boldsymbol{X}_i^{Nc-1} + \boldsymbol{V}_i^{Nc} \end{cases} \tag{1}$$

where $\boldsymbol{X}_i^{Nc}$, $\boldsymbol{V}_i^{Nc}$, and the current global best position $\boldsymbol{X}_{\boldsymbol{g}}$ are D-dimensional vectors, i=1, 2, ..., N is the index of a pigeon, R is the map and compass factor, and *rand* is a random number within [0,1].

When iteration Nc is within $(Nc_{\max}^1, Nc_{\max}^2]$, the position $\boldsymbol{X}_i^{Nc}$ and velocity $\boldsymbol{V}_i^{Nc}$ of pigeon i at iteration Nc is updated by the landmark operator expressed in the following equation where $Nc_{\max}^2$ is the maximum iteration of the landmark operator:

$$\begin{cases} N = [N/2] \\ \boldsymbol{X}_{\boldsymbol{center}}^{Nc-1} = \frac{\sum_{i=1}^{N} \boldsymbol{X}_i^{Nc-1} \cdot W(\boldsymbol{X}_i^{Nc-1})}{\sum_{i=1}^{N} W(\boldsymbol{X}_i^{Nc-1})} \\ \boldsymbol{X}_i{}^{Nc} = \boldsymbol{X}_i^{Nc-1} + rand \cdot \left(\boldsymbol{X}_{\boldsymbol{center}}{}^{Nc-1} - \boldsymbol{X}_i^{Nc-1}\right) \end{cases} \tag{2}$$

where [] is the ceiling function, and the weighted average of positions $\boldsymbol{X_{center}}$ is a D-dimensional vector. For maximum optimization problems, the weight $W(\boldsymbol{X}_i^{Nc-1})$ is defined as the cost function value $fitness(\boldsymbol{X}_i^{Nc-1})$. For minimum optimization problems, the weight $W(\boldsymbol{X}_i^{Nc-1})$ is defined as $\frac{1}{fitness(\boldsymbol{X}_i^{nc-1})+\varepsilon}$ where ε is a constant.

3. Problem formulation

3.1. Unmanned aerial vehicle model

Consider a swarm of n UAVs flying in a three-dimensional Euclidean space, and each UAV i has a position $\boldsymbol{P}^i = (x^i, y^i, h^i)$ in the inertial coordinate system, a horizontal airspeed $V_{\mathrm{xy}}{}^i$, a yaw angle ψ^i, and an altitude rate λ^i. In this paper, the complete UAV model is composed of a first order Mach-hold autopilot model, a first order heading-hold autopilot model, and a second order altitude-hold autopilot model. The model of UAV i is as the following equation [23]:

$$\begin{cases} \dot{x}^i = V_{\mathrm{xy}}{}^i \cos\psi^i \\ \dot{y}^i = V_{\mathrm{xy}}{}^i \sin\psi^i \\ \dot{h}^i = \lambda^i \\ \dot{V}_{\mathrm{xy}}^i = \frac{1}{\tau_{\mathrm{v}}}(V_{\mathrm{xy\_c}}^i - V^i) \\ \dot{\psi}^i = \frac{1}{\tau_{\psi}}(\psi_{\mathrm{c}}^i - \psi^i) \\ \dot{\lambda}^i = \frac{1}{\tau_{\mathrm{h}}}(h_{\mathrm{c}}^i - h^i) - \frac{1}{\tau_{\lambda}}\lambda^i \end{cases} \tag{3}$$

where $V_{\mathrm{xy\_c}}^i$, ψ_{c}^i, and h_{c}^i are the control inputs of UAV i's Mach-hold autopilot, heading-hold autopilot, and altitude-hold autopilot respectively, and τ_{v}, τ_{ψ}, and $(\tau_{\mathrm{h}}, \tau_{\lambda})$ are the time constants of the three autopilots respectively. The UAV motion and autopilot saturation values considered in this paper are as follows:

$$\begin{cases} V_{\mathrm{xy\_min}} \le V_{\mathrm{xy}}^i \le V_{\mathrm{xy\_max}} \\ \left|\dot{\psi}^i\right| \le n_{\max} g / V_{\mathrm{xy}}^i \\ \lambda_{\min} \le \lambda^i \le \lambda_{\max} \end{cases} \tag{4}$$

where $V_{\mathrm{xy\_max}}$ and $\lambda_{\max}$ are the upper limits of the horizontal airspeed and altitude rate respectively, $V_{\mathrm{xy\_min}}$ and $\lambda_{\min}$ are the lower limits of the horizontal airspeed and altitude rate respectively, $n_{\max}$ is the maximum lateral overload, and $g = 10\mathrm{m/s}^2$ is the gravitational acceleration.

3.2. Self-propelled flocking model

During flocking flights of a UAV swarm, the desired flocking velocity $\boldsymbol{vf}^i$ of UAV i is a function of the positions $\boldsymbol{P}^j$ and velocities $\boldsymbol{v}^j = (V_{\mathrm{xy}}{}^j \cos\psi^j, V_{\mathrm{xy}}{}^j \sin\psi^j, \lambda^j)$ of UAV j in the swarm where j=1, 2, …, n [27]. In this paper, the desired flocking velocity $\boldsymbol{vf}^i$ is decoupled into the following horizontal- and vertical- directional equations:

$$\dot{\boldsymbol{vf}}_k^i = \begin{cases} f_{\mathrm{f}} + f_c + f_{\mathrm{a\_vn}}, & k = 1, 2 \\ K_{a\_h_e}(h_{\mathrm{e}} - \boldsymbol{P}_k^i) + K_{\mathrm{v}_e}(\boldsymbol{ve}_k - \boldsymbol{v}_k^i), & k = 3 \end{cases} \tag{5}$$

where the horizontal directional equation, which consists of flocking geometry control component f_{f}, collision avoidance control component f_c, and alignment control component $f_{\mathrm{a\_vn}}$, is designed to guarantee stable and collision-free collective motion. The vertical directional equation is intended to control the UAV swarm to fly at an expected altitude, where $K_{a\_h_e}$ is the strength of the alignment with the scheduled altitude h_{e}, K_{v_e} is the strength of the alignment with the expected vertical velocity $\boldsymbol{ve}_3$, and $\boldsymbol{ve}$ is the anticipated velocity of flocking flight.

The flocking geometry control component f_{f} is as the following equation:

$$f_{\mathrm{f}} = K_{\mathrm{f}}\left(\sum_{j\in\left\{d^{ij}\le R_{\mathrm{comm.}}^1\right\}} \boldsymbol{w}_i^j(\boldsymbol{P}_k^j - \boldsymbol{P}_k^i)\left(1 - \left(\frac{R_{\mathrm{desire}}}{d^{ij}}\right)^2\right)\right) \tag{6}$$

where K_{f} is the strength of flocking control, $d^{ij}=\sqrt{\left(x^i - x^j\right)^2 + \left(y^i - y^j\right)^2}$ is the horizontal distance between UAV i and j, $R_{\mathrm{comm.}}^1$ is the horizontal communication range, $\boldsymbol{w}_i^j$ is the influence weight of UAV j to UAV i, and R_{desire} is the factor of flocking geometry control.

The collision avoidance control component f_c is as the following equation:

$$f_c = K_{\mathrm{c}} \sum_{j\in\left\{d^{ij}\le R_{\mathrm{lim}}^1\right\}} \left(\frac{1}{\left|\boldsymbol{P}_k^i - \boldsymbol{P}_k^j\right|} - \frac{1}{R_{\mathrm{lim}}^1}\right)^2 \frac{\left(\boldsymbol{P}_k^i - \boldsymbol{P}_k^j\right)}{\left|\boldsymbol{P}_k^i - \boldsymbol{P}_k^j\right|} \tag{7}$$

where K_{c} is the strength of collision avoidance control, and R_{lim}^1 is the maximum range of collision avoidance control.

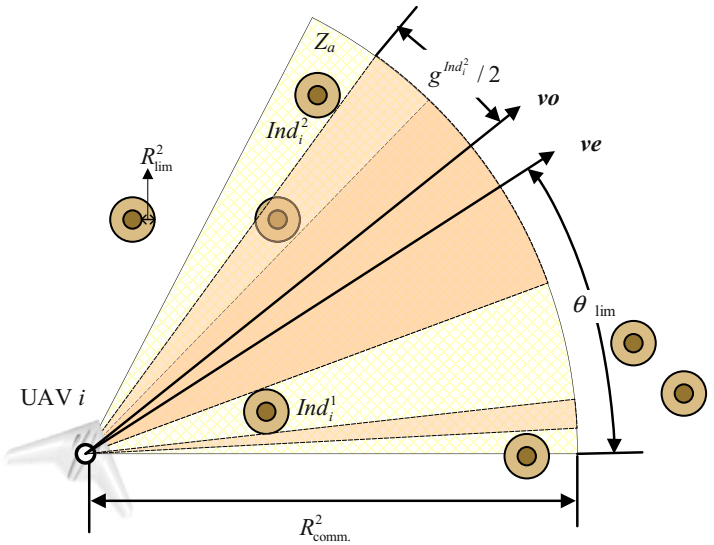


Fig. 1. Obstacle avoidance model.

The alignment control component $f_{\text{a\_vn}}$ is as the following equation:

$$f_{\text{a\_vn}}=K_{\text{a\_vn}}\left(\sum_{j\in\left\{d^{ij}\le R^1_{\text{comm.}}\right\}}\boldsymbol{w}_i^j(\boldsymbol{v}_l^j-\boldsymbol{v}_l^i)\right) \tag{8}$$

where $K_{\text{a\_vn}}$ is the strength of the alignment with neighbors' velocities.

3.3. Obstacle avoidance model

In this paper, the obstacle avoidance control is considered as a series of gap-aiming behaviors. For each time step, each UAV attempts to move through the largest gap between obstacles. The procedure of the proposed obstacle avoidance control model is as follows:

Step 1 Identify the nearest obstacle: As shown in Fig. 1, the horizontal attention zone Z_{a}^i of UAV i for obstacle avoidance is a sector of radius $R^2_{\text{comm.}}$ and central angle $2\theta_{\text{lim}}$, where $R^2_{\text{comm.}}$ is the horizontal perception range and θ_{lim} is the field of view. The expected horizontal velocity direction is in the symmetrical center line of the horizontal attention zone Z_{a}^i, where the angle between the horizontal velocity direction and the x axis of the inertial coordinate system is yaw angle. If there is a point on an external envelope surface of obstacle j within the horizontal attention zone Z_{a}^i, the obstacle j will be detected by UAV i and the index of obstacle j will be saved in obstacle set A_{o}^i of UAV i, where $j=1,2,\ldots,n_{\text{o}}$, n_{o} is the number of obstacles, the distance between the envelope and obstacle j is R^2_{lim}, and R^2_{lim} is the minimum allowable distance between UAVs and obstacles. The index of the nearest obstacle detected by UAV i is as the following equation:

$$Ind_i^1=\arg\min_{j\in A_{\text{o}}^i}(d_{\text{o}}^{ij}) \tag{9}$$

where d_{o}^{ij} is the minimum distance between UAV i and obstacle j.

Step 2 Identify the largest visual gap: To maximize clearance, UAV i will calculate the gap g^j between obstacle Ind_i^1 and obstacle j identified within the attention zone. The index of the obstacle corresponding to the largest visual gap is as the following equation:

$$Ind_i^2=\operatorname{argmax}_{j\in A_{\text{o}}^i}(g^j) \tag{10}$$

Step 3 Calculate the desired velocity for obstacle avoidance: UAV i will select the symmetrical center line of the largest visual gap $g^{Ind_i^2}$ as the steering aim θ_{m}. If there is only one obstacle within the attention zone, the steering aim $\theta_{\text{m}}=\operatorname{atan}\left(\frac{\boldsymbol{obs}_2^{Ind_i^1}-y^i}{\boldsymbol{obs}_1^{Ind_i^1}-x^i}\right)$ where $\boldsymbol{obs}^{Ind_i^1}$ is the position of a point on the external envelope surface of obstacle Ind_i^1, and the point is the marginal point, closest to UAV i, of the projection of the envelope on the vertical direction of expected

horizontal velocity. If there are no obstacles within the attention zone, the steering aim θ_{m} is the desired yaw angle $\psi_{\mathrm{m}}=\mathrm{atan}(\frac{\boldsymbol{ve}_2}{\boldsymbol{ve}_1})$. The desired obstacle avoidance velocity $\boldsymbol{vo}^i$ for UAV i is as the following equation:

$$\boldsymbol{vo}_k^i=\begin{cases}\boldsymbol{w}_i^i\|\boldsymbol{ve}_{1,2}\|\cos\theta_{\mathrm{m}}, & k=1\\ \boldsymbol{w}_i^i\|\boldsymbol{ve}_{1,2}\|\sin\theta_{\mathrm{m}}, & k=2\end{cases} \tag{11}$$

where $\boldsymbol{w}_i^i$ is the influence weight of obstacle avoidance judgement.

3.4. Performance criteria

In this paper, the criteria for UAV flocking control under obstacle environments are described in four objective functions. If there are no obstacles within the attention zone, the first objective function $\boldsymbol{Cost}_i^1$ defines the degree of passage through the obstacle environments by the projection of the horizontal position of UAV i on the expected horizontal velocity direction. If not, the first objective function $\boldsymbol{Cost}_i^1$ describes the alignment degree with the expected horizontal velocity. The first objective function $\boldsymbol{Cost}_i^1$ is defined as the following equation:

$$\boldsymbol{Cost}_i^1=\begin{cases}-\frac{\boldsymbol{P}_{1,2}^i\cdot\boldsymbol{ve}_{1,2}}{\|\boldsymbol{ve}_{1,2}\|}, & if\ A_{\mathrm{o}}^i\neq\emptyset\\ \left|\boldsymbol{ve}_1-\dot{x}^i\right|+\left|\boldsymbol{ve}_2-\dot{y}^i\right|, & if\ A_{\mathrm{o}}^i=\emptyset\end{cases} \tag{12}$$

The second objective function $\boldsymbol{Cost}_i^2$ depicts the quality of UAV flocking and the alignment degree with neighbors' velocities, which is expressed in the following equation:

$$\boldsymbol{Cost}_i^2=\sum_{j\in\left\{d^{ij}\le R_{\mathrm{comm.}}^1\right\}}\left(f_1\left|R_{\mathrm{desire}}-d^{ij}\right|+f_2\left(\left|\dot{x}^j-\dot{x}^i\right|+\left|\dot{y}^j-\dot{y}^i\right|\right)\right) \tag{13}$$

where f_1 and f_2 are the weight of the alignment degree with the desired flocking geometry and neighbors' velocities respectively.

The above two objective functions are soft constrains, while the next two objective functions are hard ones that the UAV must fulfill. The third objective function $\boldsymbol{Cost}_i^3$ shown in the following equation represents the quality of obstacle avoidance control:

$$\boldsymbol{Cost}_i^3=\begin{cases}1, & \text{if } \exists d_{\mathrm{o}}^{ij}\le R_{\mathrm{lim}}^2\\ 0, & \text{if } \forall d_{\mathrm{o}}^{ij}>R_{\mathrm{lim}}^2\end{cases} \tag{14}$$

where $j=1,2,\ldots,n_{\mathrm{o}}$.

The fourth objective function $\boldsymbol{Cost}_i^4$ expressed in the following equation portrays the quality of collision avoidance control:

$$\boldsymbol{Cost}_i^4=\begin{cases}1, & \text{if } \exists d^{ij}\le R_{\mathrm{lim}}^1\\ 0, & \text{if } \forall d^{ij}>R_{\mathrm{lim}}^1\end{cases} \tag{15}$$

where $j=1,2,\ldots,n$.

Integrating the above models, the UAV flocking control problem under obstacle environments is transformed into finding an influence weight vector $\boldsymbol{w}_i$ of UAV i which satisfies the following conditions:

(1) Minimize the soft constrains:

$$\min fitness=[\boldsymbol{Cost}_i^1,\boldsymbol{Cost}_i^2] \tag{16}$$

(2) Satisfy the hard constrains:

$$\begin{cases}\boldsymbol{Cost}_i^3=0\\ \boldsymbol{Cost}_i^4=0\end{cases} \tag{17}$$

4. Multi-objective pigeon-inspired optimization based on hierarchical learning

To solve the UAV flocking control problem under obstacle environments described by Eqs. (16) and (17), modified MPIO is proposed, where a hierarchical learning behavior is introduced to improve the capability of the proposed algorithm.

By analyzing the flight data gathered by miniature GPS during multiple pigeon flocking flights, a hierarchical network was discovered in the in-flight leader-follower relations of pigeons [17,18]. In a pigeon flock, except the general leader whose motion will not be influenced by the other pigeon, each pigeon has its rank in the hierarchy. During the flight, pigeons will attempt to follow the ones in upper ranks and lead the ones in lower ranks. The leadership hierarchy is hypothesized to be the result of feedback between learning and competence [5].

Inspired by the hierarchical learning in pigeon flocks, modified MPIO is proposed. In the basic PIO, all the pigeons will correct their positions $\boldsymbol{X}_i^{Nc}$ based on the sun and magnetic field described by the current global best position $\boldsymbol{X}_{\boldsymbol{g}}$, and the

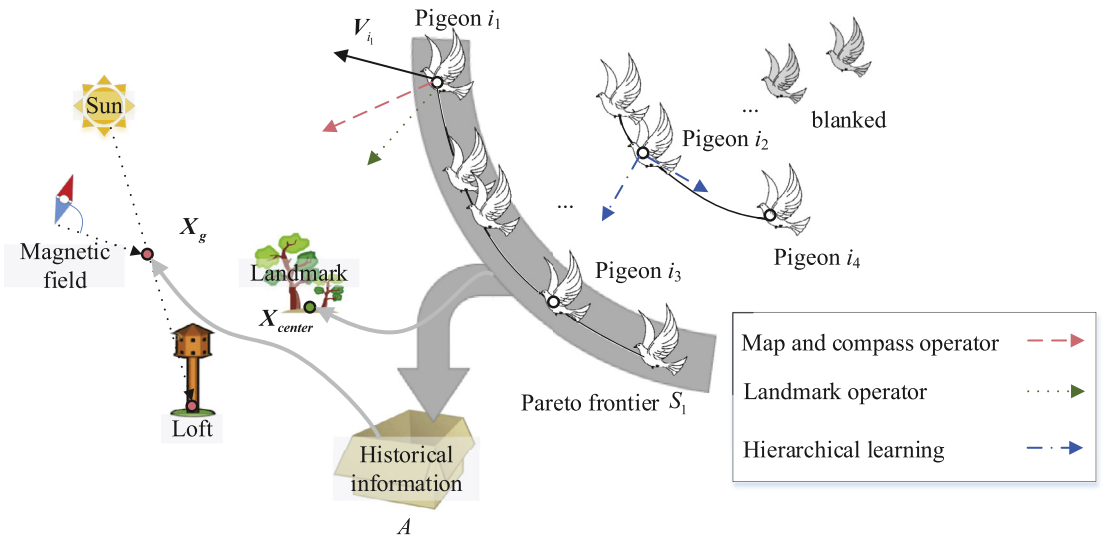


Fig. 2. Modified multi-objective pigeon-inspired optimization.

landmark image preview message specified by the weighted average of positions $\boldsymbol{X_{center}}$. In the modified MPIO, pigeons are split into two roles: One is the general leader (pigeon i_1 as shown in Fig. 2) and the other is the ordinary follower (pigeon i_2 as shown in Fig. 2). By the non-dominated sorting in Pareto sorting scheme, all the pigeons will be divided into different sets: first frontier S_1 (Pareto frontier), second frontier S_2, and so on. The crowded-comparison operator will continue to sort the pigeons in each set. As a result, a sequence of pigeons in descending order will be generated by the Pareto sorting scheme, and the sequence number N_o^i of pigeon i is its rank in the hierarchy. The pigeons with rank $N_o^i \leq [p_1 \cdot N]$ will be regarded as general leaders, which are supposed to fly based on the map and compass operator and the landmark operator, and update their states by the current global best position $\boldsymbol{X_g}$ and the weighted average of positions $\boldsymbol{X_{center}}$, where p_1 is the percentage of general leaders in the pigeon flock. The other pigeons will be regarded as ordinary followers, which are supposed to learn the pigeons in the upper ranks (pigeon i_3 and i_4 as shown in Fig. 2) by copying their positions. The pseudocode of the modified MPIO is shown below:

Algorithm Modified MPIO.

Input: function to be optimized ***Cost***
dimension of search space D
upper bound of positions $\boldsymbol{X_U}$
lower bound of positions $\boldsymbol{X_L}$
upper bound of velocities $\boldsymbol{V_U}$
lower bound of velocities $\boldsymbol{V_L}$
number of pigeons N
maximum iteration $Nc^3_{\max}$
map and compass factor R
transition factor f_t
reduced number of pigeons at each iteration N_d
percentage of general leaders in the pigeon flock p_1
learning error e
learning strength s_l
Initialization: positions of pigeons $\boldsymbol{X}^1$
velocities of pigeons $\boldsymbol{V}^1$
iteration $Nc \leftarrow 1$
Calculate objective function value $Cost$
For $Nc = 1 \rightarrow Nc^3_{\max}$ **Do**
Sort positions $\boldsymbol{X}^{Nc}$ by Pareto sorting
Calculate landmark $\boldsymbol{X_{center}}^{Nc}$ by Eqs. (18)
historical information set $A \leftarrow \{A, S_1\}$
Sort A by Pareto sorting
$A \leftarrow S_1^A$
Calculate current global best position $\boldsymbol{X_g}$
$Nc \leftarrow Nc+1$
For $i = 1 \rightarrow N$ **Do**
If $N_o^i \leq [p_1 \cdot N]$ **Then**

(*continued on next page*)

```
11:        Calculate velocity V_i^Nc by Eqs. (19)
12:        Calculate position X_i^Nc by Eqs. (20)
13:      Else
14:        For k = 1 → s_l Do
15:          Calculate position X_i^Nc by Eqs. (21)
16:        End For
17:      End If
18:      Calculate objective function value Cost_i
19:      If X_i^{Nc-1} ≺ X_i^Nc Then
20:        X_i^Nc = X_i^{Nc-1}
21:      End If
22:    End For
23:    If Nc ≤ Nc^3_max Then
24:      If N − N_d ≤ N_o^i ≤ N Then
25:        X_i^Nc ← Null
26:      End If
27:      number of pigeons N ← N − N_d
28:    End If
29:  End For
30:  Sort positions X^Nc by Pareto sorting
31:  Return Pareto frontier S_1.
```

The specific steps of the modified MPIO are as follows:

Initialization: Initialize the positions $\boldsymbol{X}^1$ and velocities $\boldsymbol{V}^1$ of N pigeons within a D-dimensional search space. The upper bound and lower bound of the position $\boldsymbol{X}$ is $\boldsymbol{X_U}$ and $\boldsymbol{X_L}$ respectively. The upper bound and lower bound of the velocity $\boldsymbol{V}$ is $\boldsymbol{V_U}$ and $\boldsymbol{V_L}$ respectively. Calculate the objective function values $\boldsymbol{Cost}$ of all the pigeons. The maximum iteration is $Nc^3_{\max}$. The current iteration $Nc = 1$.

Step 1: Rank the pigeons' positions $\boldsymbol{X}^{Nc}$ by Pareto sorting scheme. The Pareto frontier S_1 and rank N_o^i of pigeon i are generated. Then obtain the landmark $\boldsymbol{X_{center}}^{Nc}$ by the following equation:

$$\boldsymbol{X_{center}}^{Nc} = \frac{\sum_{i=1}^{n_X} \boldsymbol{X}_i}{n_X} \tag{18}$$

where n_X is the number of positions in the Pareto frontier S_1.

Step 2: The Pareto frontier S_1 will be saved in a historical information set A. Rank all the positions saved in set A and obtain the Pareto frontier S_1^A. Empty set A and put the positions $\boldsymbol{X}_i^A$ in Pareto frontier S_1^A into set A, where $i = 1, 2, \ldots, n_A$ and n_A is the number of positions in the Pareto frontier S_1^A. Randomly select a position in the Pareto frontier S_1^A as the current global best position $\boldsymbol{X_g}$.

Step 3: The current iteration $Nc = Nc$+1.

Step 4: If pigeon i's rank $N_o^i \le [p_l \cdot N]$ where p_l is the percentage of general leaders in the pigeon flock, update velocity $\boldsymbol{V}_i^{Nc}$ by the following equation:

$$\begin{aligned}\boldsymbol{V}_i^{Nc} &= e^{-R\cdot Nc}\cdot \boldsymbol{V}_i^{Nc-1} + rand_1 \cdot f_t(1 - \lg^{Nc}_{Nc^3_{\max}})(\boldsymbol{X_g} - \boldsymbol{X}_i^{Nc-1}) \\ &\quad + rand_2 \cdot f_t \cdot \lg^{Nc}_{Nc^3_{\max}}(\boldsymbol{X_{center}}^{Nc-1} - \boldsymbol{X}_i^{Nc-1})\end{aligned} \tag{19}$$

where R is the map and compass factor, $rand_1$ and $rand_2$ are random numbers within [0,1], and f_t is the transition factor from the map and compass operator to the landmark operator. Then update position $\boldsymbol{X}_i^{Nc}$ by the following equation:

$$\boldsymbol{X}_i^{Nc} = \boldsymbol{X}_i^{Nc-1} + \begin{cases} \boldsymbol{V_U}^i, & if\ \boldsymbol{V}_i^{Nc} > \boldsymbol{V}_U^i \\ \boldsymbol{V}_i^{Nc}, & if\ \boldsymbol{V_L}^i \le \boldsymbol{V}_i^{Nc} \le \boldsymbol{V_U}^i \\ \boldsymbol{V_L}^i, & if\ \boldsymbol{V}_i^{Nc} < \boldsymbol{V_L}^i \end{cases} \tag{20}$$

Then revise velocity $\boldsymbol{X}_i^{Nc}$ within range $[\boldsymbol{X_L}^i, \boldsymbol{X_U}^i]$.

If pigeon i's rank $N_o^i > [p_l \cdot N]$, learn the pigeon in upper ranks by the following equation:

$$\boldsymbol{X}_i^{Nc}(d^*) = \boldsymbol{X}_j^{Nc-1}(d^*) + e \cdot rand \tag{21}$$

where $d^* = [rand \cdot D]$ is the index of dimension for learning, j is the index of the pigeon to be learned which satisfies that $N_o^j = [rand \cdot (N_o^i - 1)]$, and e is the learning error. The above learning process will repeat s_l times, where s_l is the learning strength.

Step 5: Calculate objective function value $\boldsymbol{Cost}_i$ of each pigeon. If $\boldsymbol{X}_i^{Nc-1}$ dominates $\boldsymbol{X}_i^{Nc}$ (i.e., $\boldsymbol{X}_i^{Nc-1} \prec \boldsymbol{X}_i^{Nc}$), $\boldsymbol{X}_i^{Nc} = \boldsymbol{X}_i^{Nc-1}$.

Step 6: If current iteration $Nc \le Nc^3_{\max}$, delete the pigeons which satisfy that rank $N_o^i \in [N - N_d{+}1, N]$, the number of pigeons $N = N - N_d$, and return to **Step 1**, where N_d is the reduced number of pigeons at each iteration. Otherwise, rank the pigeons' positions $\boldsymbol{X}^{Nc}$ by Pareto sorting scheme and output the Pareto frontier S_1.

As shown in the pseudocode, it is evident that the time complexity of the modified MPIO mainly lies in the hierarchical learning operation regardless of the Pareto sorting operation. The time complexity of the hierarchical learning operation is

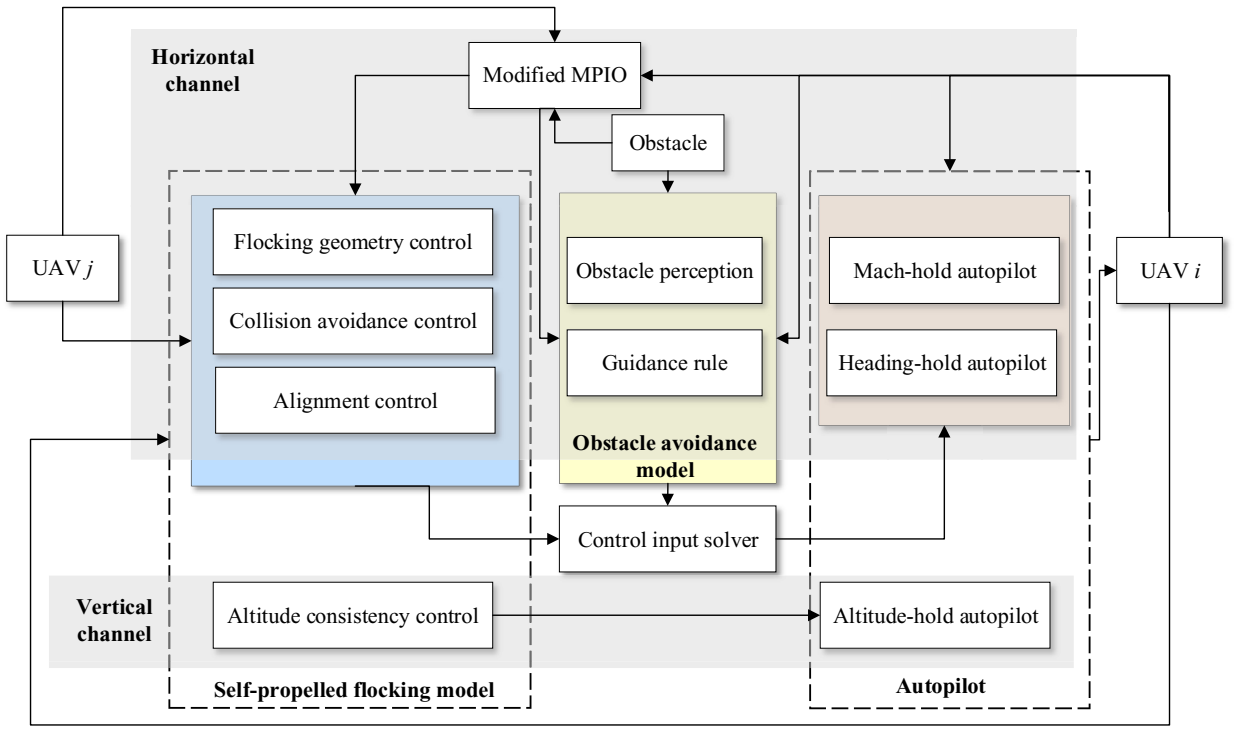


Fig. 3. UAV flocking control algorithm with obstacle avoidance.

$O(s_l N)$ in each iteration, while the Pareto sorting operation is $O(mN^2)$, where m is the number of the objective functions. Due to that $s_l < mN$, the time complexity of the modified MPIO is shown as the following equation:

$$\begin{aligned} T &= \sum_{Nc=1}^{Nc_{\max}^3} O\left(mN^2\right) = \sum_{Nc=1}^{Nc_{\max}^3} O\left(m(N - N_{\mathrm{d}}(Nc-1))^2\right) \\ &= O\left(m\left(Nc_{\max}^3 N^2 + \frac{Nc_{\max}^3\left(Nc_{\max}^3-1\right)\left(2Nc_{\max}^3-1\right)(N_{\mathrm{d}})^2}{6} - Nc_{\max}^3\left(Nc_{\max}^3-1\right)N_{\mathrm{d}}N\right)\right) \\ &= O\left(mNc_{\max}^3\left(N^2 + \frac{\left(Nc_{\max}^3\right)^2(N_{\mathrm{d}})^2}{3} - Nc_{\max}^3 N_{\mathrm{d}}N\right)\right) \end{aligned} \tag{22}$$

Considering that $N - N_d\left(Nc_{\max}^3 - 1\right) > 0$ and $N_d > 0$, Eqs. (22) can be expressed as the following equation:

$$\begin{aligned} T &< O\left(mNc_{\max}^3\left(N^2 + \frac{\left(Nc_{\max}^3\right)^2\left(\frac{N}{Nc_{\max}^3-1}\right)^2}{3}\right)\right) = O\left(mNc_{\max}^3\left(N^2 + \frac{N^2}{3}\right)\right) \\ &= O\left(mNc_{\max}^3 N^2\right) \end{aligned} \tag{23}$$

The time complexity of NSGA-II is $O(mItr_{\max} Num^2)$, where Num and $Itr_{\max}$ are the number of population and the maximum iteration respectively. Under the same initial condition that $Nc_{\max}^3 = Itr_{\max}$ and $N = Num$, the time complexity of the modified MPIO is strictly less than the time complexity of NSGA-II.

5. UAV distributed flocking control algorithm with obstacle avoidance based on modified multi-objective pigeon-inspired optimization

By integrating the above models, a UAV distributed control algorithm based on the modified MPIO is proposed to coordinate a UAV swarm to pass through obstacle environments. The distributed UAV flocking control optimization frame designed in this paper is shown in Fig. 3. The UAV flocking with obstacle avoidance is decoupled into the behavior control of horizontal- and vertical- channels. The control algorithm in the vertical channel is described as the alignment with

the expected altitude and vertical velocity. In the horizontal channel, the designed control algorithm is composed of two parts: One is the flocking control based on the neighbors' states gained by distributed communication, and the other is the obstacle avoidance control based on the real-time perceived obstacle information. The outputs of the flocking control and obstacle avoidance control are integrated by a solver to obtain the control inputs of the Mach-hold autopilot and heading-hold autopilot. The modified MPIO is applied to assist each UAV in seeking an influence weight vector $\boldsymbol{w}_i$ which satisfies the conditions expressed in Eqs. (16) and (17). As a result, the stable and collision-free collective motion of a UAV swarm under obstacle environments will be guaranteed. The proposed UAV distributed flocking control algorithm under obstacle environments based on the modified MPIO is implemented as the following specific steps:

Step 1: The initial states of n UAVs, including position $\boldsymbol{P}^i = (x^i, y^i, h^i)$, horizontal airspeed $V_{\mathrm{xy}}{}^i$, yaw angle ψ^i, altitude rate λ_i, and influence weight vector $\boldsymbol{w}_i$ are generated, where $i = 1, 2, \ldots, n$ is the UAV index. Current simulation time $t = 0$.

Step 2: $i = 1$.

Step 3: Calculate desired flocking acceleration $\dot{\boldsymbol{vf}}^i$ by the self-propelled flocking model in Section 3.2. Calculate desired velocity for obstacle avoidance $\boldsymbol{vo}^i$ by the obstacle avoidance model in Section 3.3.

Step 4: Initialize the positions $\boldsymbol{X}^1$ of N pigeons within a n-dimensional search space. The upper bound $\boldsymbol{X_U}{}^{i'}$ and lower bound $\boldsymbol{X_L}{}^{i'}$ of the position $\boldsymbol{X}$ is 1 and 0 respectively, where i' is the index of pigeons. Calculate the objective function values $\boldsymbol{Cost}$ of all the pigeons by Eqs. (12)–(15). The current iteration $Nc = 1$.

Step 5: Conduct **Steps 1–3** in Section 4. It's important to note that only the two objective function values that correspond to soft constrains (i.e., $\boldsymbol{Cost}^1$ and $\boldsymbol{Cost}^2$) are involved in the crowded-comparison operation.

Step 6: $i' = 1$.

Step 7: Conduct **Step 4** in Section 4. Then calculate the objective function value $\boldsymbol{Cost}_{i'}$ of pigeon i' by Eqs. (12)–(15). If the position $\boldsymbol{X}_{i'}^{Nc}$ fails to meet the hard constrains (i.e., $\boldsymbol{Cost}_{i'}^3 = 1$ or $\boldsymbol{Cost}_{i'}^4 = 1$), the position $\boldsymbol{X}_{i'}^{Nc}$ of pigeon i' will be recreated within the search space. If $\boldsymbol{X}_{i'}^{Nc-1}$ dominates $\boldsymbol{X}_{i'}^{Nc}$ (i.e., $\boldsymbol{X}_{i'}^{Nc-1} \prec \boldsymbol{X}_{i'}^{Nc}$), $\boldsymbol{X}_{i'}^{Nc} = \boldsymbol{X}_{i'}^{Nc-1}$.

Step 8: If $i' = N$, go to **Step 9**, otherwise $i' = i' + 1$, and then go to **Step 7**.

Step 9: If current iteration $Nc \le Nc_{\max}^3$, delete the pigeons which satisfy that rank $N_{\mathrm{o}}^i \in [N - N_{\mathrm{d}}{+}1, N]$, the number of pigeons $N = N - N_{\mathrm{d}}$, and return to **Step 5**. Otherwise, rank the pigeons' positions $\boldsymbol{X}^{Nc}$ by Pareto sorting scheme.

Step 10: The influence weight vector $\boldsymbol{w}_i = \boldsymbol{X}_{i^*}^{Nc}$, and the index i^* is as the following equation:

$$i^* = \arg\min_j (\boldsymbol{Cost}_j^2) \tag{24}$$

where j is the index of pigeons in Pareto frontier S_1.

Step 11: Calculate control input $\boldsymbol{u}^i$ by the following equation:

$$\boldsymbol{u}_l^i = \begin{cases} \dot{\boldsymbol{vf}}_1^i + (\boldsymbol{vo}_1^i - \boldsymbol{v}_1^i) \\ \dot{\boldsymbol{vf}}_2^i + (\boldsymbol{vo}_2^i - \boldsymbol{v}_2^i) \\ \dot{\boldsymbol{vf}}_3^i \end{cases} \tag{25}$$

If the control input $\left|u_l^i\right| < u_{\mathrm{lim}}$, $\boldsymbol{u}_l^i = 0$, where $l = 1, 2, 3$ and u_{lim} is dead zone threshold of control inputs. Calculate the control inputs of UAV i's autopilots $(V_{\mathrm{xy\_c}}^i, \psi_{\mathrm{c}}^i, h_{\mathrm{c}}^i)$ by the following equation:

$$\begin{cases} V_{\mathrm{xy\_c}}^i = \tau_{\mathrm{v}}(\boldsymbol{u}_1^i \cos\psi^i + \boldsymbol{u}_2^i \sin\psi^i) + V_{\mathrm{xy}}^i \\ \psi_{\mathrm{c}}^i = \frac{\tau_\psi}{V_{\mathrm{xy}}^i}(\boldsymbol{u}_2^i \cos\psi^i - \boldsymbol{u}_1^i \sin\psi^i) + \psi^i \\ h_{\mathrm{c}}^i = h^i + \frac{\tau_{\mathrm{h}}}{\tau_\lambda}\lambda + \tau_{\mathrm{h}}\boldsymbol{u}_3^i \end{cases} \tag{26}$$

If $\left|V_{\mathrm{xy\_c}}^i - \|\boldsymbol{ve}_{1,2}\|\right| < V_{\mathrm{xy\_c}}^{\mathrm{lim}}$, $V_{\mathrm{xy\_c}}^i = \|\boldsymbol{ve}_{1,2}\|$. If $\left|\psi_{\mathrm{c}}^i - \psi_{\mathrm{m}}\right| < \psi_{\mathrm{c}}^{\mathrm{lim}}$, $\psi_{\mathrm{c}}^i = \psi_{\mathrm{m}}$. $V_{\mathrm{xy\_c}}^{\mathrm{lim}}$ and $\psi_{\mathrm{c}}^{\mathrm{lim}}$ are the allowable control errors. Then Calculate the UAV states at next time by Eq. (3).

Step 12: If $i = n$, go to **Step 13**, otherwise $i = i + 1$, and then go to **Step 3**.

Step 13: If $t < T_{\max}$, $t = t + ts$ and then go to **Step 2**, where ts is the sampling time and $T_{\max}$ is the maximum simulation time.

Step 14: Output the states of n UAVs.

6. Comparative simulation results

To validate the feasibility and effectiveness of the proposed UAV distributed flocking control algorithm with obstacle avoidance based on the modified MPIO, five small UAVs (fixed or rotatory wings) are disposed to fly in a formation through a complicated scenario with obstacles. It is important to note that the proposed UAV distributed flocking control algorithm does not only apply to small UAVs.

The initial positions of the UAVs are shown in Table 1, and other initial states of the UAVs are as follows: The horizontal airspeed $V_{\mathrm{xy}}^i = 10$ m/s, yaw angle $\psi^i = 0$ rad, and altitude rate $\lambda^i = 0$ m/s. In the motion space of the UAVs, six 100-meter-high obstacles exist with the parameters shown in Table 1, where $(x_{\mathrm{o}}^j, y_{\mathrm{o}}^j)$ and R_{o}^j are the position and radius of obstacle j

Table 1
Initial parameters of the UAVs and obstacles.

	i	$x^i(m)$	$y^i(m)$	$h^i(m)$
UAV	1	14.6929	107.3676	68.1682
	2	21.2809	116.6406	34.8423
	3	20.3911	113.6529	24.6351
	4	3.5699	108.9509	96.377
	5	10.2116	111.558	30.1431
	j	$x_o^j(m)$	$y_o^j(m)$	$R_o^j(m)$
Obstacle	1	120	120	5
	2	240	75	5
	3	350	40	5
	4	240	155	5
	5	360	110	5
	6	350	180	5

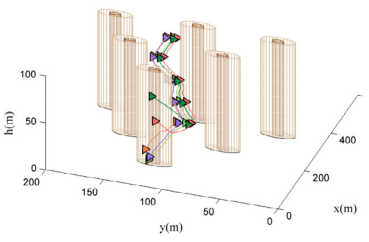
Table 2
Parameters of modified MPIO, MPIO and NSGA-II.

Algorithm	Variable	Description	Value
Modified MPIO/MPIO	N	Number of pigeons	58
	$Nc_{\max}^3$	Maximum iteration	20
	N_d	Reduced number of pigeons at each iteration	2
	$\boldsymbol{V_U}^i$	Upper bound of velocities	0.05
	$\boldsymbol{V_L}^i$	Lower bound of velocities	−0.05
	R	Map and compass factor	0.3
	f_t	Transition factor	3
	p_l	Percentage of general leaders	0.9
	e	Learning error	0.01
	s_l	Learning strength	2
NSGA-II	Num	Number of population	20
	$Itr_{\max}$	Maximum iteration	20
	S_{pool}	Size of a mating pool after tournament selection	25
	S_{tour}	Size of the tournament	2
	η_c	Crossover distribution index	20
	η_m	Mutation distribution index	20

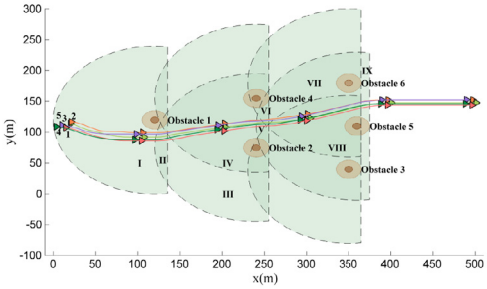
respectively. The parameters of the UAV model are as follows: The time constants of the three autopilots τ_V, τ_ψ, and (τ_h, τ_λ) are 1 s, 0.75 s, and (1 s, 0.3 s) respectively, the upper limits of the horizontal airspeed V_{xy_max}, altitude rate λ_{max}, and lateral overload n_{max} are 15 m/s, 5 m/s, and 10 g respectively, and the lower limits of the horizontal airspeed V_{xy_min} and altitude rate λ_{min} are 5 m/s and −5 m/s respectively.

The simulation is conducted according to the specific steps of the proposed UAV distributed flocking control algorithm with obstacle avoidance based on the modified MPIO in Section 5, where the maximum simulation time $T_{max} = 49.5$ s, the sampling time $ts = 0.5$ s and the initial influence weight vector $\boldsymbol{w}_i$=[1,1,1,1,1]. The parameters of the self-propelled flocking model are as follows: The expected altitude $h_e = 50$ m, the expected velocity of flocking flight $\boldsymbol{ve}$=[10,0,0]m/s, the horizontal communication range $R_{comm.}^1 = 20$ m, the maximum range of collision avoidance control $R_{lim}^1 = 2$ m, the factor of flocking geometry control $R_{desire} = 10$ m, the strength of the alignment K_{a_he}, K_{ve}, K_f, and K_{a_vn} are 30, 10, 0.1, and 0.1 respectively, and the strength of collision avoidance control $K_c = 100{,}000$. The parameters of the obstacle avoidance model are as follows: The horizontal perception range $R_{comm.}^2 = 105$ m, the field of view $\theta_{lim} = \frac{\pi}{2}$, and the minimum allowable distance between UAVs and obstacles $R_{lim}^2 = 10$m. The weight f_1 and f_2 in the second objective function are both 1. The parameters of the modified MPIO are shown in Table 2. The dead zone threshold of control inputs $u_{lim} = 0.25$ m/s^2. The allowable control errors $V_{xy\_c}^{lim}$ and ψ_c^{lim} are 0.25 m/s and 0.1 rad respectively.

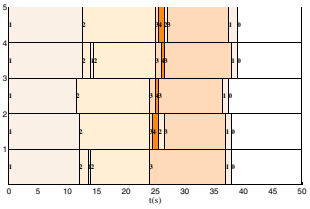
Fig. 4 depicts the detailed results of the 5 UAV distributed flocking flight under obstacle environments, in which (a)–(g) describe three-dimensional flocking trajectories, flocking trajectories in a top-down view, the number of obstacles in obstacle sets, altitudes, horizontal airspeeds, yaw angles, and altitude rates of the UAVs respectively. As shown in Fig. 4(a) and (b), the UAVs can fly safely through the obstacle environment in a stable formation without collision. As shown in Fig. 4(b), according to the detected obstacles, the regions passed by the UAVs could be divided into nine parts. The number of obstacles that can be identified in Regions I-IX are 1, 2, 1, 2, 3, 4, 2, 3, and 1 respectively, and the indexes of the obstacles are 1, (1, 2), 2, (2, 4), (2, 4, 5), (2, 4, 5, 6), (5, 6), (3, 5, 6), and 5 respectively. As shown in Fig. 4(c), the main regions among nine parts are Region I, IV, and VIII. At about 14.5 s, all the UAVs entered Region IV and will attempt to move through the gap between obstacles 2 and 4. At about 27 s, all the UAVs entered Region VIII and will attempt to move through the gap between obstacles 5 and 6. At about 39s, all the UAVs have passed all the obstacles. As shown in Fig. 4(d) and (g), the altitudes and altitude rates converge to expected altitude h_e and 0 m/s at about 10 s. As shown in Fig. 4(e) and (f), the



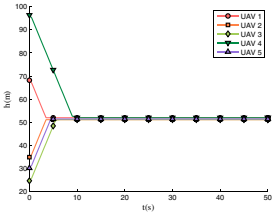
(a) Flocking trajectories



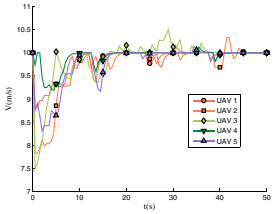
(b) Flocking trajectories in a top-down view



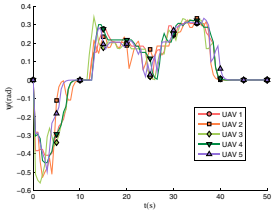
(c) The number of obstacles detected by each UAV



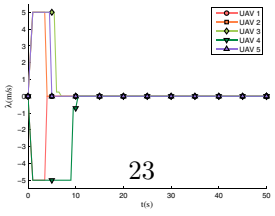
(d) Altitudes of UAVs



(e) Horizontal airspeeds of UAVs

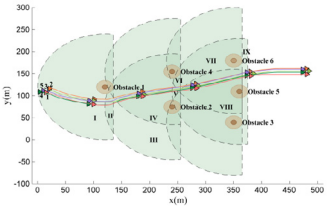


(f) Yaw angles of UAVs

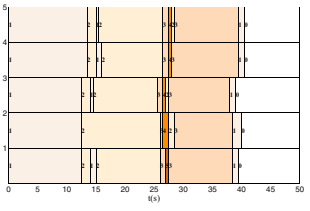


(g) Altitude rates of UAVs

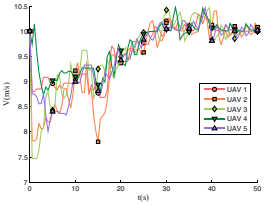
Fig. 4. UAV flocking flight based on the modified MPIO.



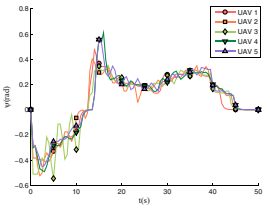
(a) Flocking trajectories in a top-down view (MPIO)



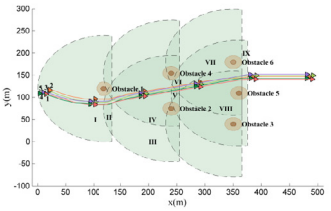
(b) The number of obstacles detected by each UAV (MPIO)



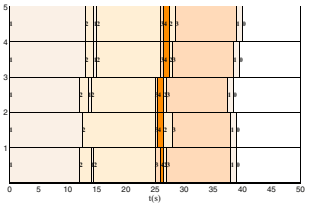
(c) Horizontal airspeeds of UAVs (MPIO)



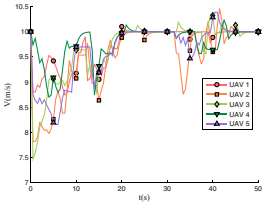
(d) Yaw angles of UAVs (MPIO)

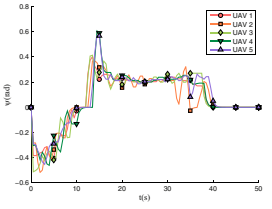


(e) Flocking trajectories in a top-down view (NSGA-II)



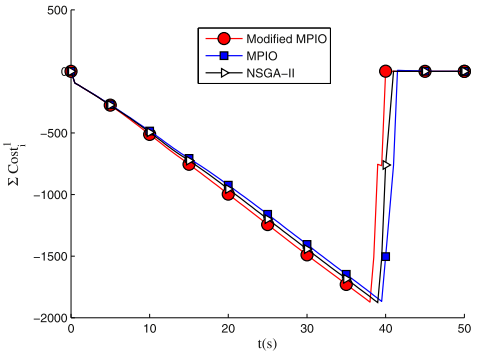
(f) The number of obstacles detected by each UAV (NSGA-II)



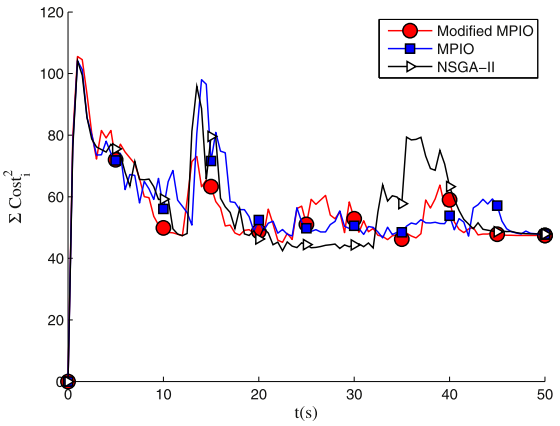


(g) Horizontal airspeeds of UAVs (NSGA-II) (h) Yaw angles of UAVs (NSGA-II)

Fig. 5. UAV flocking flight based on MPIO and NSGA-II.



(a) Sum of first objective function $\sum_i \boldsymbol{Cost}_i^1$



(b) Sum of second objective function $\sum_i \boldsymbol{Cost}_i^2$

Fig. 6. Objective functions of modified MPIO, MPIO and NSGA-II.

horizontal airspeeds and yaw angles converge to the corresponding expected states after fluctuations within the allowable range at about 46 s and 41.5 s respectively.

To further validate the superiority of the modified MPIO, the UAV flocking simulation based on the basic MPIO [21] and NSGA-II [6] was conducted with the same initial condition and parameter configuration. The parameters of the basic MPIO and NSGA-II are shown in Table 2. To guarantee the fairness and validity of comparisons, the same calculation times of objective functions and the same number of final solutions are required, which is expressed in the following equation:

$$\begin{cases} N \cdot Nc_{\max}^3 - \frac{Nc_{\max}^3\left(Nc_{\max}^3-1\right)}{2} N_d = Num \cdot Itr_{\max} \\ N - N_d\left(Nc_{\max}^3 - 1\right) = Num \end{cases} \tag{27}$$

Fig. 5 illustrates the detailed results of the distributed flocking flight of the UAVs based on the basic MPIO and NSGA-II where (a)–(d) describe the flocking trajectories in a top-down view, the number of obstacles in obstacle sets, horizontal airspeeds, and yaw angles of basic MPIO respectively, and (e)–(h) depict the flocking trajectories in a top-down view, the number of obstacles in obstacle sets, horizontal airspeeds, and yaw angles of NSGA-II respectively. It is important to note that the optimization algorithm is only applied to the horizontal control panel. As a result, the curves of altitudes and altitude rates for three optimization algorithms are identical. As shown in Fig. 5(a) and (e), after passing through Region I where only obstacle 1 can be detected, the UAVs based on the modified MPIO are always in front of the UAVs based on the other two optimization algorithms. As shown in Fig. 5(b) and (f), all the UAVs based on the basic MPIO and NSGA-II, entered Region IV at about 16 s and 15 s respectively, entered Region VIII at about 28.5 s, and passed all the obstacles at about 40.5 s and 40 s respectively. It's important to note that UAV 1 based on the basic MPIO was in a region where obstacles 2–6 can be detected from 27 to 27.5 s, which results in the short strip in Fig. 5(b) that represents that the number of obstacles detected by UAV 1 is 5. As shown in Fig. 5(c) and (g), the UAV horizontal airspeeds based on the basic MPIO did not converge, and the convergence time of the UAV horizontal airspeeds based on NSGA-II is 48s and is longer than the modified MPIO. As shown in Fig. 5(d) and (h), the convergence time of the UAV yaw angles based on the basic MPIO and

NSGA-III are about 46 s and 41.5 s respectively. The modified MPIO yielded better performance than the basic MPIO and had very competitive performance when compared with NSGA-II. The superiority of the modified MPIO is also corroborated in the curves of the cost functions corresponding to the soft constrains. The time corresponding to the lowest point of the sum of the first objective function $\sum_i \boldsymbol{Cost}_i^1$ is the time after which the first UAV will pass all the obstacles. The time, at which $\sum_i \boldsymbol{Cost}_i^1$ has stopped the rise, is the time before which the last UAV has passed all the obstacles. On the base of the roughly same flocking quality (as shown in Fig. 6(b)), the UAVs based on the modified MPIO are the fastest through all the obstacles (as shown in Fig. 6(a)). As shown in Fig. 6(b), after all the UAVs have passed all the obstacles, the modified MPIO exceeds the basic MPIO and NSGA-II by slightly good performance in flocking quality.

To sum up, the UAVs are capable of coordinating themselves to fly through complex obstacle environments in a stable and collision-free flocking formation by means of the proposed UAV distributed flocking control algorithm based on the modified MPIO. Compared with the basic MPIO and NSGA-II, the modified MPIO, which has better Pareto frontier under the condition of small population size and few iterations, is more suitable to solve the multi-objective optimization problem designed for UAV flocking and obstacle avoidance coordination control.

7. Conclusions

This paper designed a distributed optimization control frame to transform UAV flocking control to a multi-objective optimization problem. Besides, modified MPIO is presented to assist each UAV in solving the multi-objective optimization problem. By integrating the frame and modified MPIO, a UAV flocking control algorithm is proposed to coordinate UAVs to fly in a stale formation through complex environments, which is verified in a simulation case of small UAVs. The proposed UAV flocking control algorithm has the following advantages:

(1) The algorithm is proposed based on a distributed UAV flocking control optimization frame to assist a single UAV in solving a multi-objective optimization problem. Therefore, the distributed algorithm does not depend on the number of UAVs.
(2) In the algorithm, the neighbors' states are only used in the flocking control. Since the desired flocking velocity of the flocking control is a function of the positions and velocities of neighbors, the algorithm does not depend on other information of neighbors except positions and velocities.
(3) In the algorithm, the obstacle avoidance control is implemented by real-time obstacle perception. In another word, the algorithm does not depend on prior information of environments.

However, some deficiencies also exist in this paper: (1) The convergence analysis of the modified MPIO is lacking; (2) Since the control algorithm is distributed, deadlocks may still occur; (3) The simulation tests for emergency conditions and dynamic obstacles remain to be conducted. In the future, we will focus on the improvement of the distributed flocking control algorithm and the validation on UAV platforms.

References

[1] J. Alonso-Mora, E. Montijano, M. Schwager, D. Rus, Distributed multi-robot formation control among obstacles: a geometric and optimization approach with consensus, in: Proceedings of 2016 IEEE International Conference on Robotics and Automation, Stockholm, 2016, pp. 5356–5363, doi:10.1109/icra.2016.7487747.
[2] E. Besada-Portas, L. de la Torre, A. Moreno, J. Risco-Martín, On the performance comparison of multi-objective evolutionary UAV path planners, Inf. Sci. 238 (2013) 111–125, doi:10.1016/j.ins.2013.02.022.
[3] X. Bing, Y. Shen, A new disturbance attenuation control scheme for quadrotor unmanned aerial vehicles, IEEE Trans. Ind. Inf. 13 (6) (2017) 2922–2932, doi:10.1109/tii.2017.2682900.
[4] D. Biro, T. Guilf, H. Lipp, How the viewing of familiar landscapes prior to release allows pigeons to home faster: evidence from GPS tracking, J. Exp. Biol. 205 (2002) 3833–3844.
[5] D. Biro, T. Sasaki, S. Portugal, Bringing a time-depth perspective to collective animal behavior, Trends Ecol. Evol. 31 (7) (2016) 550–562, doi:10.1016/j.tree.2016.03.018.
[6] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, doi:10.1109/4235.996017.
[7] X. Dong, Q. Li, R. Wan, Z. Ren, Time-varying formation control for second-order swarm systems with switching directed topologies, Inf. Sci. 369 (2016) 1–13, doi:10.1016/j.ins.2016.05.043.
[8] H. Duan, P. Qiao, Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning, Int. J. Intell. Comput. Cybern. 7 (1) (2014) 24–37, doi:10.1108/ijicc-02-2014-0005.
[9] H. Duan, X. Wang, Echo state networks with orthogonal pigeon-inspired optimization for image restoration, IEEE Trans. Neural Netw. Learn. Syst. 27 (11) (2016) 2413–2425, doi:10.1109/tnnls.2015.2479117.
[10] D. Floreano, R. Wood, Science, technology and the future of small autonomous drones, Nature 521 (7553) (2015) 460–466, doi:10.1038/nature14542.
[11] Y. Gao, Q. Liu, X. Miao, J. Yang, Reverse k-nearest neighbor search in the presence of obstacles, Inf. Sci. 330 (2016) 274–292, doi:10.1016/j.ins.2015.10.022.
[12] T. Guilford, S. Robert, D. Biro, I. Rezek, Positional entropy during pigeon homing ii: navigational interpretation of bayesian latent state models, J. Theor. Biol. 227 (1) (2004) 25–38, doi:10.1016/j.jtbi.2003.07.003.
[13] M. Kumar, C. Guria, The elitist non-dominated sorting genetic algorithm with inheritance (I-NSGA-II) and its jumping gene adaptations for multi-objective optimization, Inf. Sci. 382–383 (2017) 15–37, doi:10.1016/j.ins.2016.12.003.
[14] Y. Ma, M. Hu, X. Yan, Multi-objective path planning for unmanned surface vehicle with currents effects, ISA Trans. 75 (2018) 137–156, doi:10.1016/j.isatra.2018.02.003.
[15] A. Mehrabian, K. Khorasani, Constrained distributed cooperative synchronization and reconfigurable control of heterogeneous networked Euler–Lagrange multi-agent systems, Inf. Sci. 370–371 (2016) 578–597, doi:10.1016/j.ins.2015.09.032.
[16] C. Mora, M. Davison, J. Wild, M. Walker, Magnetoreception and its trigeminal mediation in the homing pigeon, Nature 432 (7016) (2004) 508–511, doi:10.1038/nature03077.

[17] M. Nagy, Z. Ákos, D. Biro, T. Vicsek, Hierarchical group dynamics in pigeon flocks, Nature 464 (7290) (2010) 890–893, doi:10.1038/nature08891.
[18] M. Nagy, G. Vásárhelyi, B. Pettit, I. Roberts-Mariani, T. Vicsek, D. Biro, Context-dependent hierarchies in pigeons, Proc. Natl. Acad. Sci. 110 (32) (2013) 13049–13054, doi:10.1073/pnas.1305552110.
[19] A. Paul, P. Shill, New automatic fuzzy relational clustering algorithms using multi-objective NSGA-II, Inf. Sci. 448–449 (2018) 112–133, doi:10.1016/j.ins.2018.03.025.
[20] H. Qiu, H. Duan, Receding horizon control for multiple uav formation flight based on modified brain storm optimization, Nonlinear Dyn. 78 (3) (2014) 1973–1988, doi:10.1007/s11071-014-1579-7.
[21] H. Qiu, H. Duan, Multi-objective pigeon-inspired optimization for brushless direct current motor parameter design, Sci. China Technol. Sci. 58 (11) (2015) 1915–1923, doi:10.1007/s11431-015-5860-x.
[22] H. Qiu, H. Duan, Multiple uav distributed close formation control based on in-flight leadership hierarchies of pigeon flocks, Aerosp. Sci. Technol. 70 (2017) 471–486, doi:10.1016/j.ast.2017.08.030.
[23] H. Qiu, H. Duan, Pigeon interaction mode switch-based uav distributed flocking control under obstacle environments, ISA Trans. 71 (2017) 93–102, doi:10.1016/j.isatra.2017.06.016.
[24] H. Qiu, C. Wei, R. Dou, Z. Zhou, Fully autonomous flying: from collective motion in bird flocks to unmanned aerial vehicle autonomous swarms, Sci. China Inf. Sci. 58 (12) (2015) 1–3, doi:10.1007/s11432-015-5456-x.
[25] A. Sarabakha, N. Imanberdiyev, E. Kayacan, M. Khanesar, H. Hagras, Novel Levenberg-Marquardt based learning algorithm for unmanned aerial vehicles, Inf. Sci. 417 (2017) 361–380, doi:10.1016/j.ins.2017.07.020.
[26] M. Saska, Mav-swarms: unmanned aerial vehicles stabilized along a given path using onboard relative localization, in: Proceedings of International Conference on Unmanned Aircraft Systems, Denver, 2015, pp. 894–903, doi:10.1109/icuas.2015.7152376.
[27] C. Virágh, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, T. Vicsek, Flocking algorithm for autonomous flying robots, Bioinspiration Biomimetics 9 (2) (2014) 025012, doi:10.1088/1748-3182/9/2/025012.
[28] X. Wang, V. Yadav, S. Balakrishnan, Cooperative UAV formation flying with obstacle/collision avoidance, IEEE Trans. Control Syst. Technol. 15 (4) (2007) 672–679, doi:10.1109/tcst.2007.899191.
[29] A. Whiten, Operant study of sun altitude and pigeon navigation, Nature 237 (5355) (1972) 405–406, doi:10.1038/237405a0.
[30] B. Zhang, H. Duan, Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment, IEEE/ACM Trans. Comput. Biol. Bioinf. 14 (1) (2017) 97–107, doi:10.1109/tcbb.2015.2443789.