



Hybrid swarm intelligent algorithm for multi-UAV formation reconfiguration

Chenyang Gao¹ · Jianfeng Ma² · Teng Li² · Yulong Shen¹

Received: 30 November 2021 / Accepted: 8 October 2022 / Published online: 27 October 2022
© The Author(s) 2022

Abstract

Formation flight of unmanned aerial vehicles (UAVs) utilizes reconfiguration procedures to handle a variety of emergencies, such as collision avoidance, malfunctions, fuel savings, and member replacement. As UAVs have limited computing power and energy resources, it is necessary to optimize the control inputs to reduce the distance travelled by UAVs while reducing the computing costs during formation reconfiguration. In this paper, the problem of multi-UAV reconfiguration is decoupled into two stages: task assignment and control input optimization of UAVs. For a solution to the above problem, we propose an adaptive hybrid particle swarm optimization and differential evolution algorithm (AHPSODE) to optimize minimize the distance of the total movement and reduce the computing cost of formation reconfiguration. Based on the idea of receding horizon control (RHC) and the nonlinear model of multi-UAV formation reconfiguration, an RHC controller using AHPSODE is designed to optimize the control input of the UAV group to obtain the shortest movement distance, and this method can reduce the computation time. We use the CEC 2017 test suit to test the performance of our proposed AHPSODE algorithm, and simulate the AHPSODE-based RHC controller to manage formation reconfiguration. The results show that our proposed AHPSODE performed well in convergence and accuracy and the RHC controller is effective.

Keywords UAV · Formation reconfiguration · PSO · DE · Target assignment · Adaptive algorithm

Introduction

Multi-UAV formation flight [1] refers to the formation arrangement with precisely defined geometries and task assignment to meet the requirements of different tasks. Multi-UAV formation flights enable the onboard equipment to be distributed to various UAVs, and a complex task can be broken down into several relatively simple tasks performed by different individuals. These features can make multi-

UAV formation flights more stable, reliable, and redundant than single UAV flights. Consequently, multi-UAV formation flight has been widely used in various areas, including target strike, communications relay, electronic countermeasures, and battlefield assessment.

An multi-UAV formation flight consists of a full process that includes formation creation, maintenance, and reconfiguration, as well as the planning, organization, and execution of tasks [2]. In the absence of reconfiguration capabilities, multi-UAV formation flights would suffer from several issues, including collisions, inability to withstand external attacks, restricted scaling, and instabilities. Therefore, formation reconfiguration is of extreme significance, since the formation shape must be continuously reconfigured to cope with changes in group composition, environments, and tasks. This can enhance the robustness, reliability, and efficiency of a formation flight. However, UAVs are a kind of flight equipment with limited battery life and computing power. Therefore, the main challenge for multi-UAV formation reconfiguration is to design a controller that can optimize the control inputs to achieve the shortest total movement distance with fast computing speed and low computational overhead.

✉ Teng Li
tengli@xidian.edu.cn

Chenyang Gao
ghygao2008@gmail.com

Jianfeng Ma
jfma@mail.xidian.edu.cn

Yulong Shen
ylshen@mail.xidian.edu.cn

¹ The School of Cyber Engineering, Xidian University, Taibai Street, Xi'an 710071, Shanxi, China

² The School of Computer Science and Engineering, Xidian University, Taibai Street, Xi'an 710071, Shanxi, China

Table 1 Comparison of different methods for formation reconfiguration

Method	Time cost	Adaptiveness	Obstacles avoiding	Target assignment
HPSOGA [14]	✗	✗	✓	✗
PIO [15]	✗	✗	✓	✗
DE [16]	✓	✓	✓	✗
IDE [17]	✓	✓	✓	✗
AHPSODE	✓	✓	✓	✓

Research to reduce energy consumption and computing consumption in multi-UAV formation reconfiguration can be divided into three categories: path planning for reconfiguration, improvements to UAV controllers, and optimization of control inputs. In the path planning method, different path planning algorithms are improved and optimized to improve the performance of formation reconfiguration [3], such as A* [4], Dijkstra [5], Voronoi graph [6], artificial potential [7], and Dubin trajectory [8], etc. Despite their abilities to solve formation reconfiguration problems, these path planning methods cannot produce a moving trajectory with the shortest total distance, and are usually applied in a two-dimensional environment. The improvement of the control algorithm can improve the controller's performance and thus improve the efficiency of formation reconfiguration. The learning-based control algorithm has become a focus of research in this area, since it requires no complex mathematical model and is highly adaptable [9,10]. However, learning-based control algorithms always involve considerable uncertainty and mathematical complexity. The requirement of complexity and abundant computational resources are limitations to the use of learning-based control algorithms.

In comparison with improving the control algorithm, the method of optimizing the control input is more direct, effective, and scalable. In this method, the formation reconfiguration problem of UAV is transformed into a time-most control problem [11], in which swarm intelligent optimization methods can be an excellent choice to solve these optimization problems [12]. Wei et al. [13] propose a hybrid GA algorithm to solve the multi-UAV formation reconfiguration problem in 2D space. In the following work, Duan et al. [14] and Zhang [15] adapt CPTD (Control Parameters Time Discrete) method to construct an objective function and optimize the problem with different bionic algorithms. However, the CPTD method cannot solve the problem with too many control parameters for substantial computational overhead. Literature [16,17] adapt the rolling time-domain (RHC) method to decrease the computational overhead and improve the real-time performance of multi-UAV reconfiguration. Using this method, the global control problem can be divided into several local optimization problems with the same objective function. Consequently, by reducing the number of control parameters, the computational complexity and

computational time can be substantially reduced. However, these two methods do not consider the problem of target assignment of the formation final position.

As a solution to the problem of reducing computation overhead and determining a trajectory for formation reconfiguration with the shortest total moving distance, this paper splits the reconfiguration process into two parts: task assignment for the final positions and optimization of control inputs. We propose an adaptive hybrid particle swarm optimization and differential evolution algorithm (AHPSODE), which is designed to solve the target assignment problem in discrete space and create an RHC controller based on AHPSODE to optimize the control inputs for formation reconfiguration. Finally, to verify the validity of the proposed AHPSODE algorithm, we compared the performances of the AHPSODE with the standard PSO, standard DE, HPSOGA [14], PIO [15], and IDE [17] using the CEC 2017 test suit [18] and the experimental results show that the proposed AHPSODE have good performances. Furthermore, we simulate the AHPSODE-based RHC controller to conduct the multi-UAV reconfiguration. The simulation results show that the convergence time and search precision of the AHPSODE-based RHC are improved when achieving multi-UAV formation reconfiguration. Table 1 shows a comparison of some algorithms for solving multi-UAV formation reconfiguration.

The primary contributions of this paper are as follows:

1. We propose a new adaptive hybrid intelligent algorithm AHPSODE, which has a faster convergence speed and better ability to jump out local optimal than standard PSO and DE.
2. We first consider the task assignment problem of the final position of each UAV in a multi-UAV reconfiguration and solve this problem by a discrete AHPSODE. This method can reduce the total movement distance of the UAV formation.
3. We design an AHPSODE-based RHC controller to optimize multi-UAV formation reconfiguration control inputs' sequence, which can significantly reduce computation consumption, have higher search precision, and can obtain a better control input sequence.

4. We conduct a series of numerical simulation experiments to test the performance of the AHPSODE and to simulate the effectiveness of the AHPSODE-based RCH controller. The experiment results show that the algorithm and controller are effective.

The rest sections of this paper are organized as follows. In the section “[Related works](#)”, we make a summary of the related work. In the section “[Application scenario and algorithm principle](#)”, the application scenarios and some principles are introduced. In the section “[Approach overview](#)”, the approach is overviewed. Section “[Mathematical model](#)” presents the dynamics model of the quadrotor and the mathematical description of the target assignment. Section “[AHPSODE algorithm](#)” introduces the AHPSODE algorithm. In the section “[Design of formation reconfiguration controller](#)”, the implementation of the AHPSODE-based RHC controller is explained in detail. Section “[Experimental verification](#)” shows the experiment. Section “[Conclusion](#)” makes a summary of this paper.

Related works

The reconfiguration of multi-UAV formations is essentially a problem of formation control. In this section, we review research on formation control of multi-UAV reconfiguration from three perspectives: control structure, control mechanism, and control algorithm.

The control structure of UAV formation

The control approaches for formation reconfiguration can be classified into centralized and decentralized approaches based on their control structure. Each UAV is controlled by a single controller in the former method. The latter method involves each UAV acquiring information about its local environment from its neighbors to control itself.

Brandao [19] designed a multi-layer control scheme based on centralized formation control to guide a UAV group in desired trajectory tracking missions. After that, in [20], a centralized management structure of UAVs in the form of triangles was proposed to solve the intragroup collision problem for the UAV group. Obviously, centralized control is more efficient when there are fewer UAVs in the group. In formations with a large number of members, however, the decentralized formation control strategy offers the advantages of reducing the computing burden, improving computation efficiency, and increasing the real-time performance of the formation control. In [21], an adaptive full-order sliding mode control framework was proposed for robust multi-UAV synchronized formation motion under uncertainty. A robust 3D UAV formation flight can be achieved

with this control framework, as demonstrated in experiments. In a decentralized UAV formation flight, [22] proposed two control techniques for rejecting time-varying disturbances, PID, and integral sliding modes. In [23], a decentralized behavior-based control algorithm for UAV formation was designed. In this method, only the information of the relative position between neighboring group members is used.

The control mechanism of UAV formation

Control approaches for the formation reconfiguration can be categorized based on control mechanisms into leader–follower methods, virtual structures, and behavioral behavioral-based approach. Leader–follower approaches have been widely used in the control and management of UAV formations, and this approach has been shown to reduce the energy consumption of reconfiguring formations. Sajad Mousavi [24] designed a formation control algorithm based on the leader follower method to solve the multi-objective optimization problem of coalition formation in large-scale UAV networks. This method is inspired by quantum evolutionary algorithms and has better performance than NSGA-II. In [25], the focus is on the control problem of fixed-wing UAV formations in three-dimensional space. They discussed control law convergence using a Lyapunov stability tool, proposed an adaptive disturbance observer, and used an artificial potential method to solve the collision problem. Despite its versatility, the leader–follower method has very stringent requirements for security and safety on the leader UAV. Additionally, if the leader UAV suffers serious damage, the whole formation will be destroyed. Consequently, the visual structure method is proposed. Cezary Kownacki [26] combined the navigator algorithm and virtual structure method to improve the reliability and throughput of information sharing between UAVs in formation flight. A command generation method for UAV formations was proposed in [27] to solve the problem of wind disturbance and obstacle avoidance. Using this method, position control can be improved in the presence of wind and collisions can be avoided. Behavioral control defines several basic behaviors without the involvement of human operators, including following to avoid obstacles and composing formations, etc. Fan et al. [28] proposed an adaptive null-space-based behavioral (NSB) method to deal with the inadaptability of the NSB method in a formation control situation. The method can effectively improve formation performance. Based on the idea of method behavior-based formation control, Ferik [29] designed a behavioral adaptive fuzzy-based cluster space intelligent controller, which can well solve the formation maintenance problem. Although behavior-based approaches are very flexible in dealing with obstacle avoidance problems, they can interfere destructively in a formation group, and they can be difficult to analyze mathematically.

The control algorithms of UAV formation

From the perspective of the control algorithms, the existing UAV flight controllers can be classified into linear flight control algorithms, model-based nonlinear flight control algorithms, and learning-based control algorithms. To solve the UAV formation rearrangement problem, Mahfouz [30] designed a PID controller combined with the concept of backstepping. Vlahakis [31] improved the distributed LQR control methods to solve heterogeneous UAV formation stability problems. However, these linear control approaches easily suffer from performance degradation when the UAV departs from the nominal conditions. Some nonlinear flight control algorithms have been developed to overcome the limitations of linear algorithms based on the nonlinear model of quadrotor dynamics. Zhang [32] presented a backstepping approach that could be used to accomplish the reconfiguration of Multi-UAV formations and formation keeping. To improve the computational efficiency of the traditional MPC method, a Virtual Target Guidance (VTG)-based distributed MPC scheme for formation control of multi-UAV was proposed by CAI Zhihao [33], and the numerical simulation confirms the expectations of this proposed scheme. Learning-based control algorithms are usually independent of the UAV system model. However, these algorithms need a large amount of data to train the UAV flight control system. Examples include fuzzy logic (FL), neural networks (ANN), and reinforcement learning (RL). Quesada [9] proposed an algorithm based on a fuzzy logic approach capable of guiding a robot swarm to keep a leader–follower formation. A distributed optimal control method was also proposed using the idea of reinforcement learning to address the heterogeneous multi-UAV formation trajectory tracking problem. In [10]. This method can predict the optimal control input online without any knowledge of the dynamics of other follower UAVs. Although learning-based control algorithms do not depend on the mathematical model, these methods always involve considerable uncertainty and mathematical complexity. The requirement of complexity and abundant computational resources are limitations to the use of learning-based control algorithms.

Application scenario and algorithm principle

Application scenario

The multi-UAV formation reconfiguration problem can be described as follows: during a formation flight mission, the geometry shape of the UAV formation needs to change according to its environmental and mission requirements to avoid attacks or obstacles and improve the mission success rate. Meanwhile, the process of formation reconfig-

uration usually needs to consider some constraints, such as the computational overhead, energy consumption, obstacle avoidance, and air disturbance.

Figure 1 shows a simplified application scenario for multi-UAV formation flight and reconfiguration. In this example, both three UAVs are in landed mode at departure points A and B, respectively. Then, the procedure of multi-UAV formation reconfiguration can then be described as follows:

S1: The ground station receives the mission instructions from the operators and sends these instructions to corresponding UAVs landed in departure points A and B.

S2: After receiving the mission instructions, the UAVs will fly to the target area and complete the formation assembling.

S3: The assembled UAVs will form a formation first. Then, this UAV formation will take the corresponding flight strategy flying to the mission area to execute its task (the leader–follower method is adopted in Fig. 1).

S4: The UAV will perform tasks, such as traffic control, data collection, and communication assistance after entering the task area.

S5: After mission completion, all UAVs will assemble according to their mission requirements during the return process and maintain this flight formation.

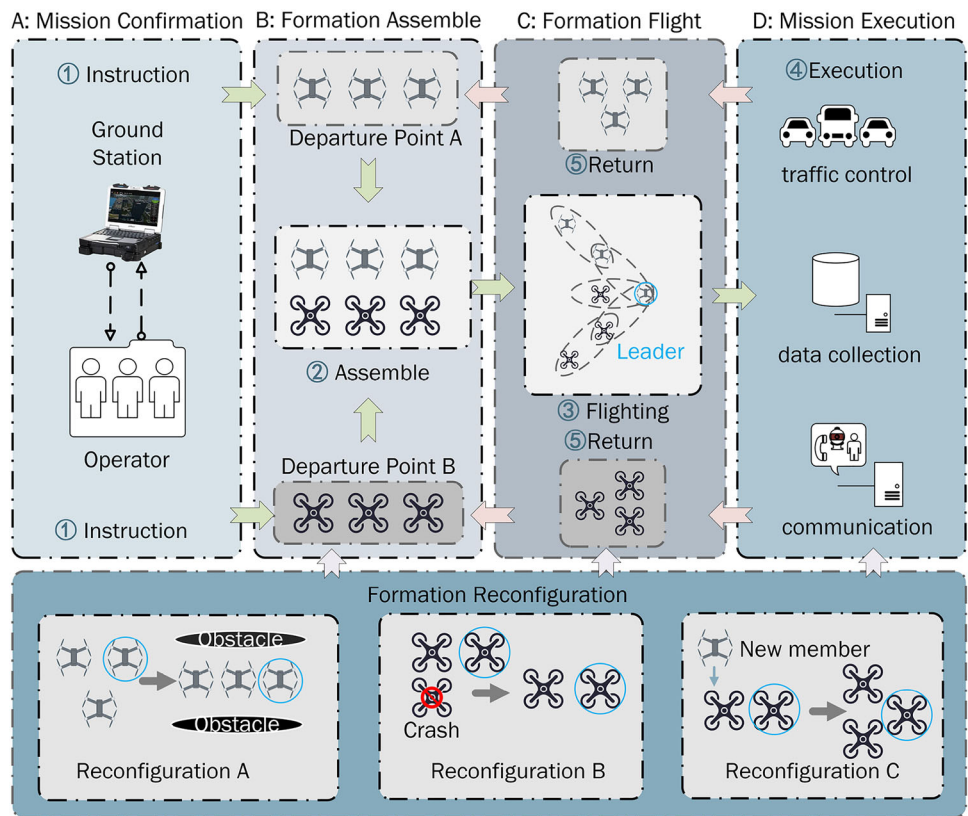
Due to the possibility of UAV malfunction or damage during the execution of the mission, and changes in the environment, the UAVs' flight formations need to be changed during each of the five flight sessions. Some cases that need formation reconfiguration, such as obstacle avoidance, adjustment of formation members, and UAV malfunction, are shown at the bottom of Fig. 1.

In this paper, the AHPSODE-based RHC method is used to optimize UAVs' control input, so the reconfiguration of multi-UAV formations can take the shortest time possible. Furthermore, all constraints can be met, such as preventing the UAVs from collisions and maintaining their communication distance.

Principle of PSO algorithm

PSO algorithm was first proposed by Kennedy and Eberhart in 1995 from the perspective of social and cognitive behavior [34] and has been widely applied in the engineering field. In the standard PSO algorithm, each particle will fly in the solution space and calculate the value of the current position's objective function to determine the next move according to the optimal position of the individual particle history and the whole group's optimal position. Ultimately, the swarm, as a whole, the same as a flock of birds working together to find food, is likely to move in the direction that can obtain the best value of the target function. In the continuous space coordinate system, the mathematical description of the PSO algorithm is as follows:

Fig. 1 Application scenario for multi-UAV formation reconfiguration



A group composed of m particles moves in a D -dimensional search space, and each particle has a certain speed. When the group is searching for the best position, each particle will consider the historical best position that it has searched for and the historical best position of the other particles in the group. Then, each particle will update its position and speed.

For each particle $x_{id}(t)$, its velocity and position update formula in d dimension ($1 \leq d \leq D$) is shown as below

$$\begin{aligned}
 v_{id}(t + 1) &= wv_{id}(t) + c_1 \cdot r_1 \cdot (p_{id}(t) - x_{id}(t)) \\
 &\quad + c_2 \cdot r_2 \cdot (p_{gd}(t) - x_{id}(t)) \\
 x_{id}(t + 1) &= x_{id}(t) + v_{id}(t),
 \end{aligned}
 \tag{1}$$

where $p_{id}(t)$ is individual optimal position, $p_{gd}(t)$ is global optimal position, and the acceleration constant c_1 and c_2 are two non-negative values. These constants can help each particle self-summarize and learn from excellent individuals to approach its own historical best position and the group's global best position. c_1 and c_2 are usually equal to 2, and r_1 r_2 are two random number ranging from $[0,1]$.

Principle of DE algorithm

Differential evolutionary algorithm (DE) is an evolutionary algorithm based on population differences proposed by Storn and Price [35] to solve Chebyshev polynomials. The basic

idea of DE is to use the differences between the individuals of the current population to recombine the population to obtain the intermediate population and then use the competition between the parent and the offspring to obtain the next-generation population.

In the process of mutation, three individuals are chosen randomly from the population, symbolized as x_{r1}, x_{r2}, x_{r3} , and these three individuals must be different from those selected from x_i . We can then define the mutated individual v_i as follows:

$$v_i = x_{r1} + F \times (x_{r2} - x_{r3}),
 \tag{2}$$

where F is the magnification factor and usually valued from $[0, 2]$. Figure 2 is the mutation process diagram: Using mutated individuals v_i and target individual x_i , the reorganization process can be expressed as follows:

$$u_{ij} = \begin{cases} v_{ij}, & \text{if } randb \leq CR \text{ or } j = randr \\ x_{ij}, & \text{otherwise} \end{cases}
 \tag{3}$$

where $randb$ is a random number between $[0,1]$, and CR is the crossover factor used to control the crossover probability. $randr$ is a random integer between $[1,D]$, which can be used to guarantee that at least one-dimension value of reconstituted individual u_{ji} comes from the mutant individ-

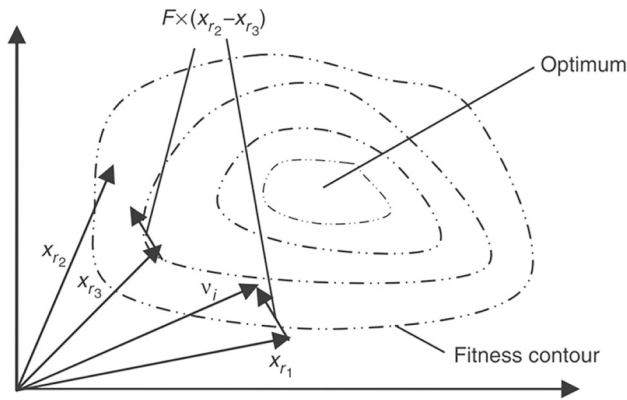


Fig. 2 Mutation process of DE

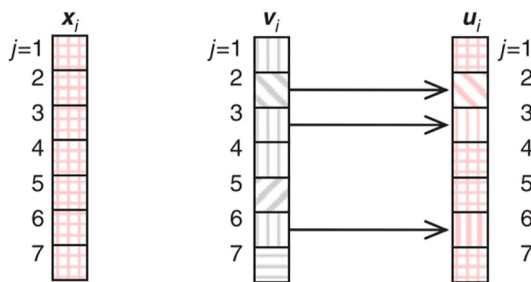


Fig. 3 Recombination process of DE

ual v_{ji} . Figure 3 shows the detail process of crossover. This recombination process can be described by Fig. 3.

During selection, the offspring is determined by the fitness values of the initial individual x_i and the mutant recombination individual u_i . Equation (4) describes this selection principle

$$x_i(t + 1) = \begin{cases} u_i, & \text{if } f(u_i) < f(x_i) \\ x_i, & \text{else.} \end{cases} \quad (4)$$

Receding horizon control (RHC)

Receding horizon control (RHC) is also called model predictive control (MPC). At each discrete moment of sampling, the system’s current state is used as the initial condition to solve a finite-time-domain open-loop optimal control problem and obtain the optimal control sequence.

For each time-domain, the cost function of local optimization is the same as global optimization. Figure 4 shows a simple generation process of control inputs’ sequences through RHC. In Fig. 4, each time horizon will calculate p inputs, but only the first several m inputs can be applied to the actual system.

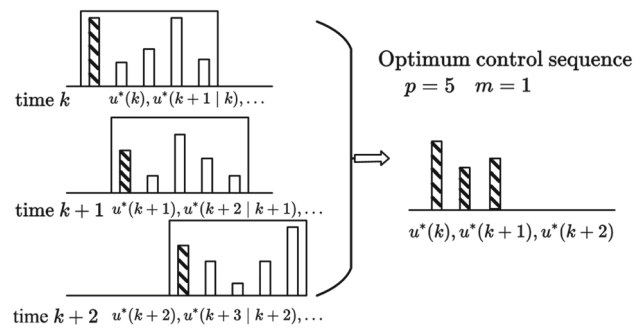


Fig. 4 Generation process of control input sequences through RHC

Approach overview

During multi-UAV formation reconfiguration, each UAV’s target formation must be determined, which is equivalent to a task assignment problem. A successful assignment can reduce the overall moving distance and time cost of reconfiguring a formation. At the same time, the optimization of the controller’s control inputs at every moment is also important for reconfiguring a multi-UAV formation. In this paper, the AHPSODE algorithm is proposed to optimize the target assignment problem. The control inputs of each UAV can also be optimized using an AHPSODE-based RHC controller.

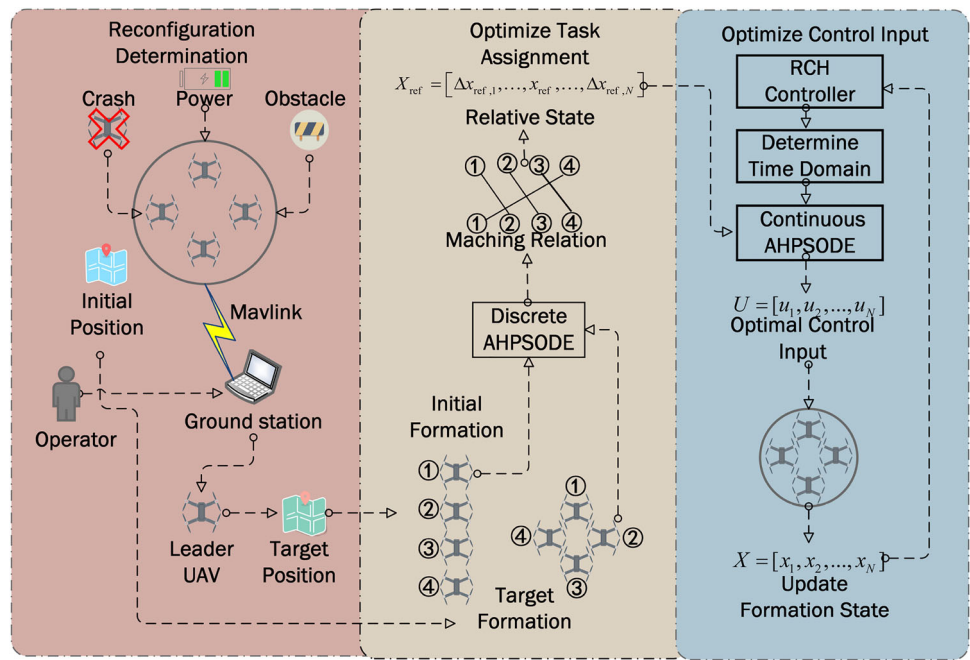
Figure 5 shows a complete process for multi-UAV formation reconfiguration. The process is mainly composed of three parts: reconfiguration determination, task assignment optimizing, and control input optimizing.

Reconfiguration determination In multi-UAV formation flight, formation reconfiguration is usually required when there occur changes in the external environment or mission instruction. These situations include obstacle avoidance, malfunction of formation members, adjustment of formation members, and reducing energy consumption. The UAVs can acquire external environment information through the onboard sensors to judge whether the current formation needs to reconfigure. Each UAV will transmit its current status parameters to the ground station through the MavLink protocol. Then, the ground station will determine the shape of the formation and analyze the UAV group’s status information to select the leader UAV.

Optimizing task assignment After determining the UAV formation’s initial state and the target position, the target assignment problem in the reconfiguration process will be solved by the discretized AHPSODE algorithm. In the target assignment’s optimization process, the UAV group’s matching relationship is optimized with the goal of the shortest total length. Then, the UAVs’ relative coordinates, relative to the leader UAV, X_{ref} can be determined, where X_{ref} plays an essential role in optimizing the cost function.

Optimizing control input This paper uses receding horizon control (RHC) to divide the global control problem into

Fig. 5 Process of multi-UAV formation reconfiguration



several local optimization problems with the same objective function. The complexity of computation and the amount of computation can be reduced by reducing the number of optimized control inputs. Additionally, as the reconfiguration of multi-UAV formations is a complex nonlinear problem, the RHC control method uses existing optimization techniques to generate the optimal sequence of control inputs. Therefore, the AHPSONE algorithm proposed in this paper is adopted to quickly calculate the optimal control input sequence in each time-domain. After optimization, the control input sequence is sent to the UAV group for execution. Afterward, each UAV will provide feedback on its current status to the RHC controller. The formation reconfiguration will be terminated if the current UAV group’s status information meets the termination condition. Otherwise, roll the time-domain and repeat the above optimization process until the termination condition is satisfied.

Mathematical model

In this paper, we focus on the optimization of control inputs for multi-UAV formation reconfiguration in 3D space. Therefore, in this section, we analyze the dynamic model of the quadrotor and determine the cost function. Finally, the target assignment problem is mathematically explained.

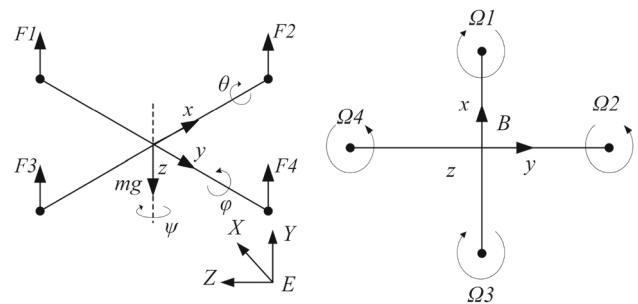


Fig. 6 Force model of UAV

The dynamic model of UAV

The quadrotor in the experimental environment consists of four fixed-pitch rotors mounted at the four ends of a simple cross frame, as shown in Fig. 6.

where ϕ, θ, ψ are, respectively, roll angle, pitch angle, and yaw angle, which are the main control variables for attitude control. F_1, F_2, F_3, F_4 are the forces generated from these four motors. mg is the weight of the quadrotor, and $oxyz$ is the body-fixed coordinate system. Transform matrix from body-fixed coordinate ($oxyz$) to ground-fixed coordinate ($OXYZ$) is shown in Eq. (5)

$$R(\phi, \theta, \psi) = \begin{bmatrix} \cos\theta \cos\psi & \sin\phi \sin\theta \sin\psi - \cos\phi \sin\psi & \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ \cos\theta \sin\psi & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\ -\sin\theta & \cos\theta \sin\phi & \cos\theta \cos\phi \end{bmatrix}. \tag{5}$$

The lifting forces F_1, F_2, F_3 , and F_4 in body-fixed coordinate ($oxyz$) are as follows:

$$F_B = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}. \tag{6}$$

Then, we need to transform the lifting forces F_B from the body-fixed coordinate to the ground-fixed coordinate F_G by rotation matrix R

$$F_G = \begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix} = R(\phi, \theta, \psi) \cdot F_B \tag{7}$$

$$= \left(\sum_{i=1}^4 F_i \right) \begin{bmatrix} \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ \cos \theta \cos \phi \end{bmatrix}.$$

According to the motion equations and Eq. (7), we can deduce the following equation:

$$\begin{cases} \ddot{x} = F_x/m = (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)U_1/m \\ \ddot{y} = F_y/m = (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)U_1/m \\ \ddot{z} = F_z/m = (\cos \theta \cos \phi)U_1/m - g. \end{cases} \tag{8}$$

Then, we can deduce attitude change from Newton–Euler formula

$$\sum M = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = I\dot{\omega} - \omega \times (I\omega), \tag{9}$$

where $I = \text{diag}(I_\phi, I_\theta, I_\psi)$ are, respectively, equivalent moment of inertia about the roll axis, pitch axis, and yaw axis. $\omega = (\dot{\theta}, \dot{\phi}, \dot{\psi})^T$ is the torque, which consists of the torque M_i generated by lifting forces. Then, according to Eq. (9), we can deduce Eq. (10)

$$\sum M = \begin{bmatrix} I_x \ddot{\phi} + \dot{\theta} \dot{\psi} (I_y - I_z) \\ I_y \ddot{\theta} + \dot{\phi} \dot{\psi} (I_z - I_x) \\ I_z \ddot{\psi} \end{bmatrix}. \tag{10}$$

According to the mechanic principle, we can get Eq. (11)

$$\sum M = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} l(f_4 - f_2) \\ l(f_3 - f_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix}. \tag{11}$$

Here, we define the input variables of the UAV as follows:

$$\begin{cases} U_1 = F_1 + F_2 + F_3 + F_4 \\ \quad = K_t (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = F_4 - F_2 = K_t (\Omega_4^2 - \Omega_2^2) \\ U_3 = F_3 - F_1 = K_t (\Omega_3^2 - \Omega_1^2) \\ U_4 = -M_1 + M_2 - M_3 + M_4 \\ \quad = K_d (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2), \end{cases} \tag{12}$$

where U_1, U_2, U_3 , and U_4 are the control variables of vertical speed, roll angle, pitch angle, and yaw angle, respectively. Ω_i is the rotating speed of the i th rotor. K_t is the coefficient of lifting forces; K_d is the coefficient of torques.

Considering Eqs. (7)–(12), the mathematical model of a quadrotor can be finally presented as Eq. (13)

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{U_1}{m} (c_\phi s_\theta c_\psi + s_\phi s_\psi) \\ \frac{U_1}{m} (c_\phi s_\theta s_\psi - s_\phi c_\psi) \\ -g + \frac{U_1}{m} c_\phi c_\theta \\ \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\psi} + \frac{IU_2}{I_x} \\ \frac{U_z}{I_y} \dot{\phi} \dot{\psi} + \frac{IU_3}{I_y} \\ \frac{U_4}{I_z} \end{bmatrix}. \tag{13}$$

To simplify the problem, this paper only considers UAVs' the position in three-dimensional space, and does not consider the state change of attitude angle. Therefore, the control inputs of i th UAV can be defined as $u_i = [\theta_i, \psi_i, \phi_i, v_{Bi}]^T$ (v_{Bi} is the velocity of UAV i th under body coordinates), and the state variable can be expressed as $x_i = [x_i, y_i, z_i]^T$. Therefore, the dynamic equation of the i th UAV in a UAV formation can be described as follows:

$$\dot{x}_i(t) = f(x_i(t), U_i(t)), i = 1, 2, \dots, N. \tag{14}$$

According to Eq. (14), we can know that if control input u_i and initial state x_0 are provided, the state of i th UAV at any time t can be obtained as follows:

$$x_i(t) = x_0 + \int_0^t f(x_i(\tau), u_i(\tau)) d\tau. \tag{15}$$

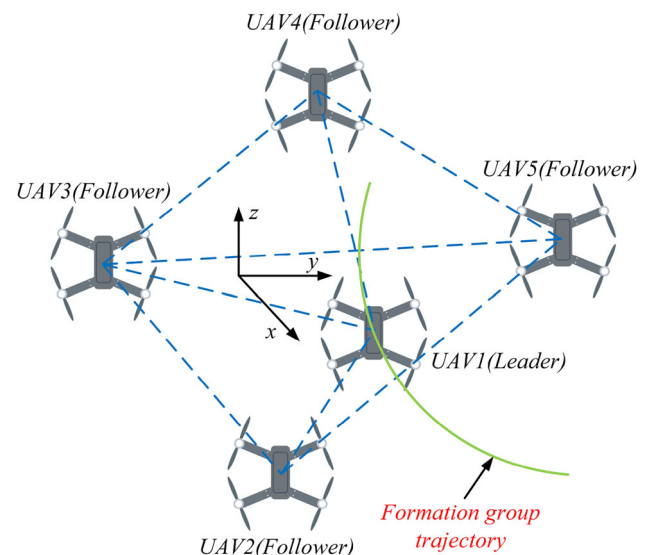


Fig. 7 Structure of leader–follower method

This paper adopts the centralized control structure and the leader–follower method as our control mechanism (Fig. 7). The state vector of the leader UAV is expressed as x_L . Then, the relative state of the i th UAV in the formation relative to the leader can be expressed as $X_{ref,i} = x_i - x_L$, and we set the initial time as $t = 0$ and the termination time as $t = T$ during a formation reconfiguration process. The formation reconfiguration process in this paper can be regarded as a control optimization problem. The goal of this optimization problem is to find a set of continuous control inputs $U = (u_1, u_2, \dots, u_n)$ that can achieve formation reconfiguration and make the cost function minimum. The quadratic form of the cost function is as follows:

$$\begin{aligned} \min J(U) &= \sum_{i=1}^N (x_{ref,i}(t) - (x_i(T|u_i) - x_L(T|u_L)))^T \\ &\quad Q (x_{ref,i}(t) - (x_i(T|u_i) - x_L(T|u_L))) \\ \text{st} \quad &x_i(t|u_i) = x_i(0) + \int_0^t f(x_i(\tau), u_i(\tau)) d\tau \\ &U_{\min} < U < U_{\max}, \end{aligned} \tag{16}$$

where $X_{ref}(t) = [x_{ref,1}(t), \dots, x_{ref,N}(t)]$ is the vector of the relative states of the all UAV in the formation relative to the leader UAV at time t . $x_i(t|u_i)$ is the state of i th UAV at time t under the control inputs sequential $u_i(t)$. For each UAV, given initial state $x_i(t_0)$, the state $x_i(t)$ will be deduced by u_i . The control input u_i will be limited according to the performance of UAV and $Q = \text{diag}(q_1, q_2, q_3, q_4)$ is a positive definite matrix.

$d_{ij}(t)$, $i, j = 1, \dots, N$ represents the distance between any two UAVs in the formation of UAVs, and its calculation formula is as follows:

$$\begin{aligned} d_{ij}(t) &= \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 - (z_i(t) - z_j(t))^2}. \end{aligned} \tag{17}$$

To prevent collisions within the formation during the formation of reconfiguration, it is required that $d_{ij}(t)$ must be greater than the safe distance D_{safe} in the period of reconfiguration

$$d_{ij}(t) \geq D_{safe}, \forall t \in [0, T], \quad \forall_{i \neq j}, j \in 1, 2, \dots, N. \tag{18}$$

At the same time, to guarantee the communication between UAVs in formation, we need $d_{ij}(t)$ less than the communication distance D_{com} between UAVs during the whole time-period of reconfiguration

$$d_{ij}(t) \leq D_{com}, \forall t \in [0, T], \quad \forall_{i \neq j}, j \in 1, 2, \dots, N. \tag{19}$$

By combining the Eqs. (16–19), a new cost function can be obtained as follows:

$$\begin{aligned} \min J_{\text{extend}}(U) &= J(U) + \omega \cdot \int_0^T \sum_{i \neq j} [\max(0, D_{\text{safe}} - d_{i,j}(t)) \\ &\quad + \max(0, d_{i,j}(t) - D_{\text{com}})] dt, \end{aligned} \tag{20}$$

where ω is the penalty coefficient of distance, which is usually a large constant.

Mathematical description of target assignment

Target assignment of multi-UAV formation reconfiguration problem is equivalent to a task assign problem (TAP). Stone first proposes TPA in 1997 [36], which assigns more than one processor in a distributed system and minimizes the execution cost of task allocation and communication cost under the constrained system resource condition. In this paper, we attempt to solve a target assignment problem similar to TAP, in which the main objective is to allocate the initial position of each UAV to a position in the target formation. This assignment needs to minimize the total moving distance during formation reconstruction without considering resource constraints.

The process of multi-UAVs formation reconfiguration can be divided into two stage. In the first stage, the geometry shape of target formation and its position are determined according to changes in the external environment or mission requirements. This stage also deals with the tasks assignment problem mentioned in this article The second stage is to control the movement of the UAV group to realize the change of the formation. This stage is related to the control parameter optimization problem mentioned in this paper, that is, the motion planning.

We decouple the formation reconfiguration problem into these two stages for two main reasons:

1. For multi-UAV formation reconfiguration, besides using the shortest path as the optimization goal for motion planning, it is also necessary to determine the moving targets of each UAV in the initial formation when determining the geometries of the target formation. Here is a straightforward example. Suppose there are three UAVs, each of which is positioned as follows: a[0 0 0], b[10 0 0], and c[10 0 0]. The positions of target formation are A[0 0 20] B[-10 0 10] C[10 0 10], and we expect that the initial line formation will transform into the target V formation. In this study, UAV heterogeneity is not taken into account, so both a-A b-B c-C and a-C b-A c-B can achieve the same target formation. Obviously, the total movement distance of the former assignment is smaller than that of the lat-

Table 2 A set of target assignment

Initial formation UAV number	1	2	3	4	5
Target formation UAV position number	1	3	5	4	2

ter. Therefore, when the number of UAVs increases, it is necessary to separate the task assignment problem from the entire formation reconfiguration. This will reduce the total movement distance of the overall formation reconstruction.

- The process of optimizing task assignment requires some computing time and consumption, and considering the real-time requirements in the process of formation reconfiguration, we hope to reduce the time costing for optimizing control parameters as much as possible. As a result, the task assignment is decoupled from the overall reconstruction process and handled before the formation change, which can reduce the complexity of the problem.

This paper uses natural numbers encoding each group of assignment results in the target assignment problem to solve the multi-UAV formation reconstruction problem. To solve the multi-UAV formation reconstruction problem, this paper uses natural numbers to encode each group of assignment results. Suppose a UAV formation consists of N UAVs, and then, there will be $n!$ targets assigned. Let x_i be the i th target assignment result of the group, then x_i can be represented as follows:

$$x_i = [x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{in}], \quad (21)$$

$$x_{ij} \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, n\},$$

where x_{ij} represent the target position of j th UAV in the i th, which is determined by i th target assignment result.

Suppose there is a flight team made up of five UAVs, and this team needs to change its formation. In the initial formation, the serial number of UAVs is set to 1–5, and the UAV positions in the target formation are also set to 1–5. Then, a set of target assignment results x_i can be expressed as $x_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}]$, $x_{ij} \in [1, 2, \dots, 5]$. Where x_{ij} represents the assigned position in target formation of UAV j and 5 is the total number of UAVs in the flight formation. If $x_i = [1, 3, 5, 4, 2]$, then the result of the target assignment can be represented as Table 2:

To minimize the total moving distance of the formation reconfiguration by target assignment, distance overhead matrix D is constructed as

$$D = \begin{bmatrix} D_{11} & D_{12} & \cdots & D_{1n} \\ D_{21} & D_{22} & \cdots & D_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ D_{n1} & D_{n2} & \cdots & D_{nn} \end{bmatrix}, \quad (22)$$

where D_{jm} represents the distance between UAV j in initial formation and position m in target formation, $j = 1, 2, \dots, n$, $m = 1, 2, \dots, n$. Furthermore, according to the target allocation vector x_i and the distance cost matrix D_i , we can obtain the total distance Total D

$$\text{Total } D = \sum_{j=1}^n D_{jx_{ij}}. \quad (23)$$

Therefore, the x_i that can minimize the value of Total D is the optimal target allocation result.

AHPSODE algorithm

The PSO algorithm is known for its low computational overhead and fast convergence speed, but it is easy to fall into the local optimum. Through the crossover process of the DE algorithm, it is possible to increase the amount of search space by encouraging information exchange between individuals. In addition, the mutation process can increase the diversity of the population and improve the algorithm's ability to escape from the local minimum. Therefore, the AHPSODE proposed in this paper combines the advantage of the PSO algorithm and the DE algorithm to find an optimal solution.

First, the PSO algorithm is used to update the position of the initial population. To reduce the complexity of the algorithm, this process runs 50% max iterations. Then, we set the population classification threshold as 30%, sort the particles' fitness values, and divide the population into superior and inferior group. The top 30% of excellent particles are superior groups, and the rest are inferior group. The individuals in inferior groups will be reinitialized to increase the diversity of the whole population. Then merge inferior groups and inferior group to generate a new population as the initial population for DE. In the early stage of this hybrid algorithm, the PSO algorithm can quickly get an approximate optimal solution for the advantage of fast convergence and the low computational cost. Meanwhile, the mutation operation of the DE algorithm can effectively prevent the algorithm from local optimization and improve the search precision.

Generally, w in the PSO has a great impact on the search ability of the algorithm. The w in the standard PSO algorithm is a constant value. As a result, the standard PSO cannot dynamically adjust its search capability during different periods of iteration and is easy to falling into local optimality. In the iteration process of the PSO algorithm, the distribution of the whole population gradually changes from a dispersive state to a centralized state. However, excessive concentration may result in the algorithm falling into local optimization. As a consequence, we need to enhance the global search abil-

ities of the PSO when the distribution of the population is too concentrated. And we need to enhance the local search abilities when the distribution is dispersed. In other words, as population density increases, the inertial weight w needs to adjust dynamically. This article improves the inertia weight w of the standard particle swarm, which integrates the distance information between particles into w , so that w can be adjusted according to the distribution of the particle swarm adaptively.

First, define the average distance between the i th particle and other $N - 1$ particles as d_i

$$d_i = \frac{1}{N - 1} \sum_{j=1, j \neq i}^N \sqrt{\sum_{k=1}^D (x_{ik} - x_{jk})^2}, \tag{24}$$

where N is the population size, and D is the dimension. Then, define the adaptation factor σ as follows:

$$\delta = \frac{d_g - d_{\min}}{d_{\max} - d_{\min}}. \tag{25}$$

In Eq. (25), d_g represents the average distance between of global optimal point. d_{\max} and d_{\min} represent the maximum and minimum average distances in the population, respectively. Then, according to the properties of adaptive factor σ , we construct an adaptive inertia weight $W(\delta)$

$$W(\delta) = \frac{1}{1 + 1.5e^{-2.6\delta}}. \tag{26}$$

According to Eq. (26), the value of $W(\delta)$ is positively correlated with σ , which means that the larger σ is, the more massive $W(\delta)$ is, and vice versa. At the beginning of the iteration of PSO, the overall distribution of particles is scattered, which means that the σ is relatively large. Therefore, the PSO algorithm with the adaptive factor σ has better global search capabilities at the early stage of the iteration. As the algorithm converges, the position of the particle swarm is relatively concentrated, so that the σ will decrease, which can lead to the decrease of the $W(\delta)$ and increase the local search ability of the algorithm. Besides, the coefficients of $W(\delta)$ are used to adjust the transformation range, and the current coefficient can determine that the range of $W(\delta)$ is [0.4, 0.9].

In DE, the crossover probability CR has a great impact on the population’s diversity. When the value of CR is large, there will be a big probability of mutation for the population, which can guarantee the population’s diversity, and the algorithm is more likely to find the global optimal solution. When the value of CR is small, the population’s information is easier to be inherited, which is conducive to the algorithm for stable search in the current solution space. Therefore, this

Algorithm 1 AHPSODE

```

Require: Population: $M$ ; Dimension: $D$ ; Generation: $T$ ; Learning
Factors: $c1, c2$ ; Scaling Factors: $F$ 
Ensure: Run max iteration
1: for  $i = 1$  to  $M$  do
2:   for  $i = 1$  to  $M$  do
3:      $x_{ij}(t) = Minx_{ij} + \text{rand}(0, 1) \cdot (Maxx_{ij} - Minx_{ij})$ 
4:      $v_{ij}(t) = Minv_{ij} + \text{rand}(0, 1) \cdot (Maxv_{ij} - Minv_{ij})$ 
5:   end for
6:    $P_i = x_i$  (Pbest)
7: end for
8:  $P_g = \min(P_i)$  (Gbest)
9:  $t \leftarrow 1$ 
10: while ( $|f(Gbest)| \geq \epsilon$ ) or ( $t \leq 0.5 * T$ ) do
11:   for  $i = 1$  to  $M$  do
12:     (Adaptive factor of PSO and DE)
13:     Calculate  $d_i, \sigma, \omega(\sigma), CR(\sigma)$ 
14:     for  $j = 1$  to  $D$  do
15:        $v_{ij}(t + 1) = \omega(\sigma)v_{ij}(t) + c_1 \cdot r_1 \cdot (P_{ij}(t) - x_{ij}(t))$ 
16:          $+ c_2 \cdot r_2 \cdot (P_{gij}(t) - x_{ij}(t))$ 
17:        $x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t)$ 
18:     end for
19:     if  $f(x_i(t + 1)) < P_i$  then
20:        $P_i \leftarrow x_i(t + 1)$ 
21:     if  $f(P_i) < f(P_g)$  then
22:        $P_g \leftarrow P_i$ 
23:     end if
24:   end if
25: end for
26:    $t \leftarrow t + 1$ 
27: end while
28: Sort  $f(x_i)$ 
29: SuperiorGroup  $\leftarrow$  Top50%
30: InferiorGroup  $\leftarrow$  Rest
31: ReinitializetheInferiorGroup
32: PopulationM  $\leftarrow$  SuperiorGroup  $\cap$  InferiorGroup
33: while ( $|f(Gbest)| \geq \epsilon$ ) and ( $t \leq T$ ) do
34:   for  $j = 1$  to  $D$  do
35:      $v_{i,j}(t) = \text{Mutation}(x_{i,j}(t))$ 
36:      $u_{i,j}(t) = \text{Crossover}(x_{i,j}(t), v_{i,j}(t))$ 
37:     if  $f(u_i(t)) < f(x_i(t))$  then
38:        $x_i(t) \leftarrow u_i(t)$ 
39:     if  $f(x_i(t)) < f(P_i)$  then
40:        $P_i \leftarrow x_i(t)$ 
41:     if  $f(x_i(t)) < f(P_g)$  then
42:        $P_g \leftarrow x_i(t)$ 
43:     end if
44:   end if
45: end if
46: end for
47:    $t \leftarrow t + 1$ ;
48: end while
49: returnthebestsolution  $P_g$  :

```

article uses the distance adaptation factor mentioned above to improve CR

$$CR(\delta) = \frac{1}{1 + e^{-2.2\delta}}. \tag{27}$$

According to Eq. (27), then range of $CR(\delta)$ is [0.5, 0.9], and the improved DE algorithm will adaptively adjust the cross

probability according to the concentration degree of population, which can improve the search accuracy of the standard DE algorithm. Algorithm (1) is the pseudo of AHPSODE.

Design of formation reconfiguration controller

RHC-based formation reconfiguration

RHC divides the global control problem into several local optimization problems, and these local optimization problems have the same optimization goal as the global optimization. In the k th sampling period, the state equation of the i th UAV in the UAV formation can be described as follows:

$$x_i(k+1) = x_i(k) + \int_{kT}^{(k+1)T} f(x_i(k), u_i(k)) dt. \tag{28}$$

The constraints for the $u_i(k)$ and $x_i(k)$ are as follows:

$$\begin{aligned} u_{\min} &\leq u_i(k) \leq u_{\max} \\ x_{\min} &\leq x_i(k) \leq x_{\max}, \end{aligned} \tag{29}$$

where $x_i(k) = [x_i(k), y_i(k), z_i(k)] \in R^3$ represent the state of i th UAV in the k th sampling period. $u_i(k) = [\phi_i(k), \theta_i(k), \psi_i(k), v_{b_i}(k)]^T \in R^4$ represents the corresponding control input, which will keep stable until the next sampling period, and T represents the sampling period's time interval. At time horizon k , the RHC controller can calculate each UAV's control input sequence in the current time horizon and the future p predicted time horizon through the UAV formation's current state and constraint conditions. These control input sequences can be expressed as $u_i(k|k), u_i(k+1|k), \dots, u_i(k+p-1|k)$. Then, the corresponding states in the next p time horizons k can be represented as $x_i(k+1|k), x_i(k+2|k), \dots, x_i(k+p|k)$.

Through Eq. (16) and Eqs. (28)–(30), we can obtain the quadratic cost by the following cost function at the k th time horizon:

$$\begin{aligned} \min J(k) &= \sum_{j=1}^p \sum_{i=1}^N (x_{\text{ref},i} - (x_i(k+j|k) - x_L(k+j|k)))^T \\ &\quad \times Q (x_{\text{ref},i} - (x_i(k+j|k) - x_L(k+j|k))) \\ \text{st } x_i(k+1|k) &= x_i(k) + \int_0^T f(x_i(k), u_i(k|k)) dt \\ x_i(k+j+1|k) &= x_i(k+j|k) \\ &\quad + \int_0^T f(x_i(k+j|k), u_i(k+j|k)) dt \\ u_{\min} &\leq u_i(k+j|k) \leq u_{\max}. \end{aligned} \tag{30}$$

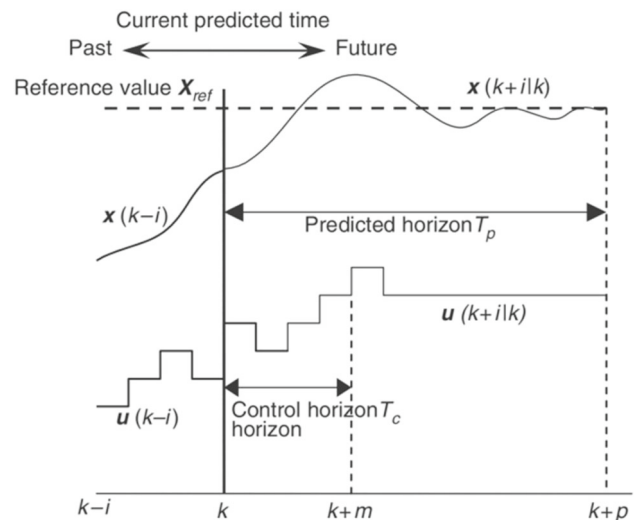


Fig. 8 Process of RHC control

Finding a solution that minimizes the fitness function at a given time horizon k , we can obtain the optimal control input sequence $u_i^*(k+j-1|k), j = 1, 2, \dots, p$, where we apply the preceding m control input $u^*(k|k), u^*(k+1|k), \dots, u^*(k+m-1|k), (0 \leq m \leq p)$ to the corresponding UAVs in the current and following $m-1$ time horizons, respectively. The final state of multi-UAV formation can then be approximated using the RHC method. The process of an RHC method can be described in Fig. 8.

Although the RHC method can divide a global control problem into a series of local optimization problems to reduce the computational complexity and the time overhead, the multi-UAV formation reconfiguration problem is essentially a constrained nonlinear problem. Therefore, it is hard to solve this problem using traditional optimization methods. Therefore, how to calculate the control law of RHC is a crucial technology. Aiming at the above problems, an adaptive hybrid intelligent algorithm AHPSODE is proposed to optimize the target assignment problem and the control inputs' sequence of RHC.

AHPSODE-based RHC controller

In this paper, we use the AHPSODE algorithm to optimize each UAV's control input to realize the reconfiguration of multi-UAV formation. Suppose there is a UAV flight formation composed of N UAVs, and the size of predicted time horizon is p . Then, the control input of i th UAV in the formation at time horizon k can be expressed as $u_i(k+j-1|k) = [SF_i, \phi_i, \theta_i, \psi_i]^T, i = 1, \dots, N, j = 1, \dots, p$. Since UAV's control input has four parameters, the dimensionality of the search space of the AHPSODE algorithm can be obtained as $D = 4 * N * p$. Furthermore, the particle x_i in

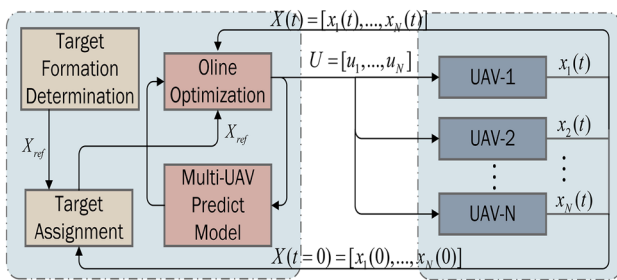


Fig. 9 Schematic of AHPSODE-based RHC controller

the AHPSODE algorithm at time horizon k can be expressed as

$$x_i(k) = [u_1(k|k), u_2(k|k), \dots, u_N(k|k), \dots, u_1(k + 1|k), u_2(k + 1|k), \dots, u_N(k + 1|k), \dots, u_1(k + p - 1|k), u_2(k + p - 1|k), \dots, u_N(k + p - 1|k)]. \quad (31)$$

The schematic of the AHPSODE-based RHC controller is shown in Fig. 9. Here, $X_{ref} = [\Delta x_{ref,1}, \dots, x_{ref}, \dots, \Delta x_{ref,N}]$ is the relative state vector between the leader UAV and other UAVs. x_{ref} is the reference state of the Leader UAV x_L , and $\Delta x_{ref,i}$ represents the i th UAV's expected state relative to x_L .

The target formation determination module is used to determine the leader UAV's number and the state of the target formation. All of the above information will send to the target assignment module. Subsequently, the target assignment module optimizes each UAV's moving targets through the information transmitted by the target formation determination module and the current state of the UAV group. Then, the controller will update the relative state.

After acquiring the X_{ref} , the online optimization module will calculate the optimal control sequence U in the current time horizon and send the control input sequence to the corresponding UAV for execution. After each UAV completing the execution of control input, the UAV formation will feedback the current status to the online optimization module and repeat the above steps until the UAV team's states satisfy the termination condition. Considering the complexity of computation and the controller's timeliness, we adopted the first-order predictive control strategy in this controller. Therefore, we set $p = m = 1$.

Experimental verification

Performance analysis of AHPSODE

We employed a set of 29 benchmark functions for explaining the performance of the proposed AHPSODE. The current well-known single objective optimization test suites contain

Table 3 Parameter settings for different algorithms

Algorithm	Parameter setting
SPSO	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2.0$
SDE	$CR = 0.5 \sim 0.9, F = 1.2$
APSO	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2.0$
ADE	$CR = 0.5 \sim 0.9, F = 1.2$
AHPSODE	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2, F = 1.2$
HPSOGA[14]	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2, P_c = 0.9, P_m = 0.05$
IDE[17]	$CR = 0.5 \sim 0.9, F = 1.2$
PIO[15]	$N1 = 0.9 * N, N2 = 0.1 * N, R = 0.003$

CEC 2005, 2013, 2014, and 2017. We chose the CEC 2017 as the test suite for this experiment. In the CEC 2017, there are two unimodal functions (F1, F3), seven simple multimodal functions (F4–F10), ten expanded hybrid functions (F11–20), and ten composition functions (F21–F30). Please refer to [18] for more details about CEC 2017.

We compared the performance of AHPSODE with eight relevant algorithms, namely: SPSO, SDE, APSO, ADE, HPSOGA [14], IDE [17], and PIO [15]. The APSO algorithm and ADE algorithm are derived from SPSO and SDE using distance adaptive factors. By comparing APSO and ADE to PSO and DE, the effectiveness of the proposed distance adaptive factor can be validated. On this premise, the comparison of the AHPSODE algorithm with ADE and APSO can further demonstrate the performance of the AHPSODE algorithm. HPSOGA, IDE, and PIO algorithms are used in similar work to solve the formation reconfiguration problem, and hence, these three algorithms are also chosen to compare with AHPSODE.

Experiment setting

To have a fair comparison, all the experiments were coded in MATLAB language and were implemented on Windows 10 Pro-64 bit of a PC with 32 GB of RAM and Inter(R) Core(TM)i7-9700 CPU @ 3.0GHz.

We set the population N as 100 and the maximum generation G as 3000 for all algorithms. On the premise of retaining the theme idea of the algorithm, we will ensure that the parameters of the tested algorithm are consistent. For example, the ADE and the IDE have different improvement strategies for parameter CR, which means that our goal is to keep their range of changes for CR consistent. The detailed parameter settings are shown in Table 3.

Comparison for AHPSODE with other algorithms

In this part, the performance results on CEC 2017 at 30-D and 50-D are presented. In CEC 2017, there are a total of 30 test

Table 4 Statistics of mean value for AHPSODE and compared algorithms on CEC 2017

Dimension	Function type	Mean value							
		SPSO	SDE	APSO	ADE	AHPSODE	HPSOGA	IDE	SPIO
30	Unimodal functions					2			
	Multimodal functions	3				3	1		
	Hybrid functions	3	2			4	1		
	Composition functions					4	4		2
	SUM	6	2	0	0	13	6	0	2
50	Unimodal functions					2			
	Multimodal functions					7			
	Hybrid functions			2		7			1
	Composition functions					6	4		
	SUM	0	0	2	0	22	4	0	1

Table 5 Statistics of standard deviation for AHPSODE and compared algorithms on CEC 2017

Dimension	Function type	Standard deviation							
		SPSO	SDE	APSO	ADE	AHPSODE	HPSOGA	IDE	SPIO
30	Unimodal functions					1			1
	Multimodal functions		3						4
	Hybrid functions	3	2			2	1		2
	Hybrid functions	1	2		2	1	2		2
	SUM	4	7		2	4	3	0	9
50	Unimodal functions					2			
	Multimodal functions		2			2			3
	Hybrid functions	1	1	1	1	3			3
	Composition functions		4		2	2	1		1
	SUM		4		2	2	1	0	1

functions (F2 has been excluded, because it shows unstable behavior, especially for higher dimensions). The maximum generation G for each algorithm is set as 3000. Each algorithm needs to run 15 times for a same test function at each dimension and calculate the mean value and standard deviation.

Tables 4 and 5 present the statistical data for mean value and standard deviation at 30-D and 50-D, respectively. The Function Type is the collection of all test functions of the same type (e.g., Unimodal Function = F1, F3). The values in the table represent the number of times that the algorithm finds the optimal value compared with other algorithms for the same function type and dimension. In this situation, a blank cell indicates that the algorithm did not get a good result on the test function when compared to other algorithms. SUM represents the sum optimum statistics for a given algorithm over all test functions. The detailed results are shown in the Appendix, Tables 14 and 15, respectively.

Table 4 shows that the APSO algorithm and the ADE algorithm are not more effective than other algorithms when handling 30-dimensional problems, but the SPSO and SDE

algorithms in F7 F8 F9 F12 F13 F18 and F14 F15 obtain better solutions. However, in 30 dimensions, SDE can only get a better solution than ADE when solving F10 F14, so distance adaptive factor does have some improvement for ADE to deal with low-dimensional problems. In 50 dimensions, the mean value of APSO on the 25 test functions is better than those of SPSO. On the other hand, SPSO and SDE did not find the best solution for 50-dimensional problems, whereas APSO found a better solution for F13 and F14. Consequently, an enhanced SPSO and SDE incorporating distance adaptation factor do not have excellent performance in low dimensions. However, the distance factor has a much better effect when dealing with high dimension problems. In Table 4, we can also find that AHPSODE has achieved optimal solutions for all F1–F10 problems with 30 dimensions. This illustrates that the AHP algorithm is effective in solving low-dimensional unimodal and multimodal problems. In 50 dimensions, there are 22 optimal solutions to 30 test functions using AHPSODE. This shows that the AHP algorithm has a significant performance improvement compared to SPSO and SDE, especially when solving high-

dimensional complex problems, the algorithm can avoid falling into local optimums and improve global search capability. Additionally, AHPSODE provides better solutions than HPSOGA IDE and SPIO algorithms when dealing with 30-dimensional problems. However, AHPSODE performs similarly to HPSOGA when solving complex problems of high dimension.

Table 5 shows that although the AHPSODE algorithm proposed in this paper has good global search ability, the standard deviation is lower than that of SDE for 30-dimensional problems. This means that the AHPSODE algorithm has a small probability calculating a less desirable solution when dealing with low-dimensional problems. SPIO is significantly more stable in 30 dimensions than SDE. However, the algorithm's accuracy is poor.

The boxplots for experiment data are shown in Appendix Figs. 19, 20, 21, 22. With the boxplots, we can intuitively visualize the calculation results of each algorithm on the specified test function. Data deviation is especially intuitive with the box plot. We also plot the convergence curves of all algorithms in different dimensions for different test functions as shown in Appendix Figs. 23, 24, 25, 26. These figures show that at the early stage of convergence, AHPSODE has a faster convergence speed than SDE. At the later stages of the algorithm, AHPSODE has a better search accuracy than SPSO. The above data analysis demonstrates that by integrating the SPSO and SDE algorithms and including the distance adaptive factor, the AHPSODE performs much better than the other algorithms in global search.

Run-time complexity

In this subsection, we use the widely known running-time complexity criterion proposed in CEC 2017 to measure the efficiency of AHPSODE with SPSO, SDE, APSO, and ADE. In Table 6, the running-time complexity results of above algorithms on the CEC 2017 at 10-D, 30-D, and 50-D are calculated.

From Table 6, we can see that the SDE consumes more running-time than SPSO, and the running-time of the SPSO increases faster with the improvement of problem dimensions than that of DE. We also find that the running-time of the ADE and APSO is larger than that of the standard SPSO and SDE. The reason is that APSO and ADE need to calculate the distance adaptive factor at each iteration. However, this distance adaptive can improve the calculation accuracy of the SPSO and SDE. Due to the fact that the AHPSODE algorithm integrates the distance adaptive factor, the algorithm's running-time is also bigger than that of the SPSO and SDE. However, the running-time of the AHPSODE is substantially lower than that of ADE for the algorithm combines the properties of SPSO and SDE, and the running-time will not increase much as the problem dimension expands.

Table 6 Running-time complexity for SPSO, SDE, APSO, ADE, and AHPSODE

Algorithm	Dimension	T0	T1	T2	(T2–T1)/T0
SPSO	10	0.024	0.116	0.1458	1.24
	30		0.515	0.5825	2.81
	50		1.169	1.3678	8.28
SDE	10	0.024	0.116	0.4745	14.94
	30		0.515	1.0312	21.51
	50		1.169	1.8201	27.13
APSO	10	0.024	0.116	1.6087	62.20
	30		0.515	3.4752	123.34
	50		1.169	5.6537	186.86
ADE	10	0.024	0.116	2.079	81.86
	30		0.515	4.459	164.35
	50		1.169	6.823	235.58
AHPSODE	10	0.024	0.116	1.87	73.15
	30		0.515	3.78	136.18
	50		1.169	5.75	190.91

Meanwhile, the algorithm's computational precision is superior than that of other algorithms. In Table 4, this finding has been verified.

Simulation of multi-UAV reconfiguration

In this paper, the problem of multi-UAV formation reconfiguration is decoupled into two parts. In the first part, we construct the target position determination problem in the formation change process as a task assignment problem. Then, we solve it through the AHPSODE algorithm to determine the target position of each UAV. In the second part, the control input sequence for each UAV is generated by the AHPSODE-based RHC controller to reform the target formation with the shortest moving distance. Therefore, this section will simulate the above two decoupled processes to evaluate the effectiveness of AHPSODE-based RHC controller for multi-UAV reconfiguration.

All the simulation experiments were coded in MATLAB language and were implemented on Windows 10 Pro-64 bit of a PC with 32 GB of RAM and Inter(R) Core(TM)i7-9700 CPU @ 3.0GHz.

Simulation for target assignment

In this subsection, we use AHPSODE to solve the discrete task assignment problem in multi-UAV formation reconfiguration and test its performance and effectiveness. We first set up a virtual formation reconfiguration scene. In this scene, a group of 10 UAVs are randomly generated limited to a spherical sphere with (0, 0, 0) as its center and 30m as its radius. The target is to control these UAVs to form a circle forma-

Table 7 States of UAV group

UAV number	Initial states	Target states
1	(−8.7, −5.3, 28.2)	(24.3, 17.6, 0)
2	(15.7, 16.4, −19.6)	(9.3, 28.5, 0)
3	(−23.2, −3.4, 18.7)	(−9.3, 28.5, 0)
4	(−24.3, −16.5, 6)	(−24.3, 17.6, 0)
5	(0.2, −27.9, −11)	(−30, 0, 0)
6	(2.2, −11.5, −27.6)	(−24.3, −17.6, 0)
7	(−4.8, −6.6, 28.9)	(−9.3, −28.5, 0)
8	(−0.1, 0.5, −30)	(9.3, −28.5, 0)
9	(19.9, −7.9, 21)	(24.3, −17.6, 0)
10	(2.5, 0.7, 29.9)	(30, 0, 0)

Table 8 Parameter settings for different algorithms

Algorithm	Parameter setting
SPSO	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2.0$
SDE	$CR = 0.5 \sim 0.9$
APSO	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2.0$
ADE	$CR = 0.5 \sim 0.9, F = 1.2$
AHPSODE	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2, F = 1.2$
HPSOGA[14]	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2, P_c = 0.9, P_m = 0.05$
IDE[17]	$CR = 0.5 \sim 0.9, F = 1.2$
PIO[15]	$N1 = 0.9 * N, N2 = 0.1 * N, R = 0.003$

tion. The coordinates of the initial state and the target state are shown in Table 7.

Since the above task assignment problem is discrete, we discretize AHPSODE to solve this problem with the goal of minimizing the total moving distance. The SPSO, SDE, APSO, ADE, IDE, AHPSOGA, and SPIO algorithms are also compared to the AHPSODE algorithm in the present experiment. The parameters of the above algorithms are shown in Table 8. According to the experiment results, we find the optimal solution $g_{best} = [4 \ 1 \ 5 \ 6 \ 8 \ 9 \ 7 \ 3 \ 10 \ 2]$. Figure 10 shows the initial position and target position of each UAV, and the solid lines in different colors correspond to g_{best} . Figure 11 shows the convergence curves of SPSO, SDE, APSO, ADE, IDE, PIO, HPSOGA, and AHPSODE in solving the target assignment problem of above UAV formation reconfiguration problem. The x -coordinate is the number of iterations, and the y -coordinate is the fitness value (the total moving distance).

As shown in Fig. 11a, the APSO algorithm has better convergence accuracy than the SPSO algorithm, since it is easier to jump out of the local optimum. In comparison to SDE, the ADE algorithm has a faster convergence speed, mainly because the distance adaptive factor adjusts the mutation probability of the algorithm, helping it obtain better global search ability in the early stages of algorithm iteration. Based on the aforementioned results, the distance adaptive factor

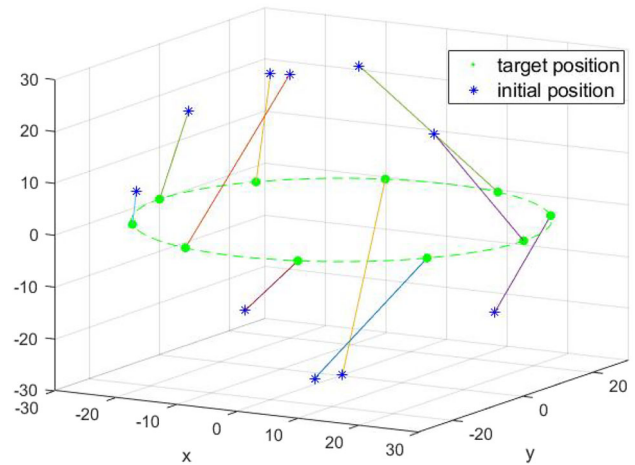


Fig. 10 The optimal task assignment result of 10 UAVs in 3D space

proposed in this paper can effectively improve the accuracy of the algorithm in the later iteration of the discrete PSO and discrete DE when dealing with the problem of target assignment. At the same time, according to the convergence curve, it can be found that the convergence performance of AHPSODE algorithm is better than that of ADE and APSO.

As illustrated in Fig. 11b, the convergence speed of the AHPSODE algorithm in the early stages of the iteration is slightly lower than that of SPIO, but the global convergence speed is higher than SPIO, IDE, and HPSOGA. The AHPSODE also has better convergence accuracy than the above algorithm. As a result, the AHPSODE algorithm can not only find the optimal assignment result in this problem of multi-UAV formation reconstruction, but also provide a relatively good performance.

Simulation of AHPSODE-based RHC controller

In this part, we construct a virtual scene for multi-UAV formation reconfiguration. To solve this formation problem, the AHPSODE-based RHC controller is simulated in our experiment. Additionally, we compare AHPSODE-based RHC controller with similar methods PIO, HPSOGA, and IDE on the basis of RHC controllers. The parameters of the above test algorithm are displayed in the Table 9.

Simulation setting In our experiment scenario, there is a UAV group of 9 UAVs with the same specifications. We numbered these UAVs as 1–9, and the initial state of each UAV is shown in Table 10. Among this UAV group, UAV₅ is selected as leader UAV. The goal is to transform this initial formation into a circular formation, and we use our proposed controller to generate optimized control inputs for each UAV. The states of the target circular formation are also shown in Table 10.

Furthermore, we add several constraints on the control inputs in our experiment to ensure that the optimized control

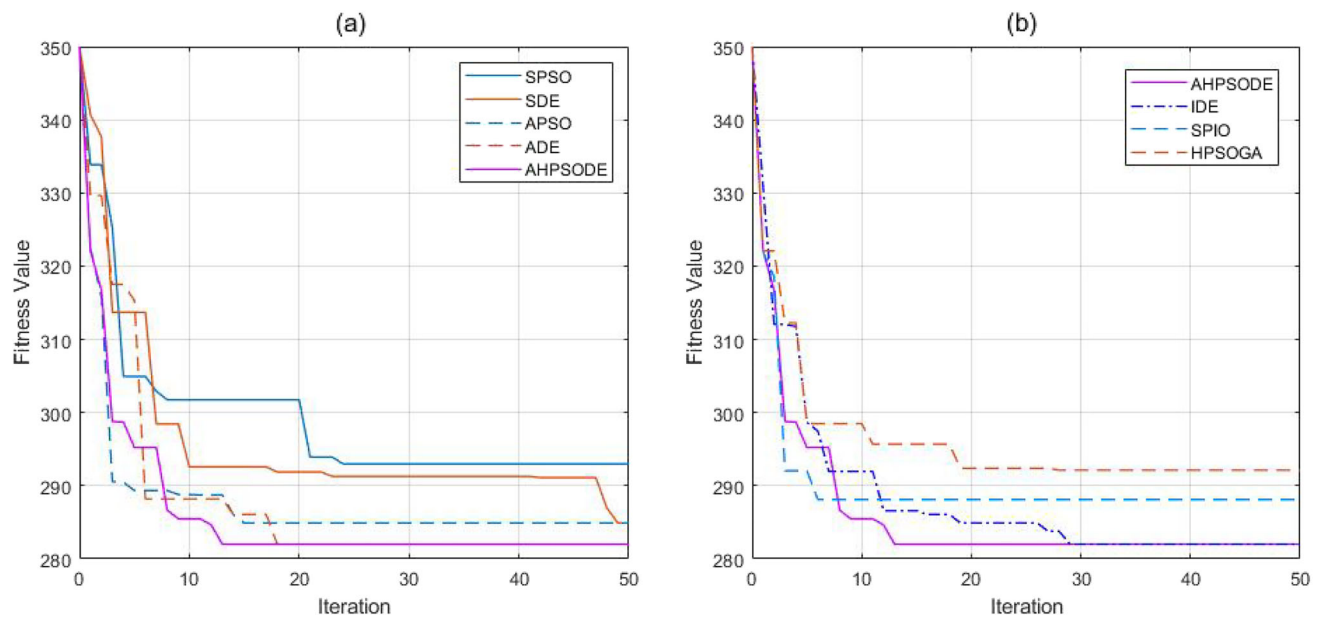


Fig. 11 Convergence curve of different algorithms in solving the target assignment problem of multi-UAV reconfiguration

Table 9 Parameter settings for different algorithms

Algorithm	Parameter setting
AHPSODE	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2, F = 0.2$
HPSOGA[14]	$\omega = 0.4 \sim 0.9, c_1 = c_2 = 2, P_c = 0.9, P_m = 0.05$
IDE[17]	$CR = 0.5 \sim 0.9, F = 0.2$
PIO[15]	$N1 = 0.9 * N, N2 = 0.1 * N, R = 0.003$

Table 10 States of UAVs' group

Serial number	Initial states	Target states
1	(-20, 0, 0)	(21.2, 21.2, 30)
2	(-15, 0, 0)	(0, 30, 30)
3	(-10, 0, 0)	(21.2, 21.2, 30)
4	(-5, 0, 0)	(-30, 0, 30)
5	(0, 0, 0)	(0, 0, 30)
6	(5, 0, 0)	(-21.2, -21.2, 30)
7	(10, 0, 0)	(0, -30, 30)
8	(15, 0, 0)	(-30, 0, 30)
9	(20, 0, 0)	(21.2, -21.2, 30)

input sequence can provide a smooth trajectory for the UAV group (Table 11).

Performance analysis of AHPSODE-based RHC controller

Based on the states of Table 10, we can get the optimal task assignment solution $assign = [5\ 9\ 4\ 3\ 1\ 2\ 7\ 8]$ by the discrete AHPSODE algorithm. Then, the relative states ΔX_{ref} can be determined. Figure 12 shows the initial position and target position of the UAV group. The blue star indicates the initial

Table 11 Constraints for UAV control inputs and states

Parameters	Value
Sum force $V_B(m/s)$	$[-15, 15]$
Roll angle ϕ_i ($^\circ$)	$[-45, 45]$
Pitch angle θ_i ($^\circ$)	$[-45, 45]$
Yaw angle ψ_i ($^\circ$)	$[-90, 90]$
Position x (m)	$[0, 100]$
Position y (m)	$[-100, 100]$
Position z (m)	$[-100, 100]$

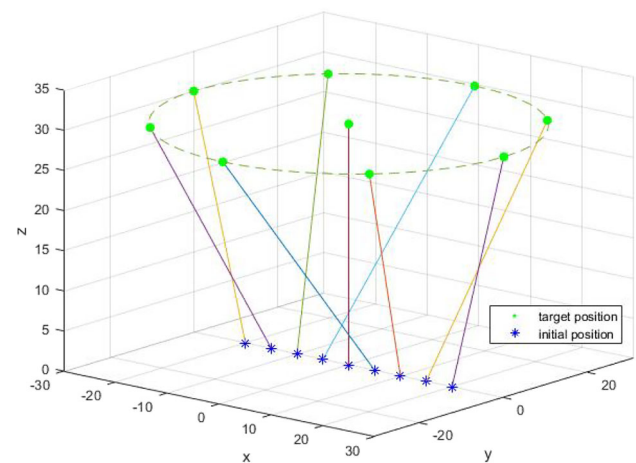


Fig. 12 Initial positions, target positions, and moving targets of 9 UAVs' group in 3D space

position of UAV formation. The green circle indicates the target position of the reconstructed formation, and the colored solid line indicates the matching relationship calculated

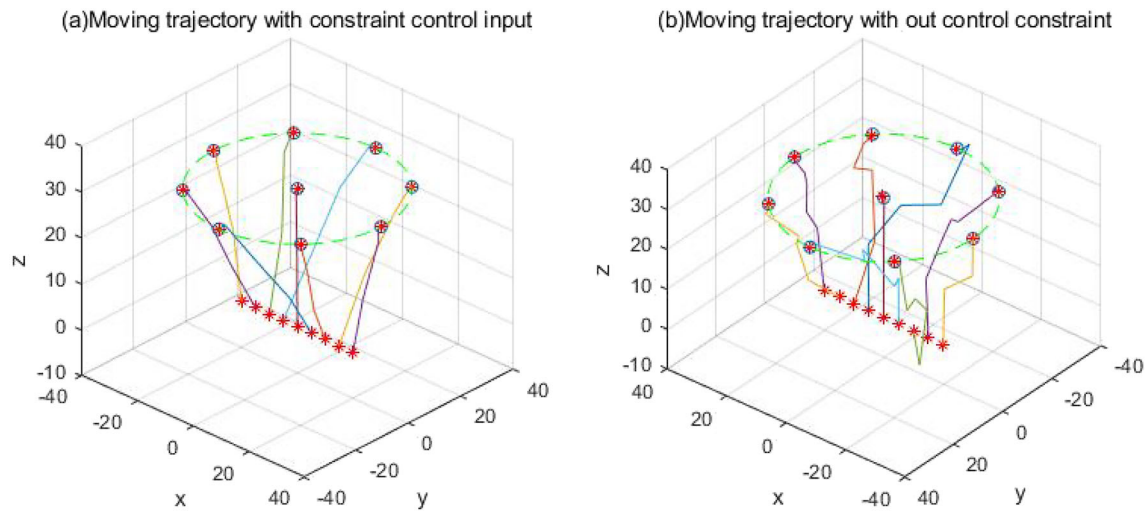


Fig. 13 Effect of constraints on generated trajectory

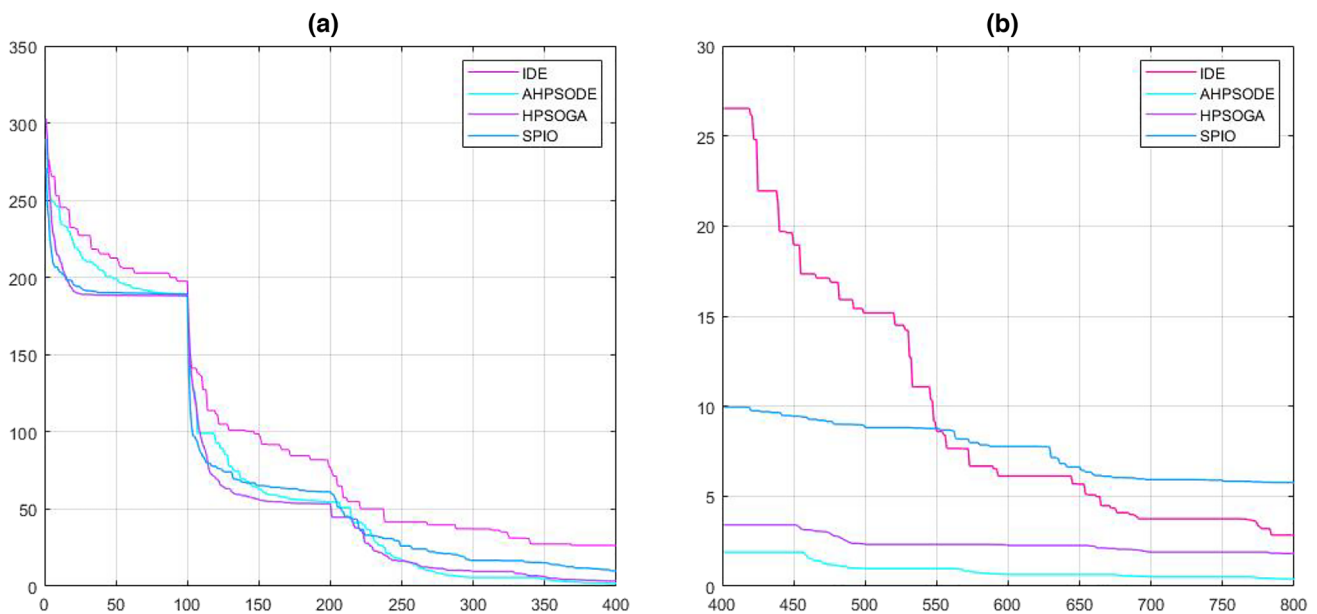


Fig. 14 **a** Evolution curve of all algorithms at the 1st–4th time horizon; **b** evolution curve of all algorithms at the 5th–8th time horizon

by the discrete AHPSODE. Figure 13 illustrates the effect of constraints on the generated trajectory. By the AHPSODE-based RHC controller, the original straight line formation can be successfully changed into the target circular formation. However, the trajectory generated without constraints is quite jagged.

The convergence curve of the IDE, SPIO, HPSOGA, and AHPSODE is shown in Fig. 14. In Fig. 14, every 100 iterations represent a control time domain of 1s. We also divided the 800 iterations into the first 400 iterations (Fig. 14a) and the last 400 iterations (Fig. 14b) to better observe the conver-

gence characteristics. According to the convergence curve, the controller will converge in time-domain 8, and the proposed AHPSODE has a faster convergence speed than IDE and SPIO. It also has a higher accuracy than HPSOGA. At the same time, according to the convergence curve of each time-domain, it can be found that the AHPSODE is easy to jump out local optimum and search a new solution with better fitness. This can help AHPSODE have a bigger probability finding a better solution when dealing with real problems.

To verify the performance of the RHC method, we also compared RHC method with the CPTD method mentioned in

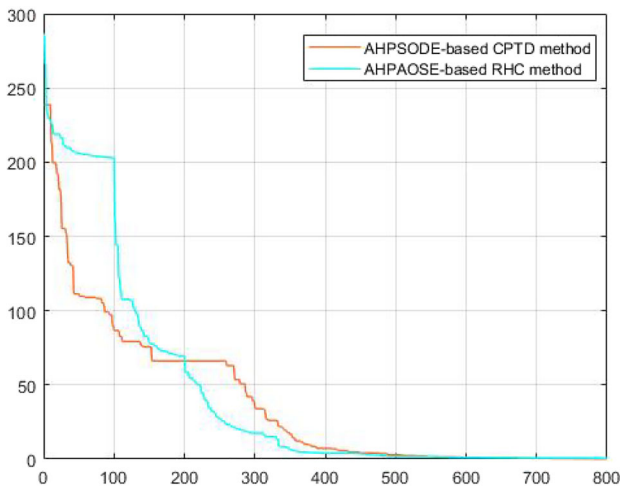


Fig. 15 The convergence curve of AHPSODE-based RHC controller and AHPSODE-based CPTD controller

[14]. We first construct an AHPSODE-based RHC controller and AHPSODE-based CPTD controller. For a fair comparison, the population size is set as 100, and all populations will update 800 times. Then, two controllers are used to solve the problem of multi-UAV formation reconfiguration mentioned in this section. Figure 15 shows the convergence curve of AHPSODE-based RHC controller and AHPSODE-based

CPTD controller. Although the search accuracy of these two methods is similar at the same iteration, the CPTD method takes 28s for 800 iterations, while the RHC method only takes 7 s. Therefore, the RHC method can save more computing resources than the CPTD method.

Figure 16 illustrates the optimized control inputs in eight time domains for each UAV. The results demonstrate that each UAV’s control inputs in all time-domains satisfy our constraint. Figure 17 depicts the distance between UAV5 and other UAVs in each time domain, and it can be seen that UAV5 satisfies both the safe distance and communication distance requirements in each time domain.

The average fitness values and average time cost of ten repeated experiment for different algorithms in each time-domain are shown in Table 12. From Table 12, the AHPSODE method requires the least amount of time and has the highest level of computing precision compared to all other methods.

To verify the performance of our proposed method when solving large scale UAV formation reconfiguration, we set the number of population size $n = 50$, the max iteration time $G = 50$, and the $GroupScale = [10, 20, 30, \dots, 100]$, $v_B = 15$. As this part of the experiment investigates the time consumption of the method described in this paper in solving the reconfiguration problem of multi-UAV formations of different group scale, the initial positions are equidistant

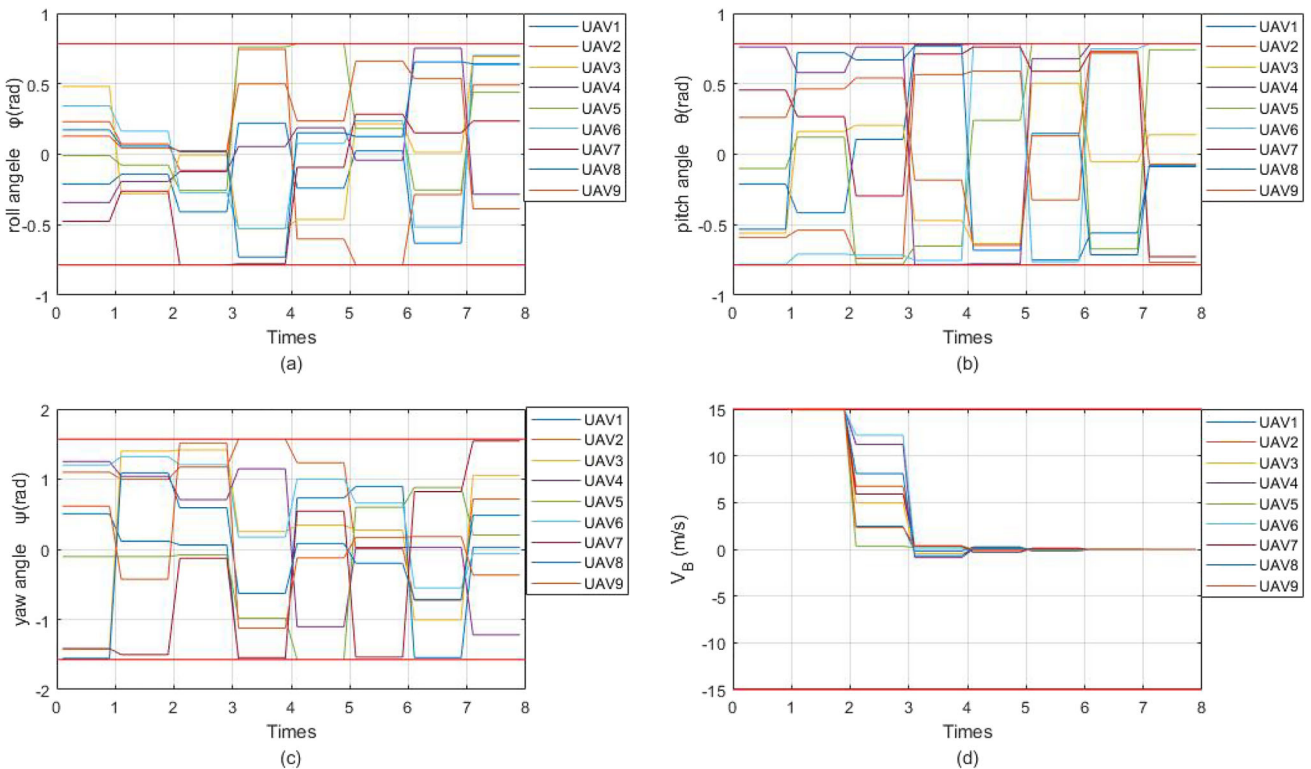


Fig. 16 Control inputs’ curve for each UAV at all time-domain

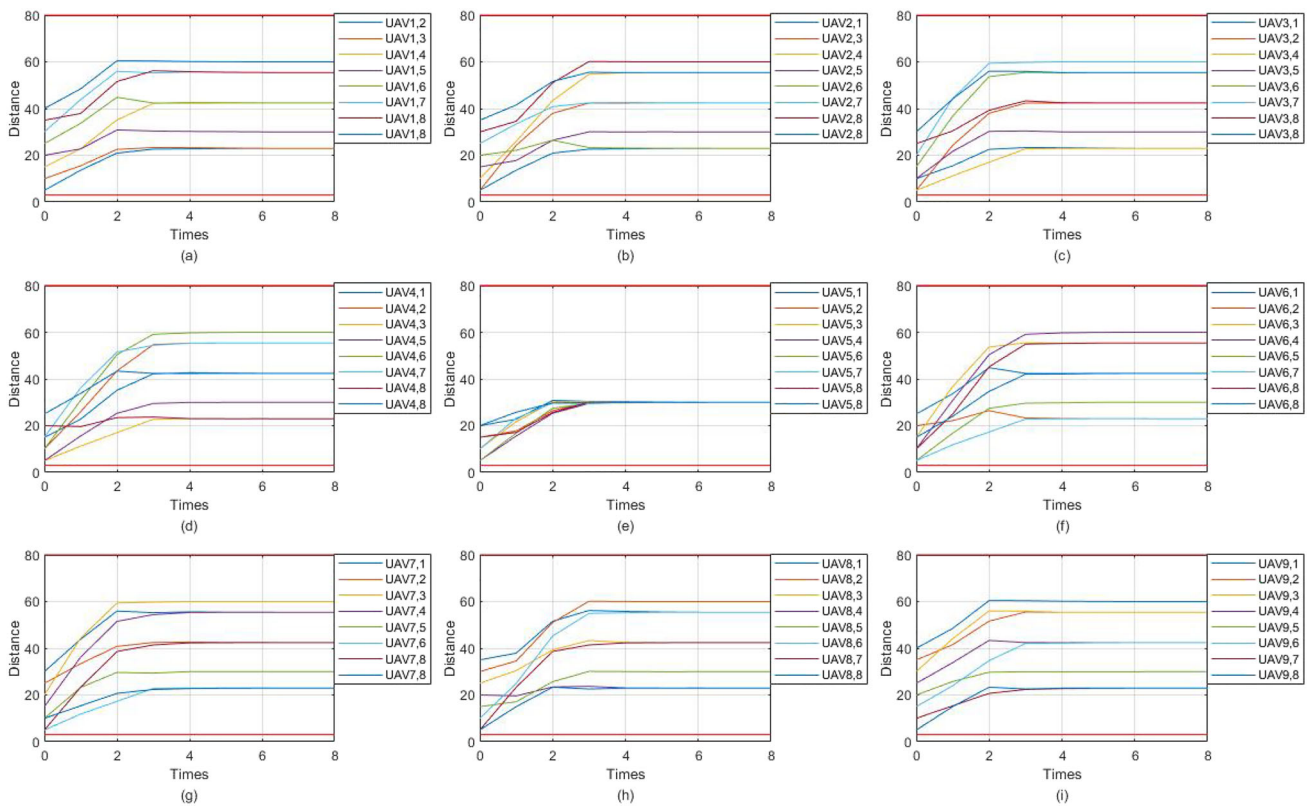


Fig. 17 Distance curve between each UAV for UAV 1–9 (a)–(i)

Table 12 Average fitness value for different algorithms

Domain	IDE	11SPPIO	HPSOGA	AHPSODE
1 s	201.52	194.73	187.1535	189.188
2 s	74.98	65.35	59.27	61.19
3 s	25.43	19.54	10.46	9.18
4 s	14.72	14.05	2.68	6.75
5 s	7.97	10.84	1.14	3.32
6 s	5.19	8.73	0.56	1.56
7 s	3.45	8.78	0.844	0.548
8 s	2.33	6.21	0.611	0.196
Time cost	8.49s	8.96s	9.63s	6.051s

distributed in $[-30, 30]$, and the target positions are on a circular area with $[0, 0, 50]$ as the center and a radius of 15 without considering collisions and communication distance. The experimental scene is shown in Fig. 18.

Table 13 shows the experimental results for different scales of UAV formation reconfiguration scenarios. It can be seen from Table 13 that the time consumption increases rapidly as the number of UAVs increases. Therefore, our proposed AHPSODE-based method cannot solve the real-

time formation reconfiguration problem for large-scale UAV groups.

According to the above simulations, we can conclude that our proposed AHPSODE algorithm can utilize the advantages of both DE and PSO in solving optimization problems. Meanwhile, the convergence curve of different algorithms also indicates that the AHPSODE has higher search veracity than SPSO, more rapid convergence speed. Finally, the proposed AHPSODE-Based RHC Controller can successfully transform the initial UAV formation to the desired target formation in 3-D space and each UAV satisfies all constraints in the reconfiguration process.

Conclusion

In this paper, an AHPSODE-based RHC controller is proposed to solve multiple-UAV formation reconfiguration problem in a 3-D environment. The proposed AHPSODE algorithm is first used to solve the discrete target assignment problem of formation reconfiguration. Then combined with the idea of rolling time-domain control, the global control problem is decomposed into several local optimization problems, and then, the AHPSODE algorithm is used to

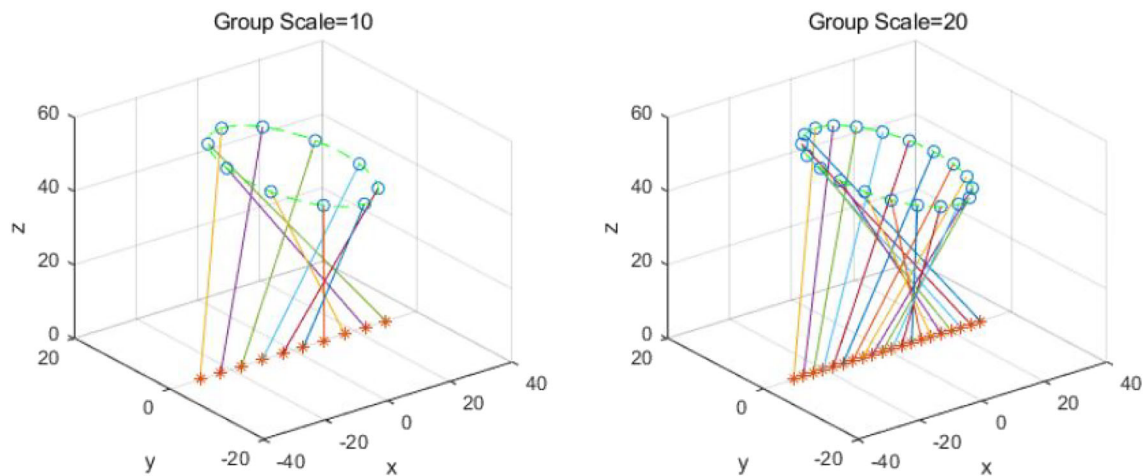


Fig. 18 Scenes of formation reconfiguration for multi-UAV formation of 10 scale and 20 scale

Table 13 Time cost for different scale of UAV group by AHPSODE

Group scale	10	20	30	40	50	60	70	80	90	100
Timecost	6.679	11.126	15.932	21.262	26.851	33.13	39.26	45.61	52.54	59.707
gbestvalue	1.02	12.47	68.22	147.25	330.73	473.97	670.17	817.23	1120.80	1243.09

find the optimal solution of the local optimization problem. Through the numerical simulation experiment, it can be proved that the AHPSODE-based RHC controller has higher search precision and speed compared. At the same time, due to the decomposition of the global control problem, the complexity and computational overhead of multi-UAV reconfiguration is reduced, and the real-time performance of the controller in the formation reconfiguration task is improved. In future work, the problem of formation reconfiguration with multi-objective optimization and large-scale problem will be consider. Furthermore, the optimization of the controller within a framework of optimal control will also be considered as a further research direction to enhance the integrity and practicality of the formation reconfiguration strategy.pg

Acknowledgements This research is funded by National Natural Science Foundation of China (No. 61902291), China Postdoctoral Science Foundation Funded Project (2019M653567), National Natural Science Foundation of Shaanxi Province (2019JM-425), and National Natural Science Foundation of China (No. 11961001).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix

See Tables 14 and 15 and Figs. 19, 20, 21, 22, 23, 24, 25, 26.

Table 14 Results on AHPSODE compared with other algorithms at 30-D

Fun		SPSO	SDE	APSO	ADE	AHPSODE	HPSOGA	IDE	SPIO
F1	Mean	4.10E+08	2.39E+10	1.88E+09	1.68E+10	1.59E+08	1.10E+10	6.76E+09	1.46E+09
	Std	8.08E+08	2.39E+09	1.06E+09	4.20E+09	5.00E+08	6.75E+09	7.16E+09	3.97E+08
F3	Mean	2.90E+04	2.01E+05	4.73E+04	1.95E+05	1.96E+03	2.10E+05	1.54E+05	6.64E+04
	Std	7.83E+03	1.86E+04	1.33E+04	1.47E+04	1.48E+03	1.41E+05	2.40E+04	5.67E+03
F4	Mean	6.25E+02	1.68E+03	7.14E+02	9.55E+02	5.58E+02	1.04E+03	5.86E+02	9.23E+02
	Std	1.63E+02	2.30E+02	1.18E+02	1.88E+02	8.57E+01	3.76E+02	1.22E+02	6.30E+01
F5	Mean	6.48E+02	8.30E+02	6.96E+02	8.20E+02	6.36E+02	7.88E+02	7.57E+02	7.60E+02
	Std	2.51E+01	9.87E+00	2.09E+01	1.40E+01	4.00E+01	2.81E+01	7.34E+01	2.41E+01
F6	Mean	6.42E+02	6.57E+02	6.52E+02	6.44E+02	6.32E+02	6.56E+02	6.28E+02	6.30E+02
	Std	1.44E+01	3.24E+00	1.05E+01	6.09E+00	6.28E+00	1.68E+01	1.99E+01	3.45E+00
F7	Mean	9.54E+02	1.88E+03	9.68E+02	1.59E+03	9.63E+02	1.16E+03	1.38E+03	1.09E+03
	Std	5.31E+01	1.33E+02	5.94E+01	9.27E+01	7.65E+01	1.74E+02	3.26E+02	2.87E+01
F8	Mean	9.01E+02	1.15E+03	9.68E+02	1.12E+03	9.34E+02	1.09E+03	1.07E+03	1.06E+03
	Std	2.00E+01	1.17E+01	3.11E+01	2.69E+01	3.45E+01	4.05E+01	9.18E+01	2.05E+01
F9	Mean	3.09E+03	1.41E+04	4.78E+03	1.14E+04	3.14E+03	9.70E+03	1.00E+04	4.77E+03
	Std	7.71E+02	1.42E+03	1.88E+03	1.31E+03	1.10E+03	2.77E+03	4.45E+03	7.55E+02
F10	Mean	5.55E+03	5.49E+03	5.92E+03	6.46E+03	5.00E+03	7.97E+03	6.08E+03	8.14E+03
	Std	8.71E+02	1.05E+03	5.13E+02	1.05E+03	5.86E+02	6.09E+02	8.77E+02	2.74E+02
F11	Mean	1.43E+03	1.77E+03	1.67E+03	1.73E+03	1.26E+03	1.08E+04	1.82E+03	2.68E+03
	Std	6.29E+01	5.68E+01	1.88E+02	6.80E+01	5.08E+01	1.04E+04	7.36E+02	3.53E+02
F12	Mean	3.08E+07	3.35E+08	6.35E+07	3.01E+08	6.38E+07	5.58E+08	8.45E+07	2.54E+08
	Std	6.15E+07	8.36E+07	6.35E+07	1.18E+08	9.47E+07	4.68E+08	1.07E+08	7.23E+07
F13	Mean	4.07E+04	3.63E+06	1.07E+05	2.62E+06	8.07E+06	3.41E+08	4.51E+06	5.05E+07
	Std	2.25E+04	2.25E+06	6.05E+04	1.21E+06	2.24E+07	6.94E+08	4.20E+06	2.70E+07
F14	Mean	5.20E+04	1.61E+03	5.37E+03	1.73E+03	3.65E+03	4.32E+05	3.15E+05	1.82E+05
	Std	1.22E+05	2.57E+01	5.84E+03	6.53E+01	4.50E+03	6.73E+05	4.00E+05	8.05E+04
F15	Mean	3.70E+04	6.86E+03	2.43E+04	1.69E+04	1.12E+04	2.09E+06	1.66E+05	1.06E+07
	Std	5.51E+04	1.24E+03	1.66E+04	7.23E+03	1.70E+04	1.41E+06	5.36E+04	5.47E+06
F16	Mean	3.04E+03	3.05E+03	2.90E+03	2.95E+03	2.57E+03	3.64E+03	2.59E+03	3.47E+03
	Std	2.48E+02	3.30E+02	4.01E+02	3.40E+02	4.33E+02	3.50E+02	2.46E+02	2.51E+02
F17	Mean	2.40E+03	2.18E+03	2.43E+03	2.13E+03	2.26E+03	2.61E+03	2.13E+03	2.39E+03
	Std	2.69E+02	2.03E+02	3.04E+02	1.55E+02	1.77E+02	2.78E+02	1.60E+02	8.42E+01
F18	Mean	3.90E+04	3.80E+05	2.65E+05	4.96E+05	5.32E+04	2.22E+06	9.15E+05	1.91E+06
	Std	2.72E+04	4.54E+05	2.44E+05	3.26E+05	9.02E+04	3.28E+06	6.89E+05	8.26E+05
F19	Mean	3.30E+04	3.05E+05	1.72E+05	4.04E+05	2.32E+03	1.52E+07	1.14E+06	1.99E+07
	Std	3.73E+04	2.06E+05	2.21E+05	2.85E+05	5.24E+02	2.17E+07	2.51E+06	1.15E+07
F20	Mean	2.73E+03	2.58E+03	2.77E+03	2.55E+03	2.37E+03	2.83E+03	2.40E+03	2.65E+03
	Std	2.54E+02	1.61E+02	2.49E+02	1.61E+02	1.21E+02	1.88E+02	1.55E+02	6.19E+01
F21	Mean	2.47E+03	2.63E+03	2.46E+03	2.59E+03	2.43E+03	2.57E+03	2.54E+03	2.53E+03
	Std	2.89E+01	1.09E+01	3.58E+01	1.78E+01	2.08E+01	1.84E+01	6.61E+01	2.32E+01
F22	Mean	4.01E+03	8.75E+03	6.04E+03	8.74E+03	4.98E+03	5.31E+03	6.95E+03	2.56E+03
	Std	2.37E+03	1.09E+03	2.27E+03	8.29E+02	2.10E+03	2.82E+03	1.74E+03	4.77E+01

Table 14 continued

Fun		SPSO	SDE	APSO	ADE	AHPSODE	HPSOGA	IDE	SPIO
F23	Mean	2.95E+03	2.93E+03	2.99E+03	2.92E+03	2.89E+03	2.93E+03	2.85E+03	2.92E+03
	Std	9.06E+01	1.69E+01	1.01E+02	9.50E+00	4.29E+01	4.62E+01	6.10E+01	2.25E+01
F24	Mean	3.12E+03	3.08E+03	3.19E+03	3.07E+03	3.04E+03	3.09E+03	3.03E+03	3.09E+03
	Std	8.34E+01	1.17E+01	1.15E+02	1.49E+01	4.12E+01	3.02E+01	3.69E+01	2.43E+01
F25	Mean	2.98E+03	4.72E+03	3.06E+03	4.35E+03	2.90E+03	3.13E+03	3.11E+03	3.21E+03
	Std	3.50E+01	2.78E+02	6.09E+01	3.55E+02	1.02E+01	2.67E+02	2.34E+02	3.56E+01
F26	Mean	6.23E+03	6.77E+03	6.76E+03	6.55E+03	5.67E+03	6.34E+03	6.14E+03	5.14E+03
	Std	1.31E+03	2.06E+02	8.39E+02	1.75E+02	1.48E+03	1.12E+03	6.29E+02	2.73E+02
F27	Mean	3.34E+03	3.23E+03	3.38E+03	3.22E+03	3.26E+03	3.30E+03	3.22E+03	3.40E+03
	Std	6.55E+01	1.36E+01	8.82E+01	6.87E+00	4.15E+01	4.75E+01	5.10E+00	2.19E+01
F28	Mean	3.32E+03	4.99E+03	3.48E+03	4.47E+03	3.28E+03	3.68E+03	3.65E+03	3.72E+03
	Std	3.67E+01	8.32E+02	7.92E+01	1.02E+03	4.69E+01	3.82E+02	7.29E+02	5.62E+01
F29	Mean	4.49E+03	3.90E+03	4.57E+03	3.77E+03	3.88E+03	4.60E+03	3.59E+03	4.62E+03
	Std	3.73E+02	2.65E+02	2.58E+02	2.67E+02	2.28E+02	3.34E+02	1.25E+02	7.11E+01
F30	Mean	9.14E+05	5.47E+05	3.98E+06	5.91E+05	6.38E+04	1.78E+07	7.35E+04	2.07E+07
	Std	1.34E+06	1.86E+05	3.27E+06	2.13E+05	1.08E+05	2.44E+07	2.09E+04	7.37E+06

Table 15 Results on AHPSODE compared with other algorithms at 50-D

Fun		SPSO	SDE	APSO	ADE	AHPSODE	HPSOGA	IDE	SPIO
F1	Mean	8.77E+09	9.11E+10	1.83E+10	6.83E+10	1.86E+09	2.58E+10	4.88E+10	1.40E+10
	Std	3.60E+09	6.16E+09	7.15E+09	8.02E+09	1.54E+09	9.28E+09	1.58E+10	1.95E+09
F3	Mean	1.05E+05	4.37E+05	1.51E+05	4.51E+05	3.09E+04	3.89E+05	3.72E+05	1.69E+05
	Std	2.49E+04	3.32E+04	3.11E+04	2.93E+04	9.85E+03	1.89E+05	4.61E+04	1.74E+04
F4	Mean	1.28E+03	1.15E+04	2.52E+03	9.43E+03	8.11E+02	2.65E+03	3.65E+03	2.60E+03
	Std	3.27E+02	8.90E+02	9.34E+02	1.61E+03	1.85E+02	1.17E+03	1.88E+03	3.40E+02
F5	Mean	8.48E+02	1.24E+03	8.95E+02	1.22E+03	8.13E+02	1.07E+03	1.02E+03	1.07E+03
	Std	7.10E+01	2.13E+01	3.65E+01	3.17E+01	6.52E+01	7.01E+01	1.51E+02	2.68E+01
F6	Mean	6.62E+02	6.86E+02	6.66E+02	6.79E+02	6.42E+02	6.73E+02	6.78E+02	6.56E+02
	Std	7.80E+00	4.64E+00	9.65E+00	4.39E+00	5.19E+00	1.48E+01	2.59E+01	3.47E+00
F7	Mean	1.36E+03	4.01E+03	1.45E+03	3.47E+03	1.32E+03	1.83E+03	2.53E+03	1.73E+03
	Std	1.15E+02	2.50E+02	1.53E+02	5.30E+02	1.09E+02	2.71E+02	6.65E+02	5.30E+01
F8	Mean	1.15E+03	1.53E+03	1.20E+03	1.51E+03	1.11E+03	1.41E+03	1.38E+03	1.37E+03
	Std	5.85E+01	2.00E+01	5.37E+01	2.74E+01	5.99E+01	8.64E+01	1.15E+02	2.44E+01
F9	Mean	1.29E+04	4.50E+04	1.59E+04	3.96E+04	1.01E+04	4.06E+04	3.64E+04	2.39E+04
	Std	2.69E+03	3.16E+03	2.68E+03	2.83E+03	1.95E+03	1.09E+04	8.99E+03	2.80E+03
F10	Mean	9.63E+03	1.27E+04	1.07E+04	1.34E+04	8.26E+03	1.44E+04	1.16E+04	1.44E+04
	Std	9.52E+02	1.22E+03	1.07E+03	1.05E+03	7.39E+02	9.96E+02	1.92E+03	5.12E+02
F11	Mean	3.20E+03	1.49E+04	6.43E+03	1.18E+04	1.60E+03	1.54E+04	5.99E+03	1.06E+04
	Std	6.58E+02	1.40E+03	2.13E+03	1.64E+03	8.77E+01	1.13E+04	3.62E+03	1.29E+03
F12	Mean	1.41E+09	9.62E+09	2.03E+09	8.54E+09	2.36E+09	5.92E+09	2.15E+09	1.26E+09
	Std	1.62E+09	1.60E+09	1.37E+09	8.88E+08	2.00E+09	3.16E+09	2.05E+09	2.94E+08
F13	Mean	9.52E+07	3.76E+08	6.19E+06	2.75E+08	2.87E+08	3.60E+09	1.07E+08	2.34E+08
	Std	2.04E+08	2.44E+08	5.96E+06	2.21E+08	6.00E+08	4.04E+09	2.62E+08	7.13E+07

Table 15 continued

Fun		SPSO	SDE	APSO	ADE	AHPSODE	HPSOGA	IDE	SPIO
F14	Mean	5.24E+05	8.47E+05	3.27E+05	6.40E+05	5.37E+05	1.17E+06	1.40E+06	2.29E+06
	Std	1.24E+06	3.23E+05	4.51E+05	3.04E+05	1.34E+06	1.11E+06	8.86E+05	6.58E+05
F15	Mean	3.20E+04	9.46E+06	5.09E+04	2.66E+06	1.61E+04	2.40E+08	1.11E+06	1.21E+08
	Std	2.42E+04	2.18E+07	3.51E+04	1.60E+06	8.22E+03	3.76E+08	2.28E+06	7.61E+07
F16	Mean	4.10E+03	5.81E+03	4.44E+03	5.81E+03	3.58E+03	4.98E+03	4.63E+03	5.54E+03
	Std	4.13E+02	3.01E+02	4.71E+02	2.83E+02	3.92E+02	3.74E+02	8.27E+02	2.20E+02
F17	Mean	3.60E+03	4.61E+03	3.56E+03	4.59E+03	3.24E+03	4.48E+03	3.79E+03	4.34E+03
	Std	4.23E+02	1.78E+02	4.20E+02	2.27E+02	4.63E+02	3.77E+02	1.93E+02	1.96E+02
F18	Mean	7.98E+05	1.40E+07	3.38E+06	1.66E+07	6.22E+05	2.72E+07	5.71E+06	1.63E+07
	Std	1.46E+06	4.34E+06	1.62E+06	4.64E+06	1.48E+06	4.38E+07	2.58E+06	4.67E+06
F19	Mean	4.35E+05	2.36E+06	2.61E+06	2.72E+06	6.77E+03	2.86E+08	7.61E+06	4.18E+07
	Std	7.07E+05	1.76E+06	3.65E+06	3.08E+06	5.97E+03	4.65E+08	2.13E+07	1.84E+07
F20	Mean	3.10E+03	3.85E+03	3.62E+03	3.71E+03	3.06E+03	4.14E+03	3.33E+03	3.89E+03
	Std	3.74E+02	4.28E+02	3.29E+02	4.35E+02	2.37E+02	2.83E+02	3.12E+02	1.49E+02
F21	Mean	2.70E+03	3.02E+03	2.73E+03	2.96E+03	2.60E+03	2.92E+03	2.95E+03	2.79E+03
	Std	6.02E+01	1.76E+01	5.30E+01	3.84E+01	5.68E+01	4.51E+01	6.29E+01	2.63E+01
F22	Mean	1.19E+04	1.59E+04	1.17E+04	1.59E+04	9.32E+03	1.51E+04	1.36E+04	1.25E+04
	Std	1.30E+03	3.05E+02	8.24E+02	4.39E+02	1.76E+03	1.10E+03	1.78E+03	1.74E+03
F23	Mean	3.49E+03	3.42E+03	3.59E+03	3.38E+03	3.27E+03	3.38E+03	3.24E+03	3.32E+03
	Std	1.55E+02	3.41E+01	1.82E+02	2.37E+01	4.70E+01	9.62E+01	1.16E+02	4.71E+01
F24	Mean	3.57E+03	3.46E+03	3.81E+03	3.43E+03	3.40E+03	3.49E+03	3.34E+03	3.52E+03
	Std	1.40E+02	1.85E+01	1.81E+02	2.89E+01	6.65E+01	8.19E+01	6.66E+01	3.72E+01
F25	Mean	3.84E+03	1.48E+04	4.53E+03	1.07E+04	3.24E+03	4.10E+03	6.59E+03	5.74E+03
	Std	3.52E+02	2.35E+03	7.40E+02	1.37E+03	1.41E+02	9.38E+02	3.51E+03	2.65E+02
F26	Mean	9.90E+03	1.09E+04	1.17E+04	1.03E+04	8.90E+03	1.05E+04	9.68E+03	1.14E+04
	Std	1.98E+03	1.75E+02	1.53E+03	1.72E+02	1.07E+03	1.23E+03	1.65E+03	7.38E+02
F27	Mean	4.02E+03	3.51E+03	4.18E+03	3.44E+03	3.73E+03	3.91E+03	3.39E+03	4.30E+03
	Std	1.65E+02	7.20E+01	1.77E+02	5.14E+01	1.46E+02	2.48E+02	4.91E+01	9.18E+01
F28	Mean	4.48E+03	8.12E+03	5.14E+03	7.95E+03	3.68E+03	6.31E+03	7.80E+03	5.96E+03
	Std	7.05E+02	9.21E+01	5.29E+02	1.22E+02	3.59E+02	1.16E+03	2.47E+02	1.91E+02
F29	Mean	6.76E+03	6.16E+03	7.61E+03	5.84E+03	5.08E+03	6.37E+03	4.89E+03	6.89E+03
	Std	7.12E+02	2.64E+02	1.65E+03	3.08E+02	3.23E+02	6.93E+02	5.70E+02	2.04E+02
F30	Mean	5.06E+07	8.54E+07	1.76E+08	6.25E+07	2.87E+06	2.81E+08	8.87E+06	2.97E+08
	Std	2.94E+07	3.60E+07	5.13E+07	2.68E+07	2.17E+06	1.93E+08	7.28E+06	3.81E+07

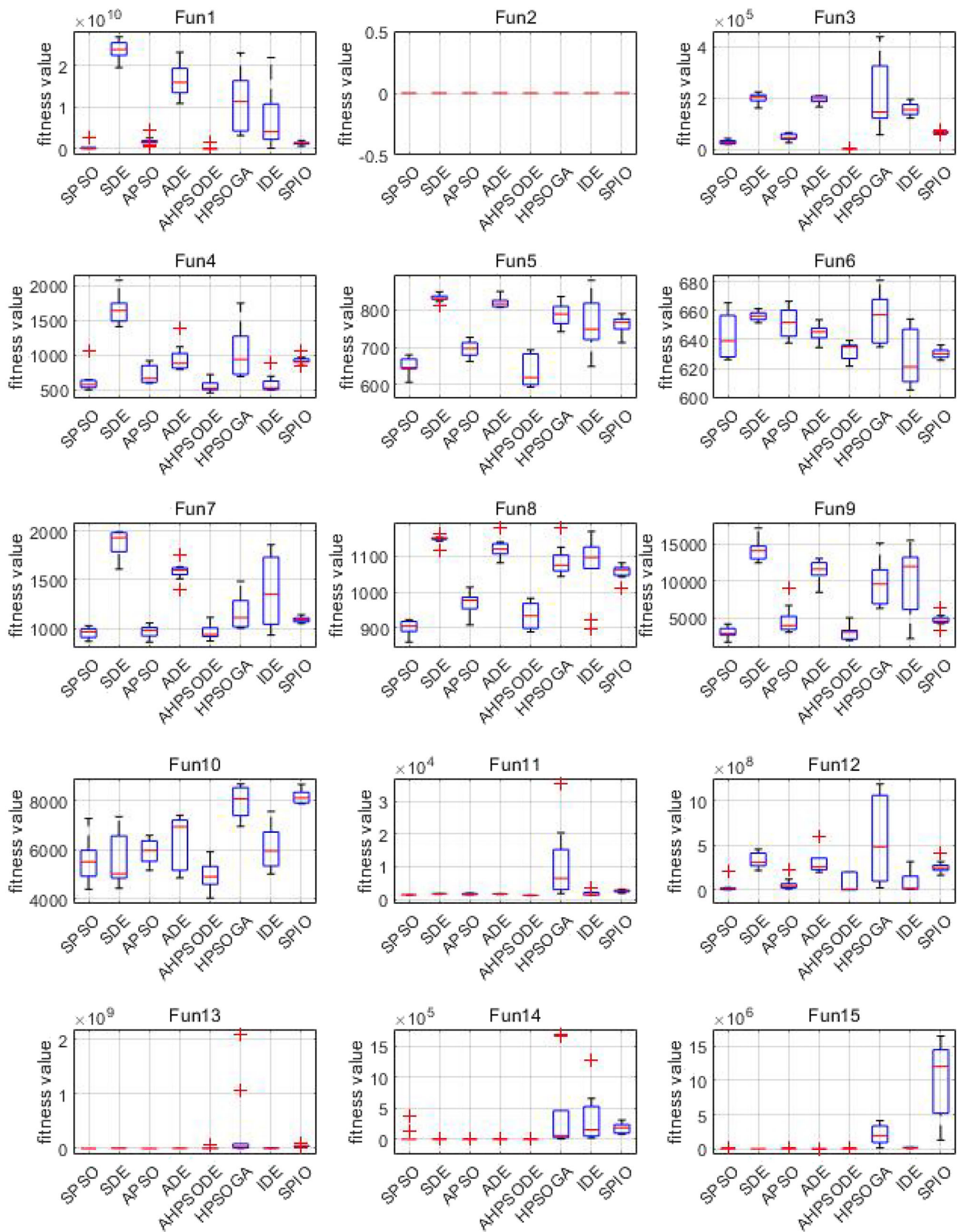


Fig. 19 Box plot of different algorithms on CEC 2017 at 30-D (Fun 1–15)

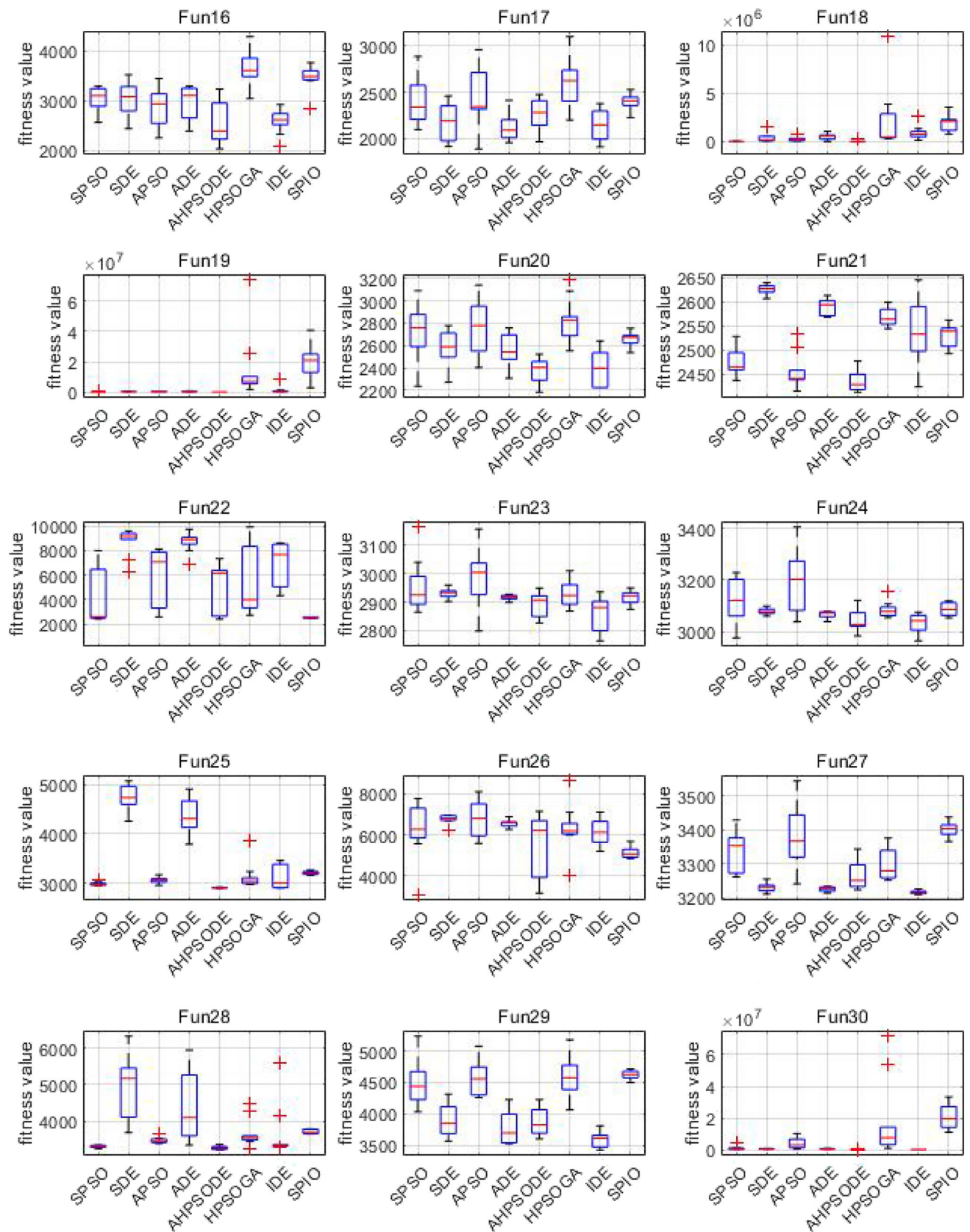


Fig. 20 Box plot of different algorithms on CEC 2017 at 30-D (Fun 16–30)

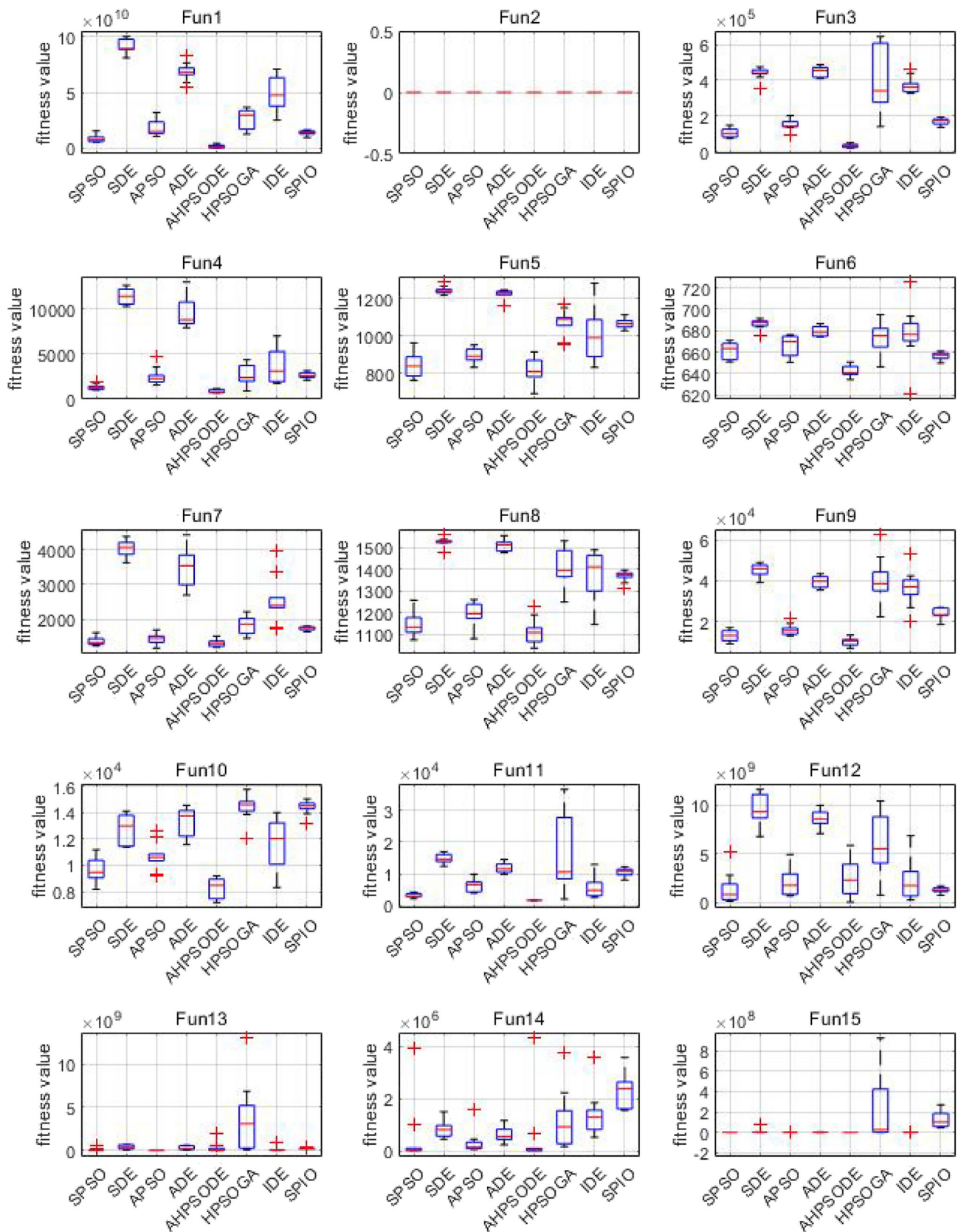


Fig. 21 Box plot of different algorithms on CEC 2017 at 50-D (Fun 1–15)

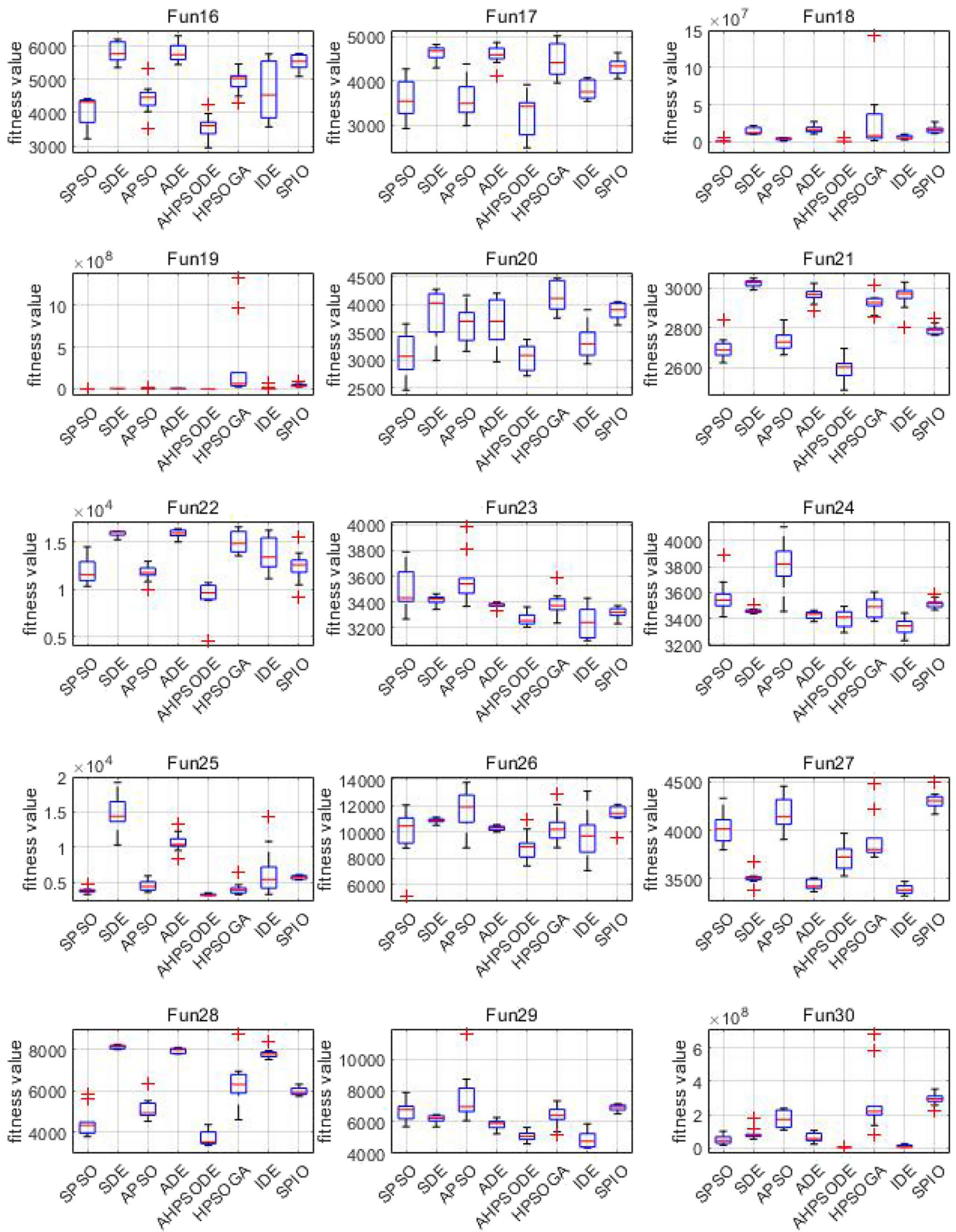


Fig. 22 Box plot of different algorithms on CEC 2017 at 50-D (Fun 16–30)

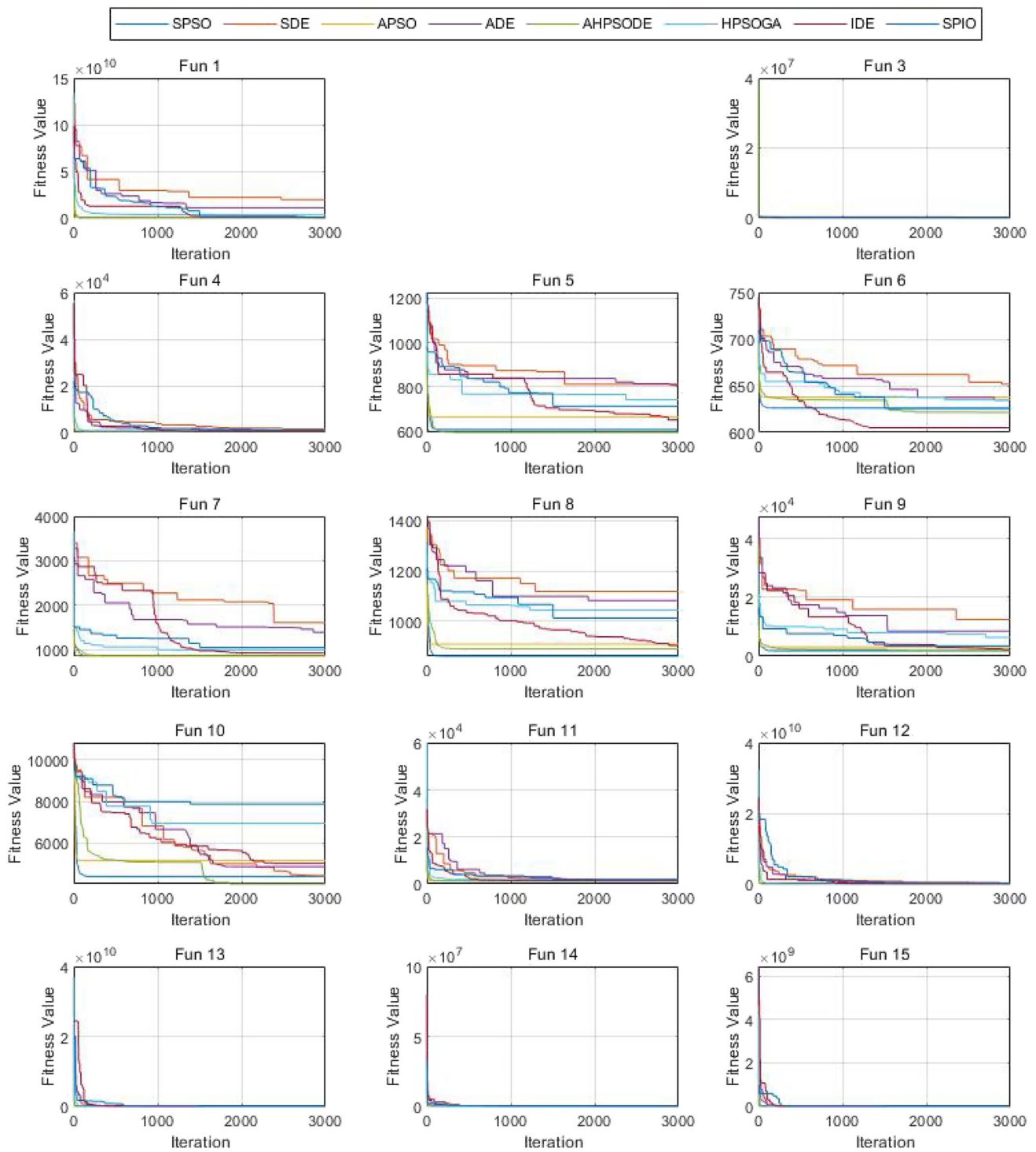


Fig. 23 Convergence curve of different algorithms on CEC 2017 at 30-D (Fun 1–15)

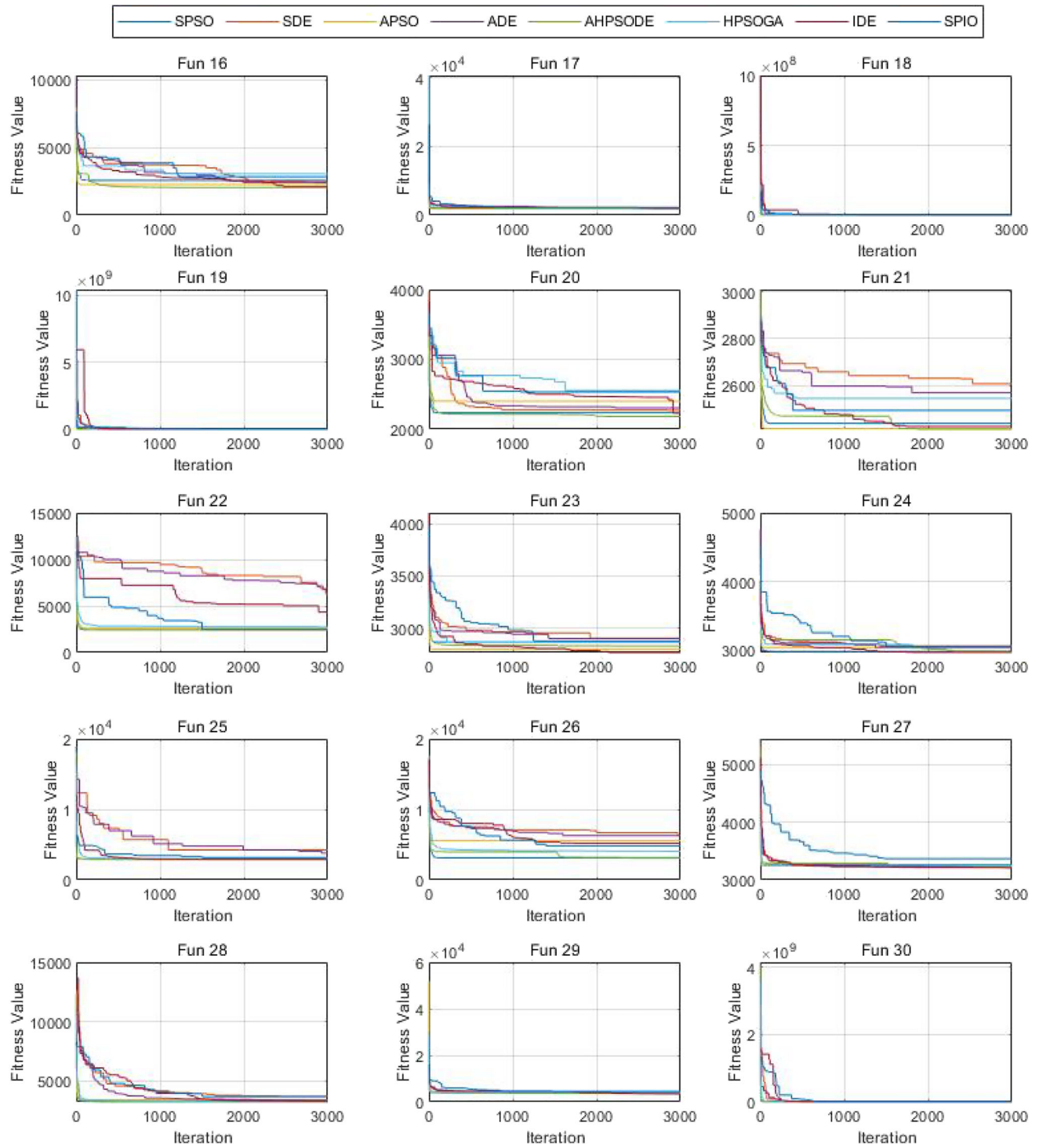


Fig. 24 Convergence curve of different algorithms on CEC 2017 at 30-D (Fun 1–15)

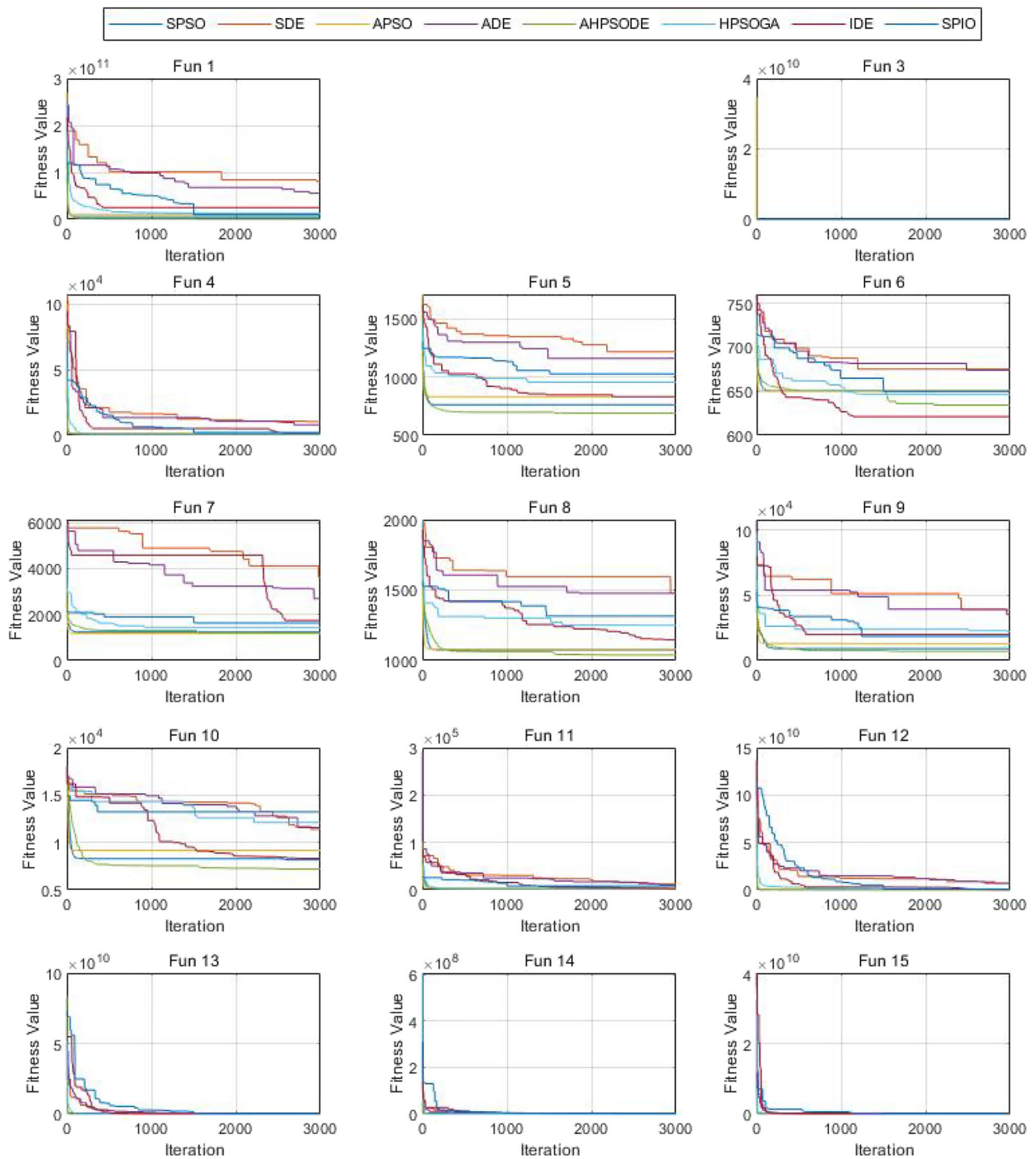


Fig. 25 Convergence curve of different algorithms on CEC 2017 at 50-D (Fun 15–30)

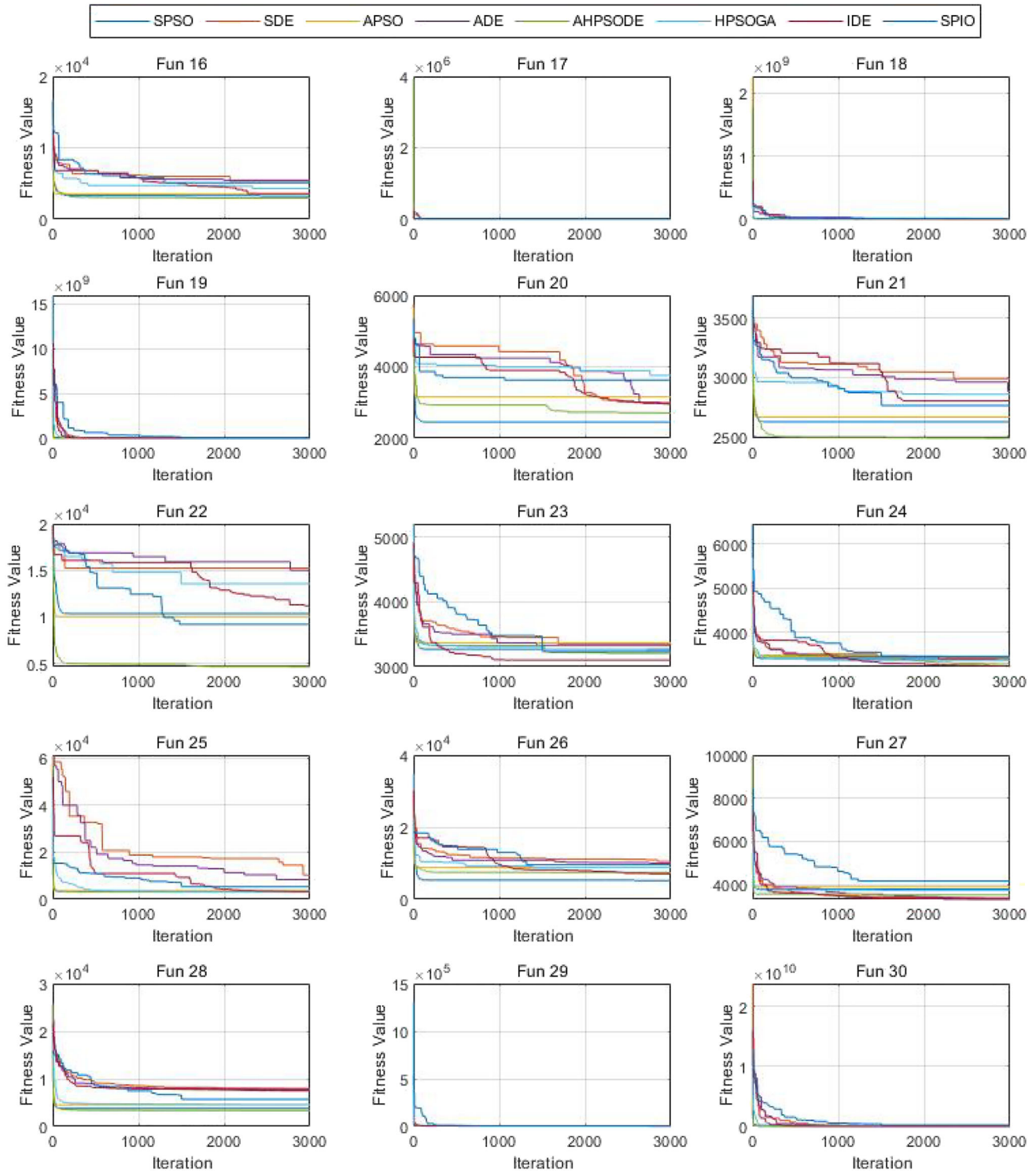


Fig. 26 Convergence curve of different algorithms on CEC 2017 at 50-D (Fun 15–30)

References

- Seo J, Kim Y, Kim S, Tsourdos A (2017) Collision avoidance strategies for unmanned aerial vehicles in formation flight. *IEEE Trans Aerosp Electron Syst* 53(6):2718–2734
- Lee D, Kim S, Suk J (2018) Formation flight of unmanned aerial vehicles using track guidance. *Aerosp Sci Technol* 76:412–420
- Chen Y, Yu J, Su X, Luo G (2015) Path planning for multi-uav formation. *J Intell Robot Syst* 77(1):229–246
- Garcia M, Viguria A, Ollero A (2013) Dynamic graph-search algorithm for global path planning in presence of hazardous weather. *J Intell Robot Syst* 69(1):285–295
- Cajo R, Mac TT, Copot C, Plaza D, De Keyser R, Ionescu C (2019) Multiple uavs formation for emergency equipment and medicines delivery based on optimal fractional order controllers. In: 2019 IEEE international conference on systems, man and cybernetics (SMC). IEEE, pp 318–323
- Chevet T, Maniu CS, Vlad C, Zhang Y (2018) Voronoi-based uavs formation deployment and reconfiguration using mpc techniques. In: 2018 international conference on unmanned aircraft systems (ICUAS). IEEE, pp 9–14
- Chen Y, Luo G, Mei Y, Yu J, Su X-l (2016) Uav path planning using artificial potential field method updated by optimal control theory. *Int J Syst Sci* 47(6):1407–1420
- Fan J, Li Y, Liao Y, Ma T, Ge Y, Wang Z (2020) A formation reconfiguration method for multiple unmanned surface vehicles executing target interception missions. *Appl Ocean Res* 104:102359
- Quesada WO, Rodriguez JI, Murillo JC, Cardona GA, Yanguas-Rojas D, Jaimes LG, Calderón JM (2018) Leader–follower formation for uav robot swarm based on fuzzy logic theory. In: International conference on artificial intelligence and soft computing. Springer, pp 740–751
- Liu H, Meng Q, Peng F, Lewis FL (2020) Heterogeneous formation control of multiple uavs with limited-input leader via reinforcement learning. *Neurocomputing* 412:63–71
- Furukawa T, Durrant-Whyte HF, Bourgault F, Dissanayake G (2003) Time-optimal coordinated control of the relative formation of multiple vehicles. In: Proceedings 2003 IEEE international symposium on computational intelligence in robotics and automation. Computational intelligence in robotics and automation for the new millennium (Cat. No. 03EX694), vol 1. IEEE, pp 259–264
- Shao S, Peng Y, He C, Du Y (2020) Efficient path planning for uav formation via comprehensively improved particle swarm optimization. *ISA Trans* 97:415–430
- Wei X, Zongji C, Rui Z (2008) Optimization of multiple flight vehicle formation reconfiguration using hybrid genetic algorithm. *Acta Aeronaut Astronaut Sin* 1:209–214
- Duan H, Luo Q, Shi Y, Ma G (2013) hybrid particle swarm optimization and genetic algorithm for multi-uav formation reconfiguration. *IEEE Comput Intell Mag* 8(3):16–27
- Zhang X, Duan H, Yang C (2014) Pigeon-inspired optimization approach to multiple uavs formation reconfiguration controller design. In: Proceedings of 2014 IEEE Chinese guidance, navigation and control conference. IEEE, pp 2707–2712
- Xiang-Yin Z, Hai-Bin D (2012) Differential evolution-based receding horizon control design for multi-uavs formation reconfiguration. *Trans Inst Meas Control* 34(2–3):165–183
- Zhang B, Sun X, Liu S, Deng X (2019) Adaptive differential evolution-based receding horizon control design for multi-uav formation reconfiguration. *Int J Control Autom Syst* 17(12):3009–3020
- Wu G, Mallipeddi R, Suganthan PN (2017) Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report
- Brandao AS, Barbosa JP, Mendoza V, Sarcinelli-Filho M, Carelli R (2014) A multi-layer control scheme for a centralized uav formation. In: 2014 international conference on unmanned aircraft systems (ICUAS). IEEE, pp 1181–1187
- Brandão AS, Sarcinelli-Filho M (2016) On the guidance of multiple uav using a centralized formation control scheme and delaunay triangulation. *J Intell Robot Syst* 84(1–4):397–413
- Xiang X, Liu C, Su H, Zhang Q (2017) On decentralized adaptive full-order sliding mode control of multiple uavs. *ISA Trans* 71:196–205
- Thien RT, Kim Y (2018) Decentralized formation flight via pid and integral sliding mode control. *Aerosp Sci Technol* 81:322–332
- Lee G, Chwa D (2018) Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. *Intell Serv Robot* 11(1):127–138
- Mousavi S, Afghah F, Ashdown JD, Turck K (2018) Leader–follower based coalition formation in large-scale uav networks, a quantum evolutionary approach. In: IEEE INFOCOM 2018-IEEE conference on computer communications workshops (INFOCOM WKSHPs). IEEE, pp 882–887
- Liang Y, Qi D, Yanjie Z (2020) Adaptive leader–follower formation control for swarms of unmanned aerial vehicles with motion constraints and unknown disturbances. *Chin J Aeronaut* 33(11):2972–2988
- Kownacki C (2016) Multi-uav flight using virtual structure combined with behavioral approach. *Acta Mech Autom* 10(2):92–99
- Peterson CK, Barton J (2015) Virtual structure formations of cooperating uavs using wind-compensation command generation and generalized velocity obstacles. In: 2015 IEEE aerospace conference. IEEE, pp 1–7
- Fan J, Liao Y, Li Y, Jiang Q, Wang L, Jiang W (2019) Formation control of multiple unmanned surface vehicles using the adaptive null-space-based behavioral method. *IEEE Access* 7:87647–87657
- El Ferik S, Nasir MT, Baroudi U (2016) A behavioral adaptive fuzzy controller of multi robots in a cluster space. *Appl Soft Comput* 44:117–127
- Mahfouz M, Hafez AT, Ashry M, Elnashar G (2018) Formation configuration for cooperative multiple uav via backstepping pid controller. In: 2018 AIAA SPACE and astronautics forum and exposition, pp 5282
- Vlahakis EE, Milonidis E, Halikias GD (2018) Cooperative distributed lqr control for longitudinal flight of a formation of non-identical low-speed experimental uav's. In: 2018 UKACC 12th international conference on control (CONTROL). IEEE, pp 295–300
- Zhang J, Yan J, Zhang P (2020) Multi-uav formation control based on a novel back-stepping approach. *IEEE Trans Veh Technol* 69(3):2437–2448
- Zhihao C, Longhong W, Jiang Z, Kun W, Yingxun W (2020) Virtual target guidance-based distributed model predictive control for formation control of multiple uavs. *Chin J Aeronaut* 33(3):1037–1056
- Eberhart R, Kennedy J (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol 4. Citeseer, pp 1942–1948

35. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
36. Stone HS (1977) Multiprocessor scheduling with the aid of network flow algorithms. *IEEE Trans Softw Eng* 1:85–93

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