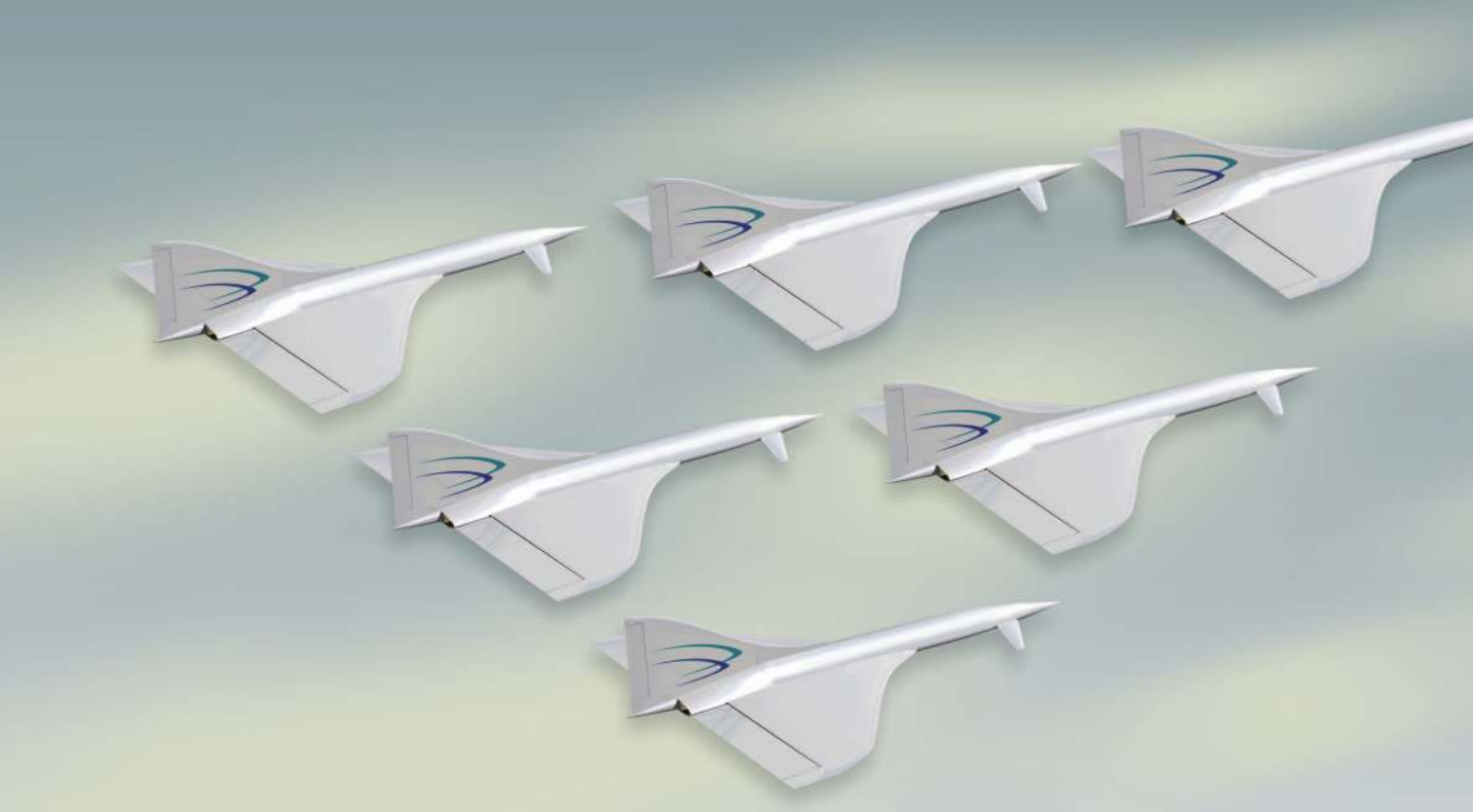

Hybrid Particle Swarm Optimization and Genetic Algorithm for Multi-UAV Formation Reconfiguration

Haibin Duan, Qinan Luo, and Guanjun Ma
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (BUAA), Beijing, PR CHINA

Yuhui Shi
Xi'an Jiaotong-Liverpool University, Suzhou, PR CHINA

I. Introduction

Unmanned Aerial Vehicle (UAV) is becoming an integral part of future military forces and will be used for complex tasks including surveillance, reconnaissance, precision strike and aerial refueling missions in the presence of disturbances, failures, and complicated battlefield subjected to uncertainties and variations. Therefore, more attention is now paid to various control problems associated with multi-UAV moving in formation [1]–[4]. The benefits of formation flight (as shown in Fig. 1) include fuel savings at certain close formation positions, tanker formation operations where flights of UAVs are ferried by a single tanker, and mission success in terms of redundancy and

Digital Object Identifier 10.1109/MCI.2013.2264577
Date of publication: 12 July 2013

Abstract—Given the initial state of an Unmanned Aerial Vehicle (UAV) system and the relative state of the system, the continuous inputs of each flight unit are piecewise linear by a Control Parameterization and Time Discretization (CPTD) method. The approximation piecewise linearization control inputs are used to substitute for the continuous inputs. In this way, the multi-UAV formation reconfiguration problem can be formulated as an optimal control problem with dynamical and algebraic constraints. With strict constraints and mutual interference, the multi-UAV formation reconfiguration in 3-D space is a complicated problem. The recent boom of bio-inspired algorithms has attracted many researchers to the field of applying such intelligent approaches to complicated optimization problems in multi-UAVs. In this paper, a Hybrid Particle Swarm Optimization and Genetic Algorithm (HPSOGA) is proposed to solve the multi-UAV formation reconfiguration problem, which is modeled as a parameter optimization problem. This new approach combines the advantages of Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), which can find the time-optimal solutions simultaneously. The proposed HPSOGA will also be compared with basic PSO algorithm and the series of experimental results will show that our HPSOGA outperforms PSO in solving multi-UAV formation reconfiguration problem under complicated environments.

battle damage assessment. Therefore, formation control is becoming more and more important. There are three main approaches to formation control, namely Leader-Wingman, Virtual Leader, and Behavioral Structures, which have been studied thoroughly [5]–[10]. In the Leader-Wingman structure, one of the UAVs in the formation is designated as the leader, with the rest of the UAVs (Wingmen) treated as followers. The basic idea is that the followers track the position and orientation of the leader. Because of its simplicity, the Leader-Wingman structure is widely used in control and management of multi-vehicle formations [5]–[9]. The common weaknesses as reported in [5]–[9] are that the rear UAV usually exhibits a poorer response than its reference due to error propagations and the leader is a single point of failure for the formation.

In the Virtual Leader structure, the entire formation is treated as a single structure. Each UAV receives the same information, which is the trajectory of the Virtual Leader. It has been applied to formations of spacecraft in [11]. The strength of the Virtual Leader structure is that it is easy to prescribe the formation behavior. However, its disadvantage is that there is no explicit feedback to the formation.

A different strategy is represented by a Behavioral approach. The basic idea of this approach is to prescribe several behaviors for each aircraft and to make the control action of each aircraft a weighted average of the control for each behavior. The behaviors may be obstacle avoidance, collision avoidance, target seeking, and formation keeping. It was first introduced by Anderson and Robbins [10] before it was further exploited by Giulietti [12] with the introduction of an imaginary point in the formation called the Formation Geometry Center (FGC). Asada, et al [13] successfully used a behavioral-based control to display soccer-playing robots. Since the sensors used for data acquisition introduce noise into the system, the handling of sensor data needs to be improved.

The application of behavioral approach to aircraft formation flight is described in [10], where the control strategies are inspired by instinctive behaviors of birds, fishes, insects, and herds. Balch and Arkin [14] presented a behavioral-based approach to robot formation keeping, where control strategies are formed by averaging several competing behaviors. In [15], obstacle avoidance while keeping formation is achieved using basic behaviors, such as move-to-goal, avoid-obstacle, maintain-relative-distance, maintain-relative-angle, and stop move to the target destination. However, in [10], [14], [15], the common shortcomings are that the information needed are too plentiful and the characteristics of the formation (like stability) cannot generally be guaranteed.

Many researchers have investigated the formation control problems, but very few literatures focus on the reconfiguration issue of UAV formation [16]. Reconfigurable control may be needed in cases of failure in one or more communication channels [16], sensor/actuator failures [17], flight path constraints or even the total loss of the aircraft without impairing its mission.

This work mainly focuses on formation reconfiguration problem, which was first addressed by Wang and Hadaegh [18]. The concept of formation reconfiguration involves determining aircraft separation distance, position and orientation, identifying the process that optimally transforms an initial formation configuration into a final configuration and identifying cooperative autonomous control (CAC) of individual aircrafts to achieve a desired final configuration. Determining the transformation process incorporates three main technologies: constraint satisfaction, quality determination, and planning. Formation reconfiguration can be classified into two basic types. For type 1, each aircraft is required to occupy a specified position in the desired reconfigured formation, while for type 2, a specified

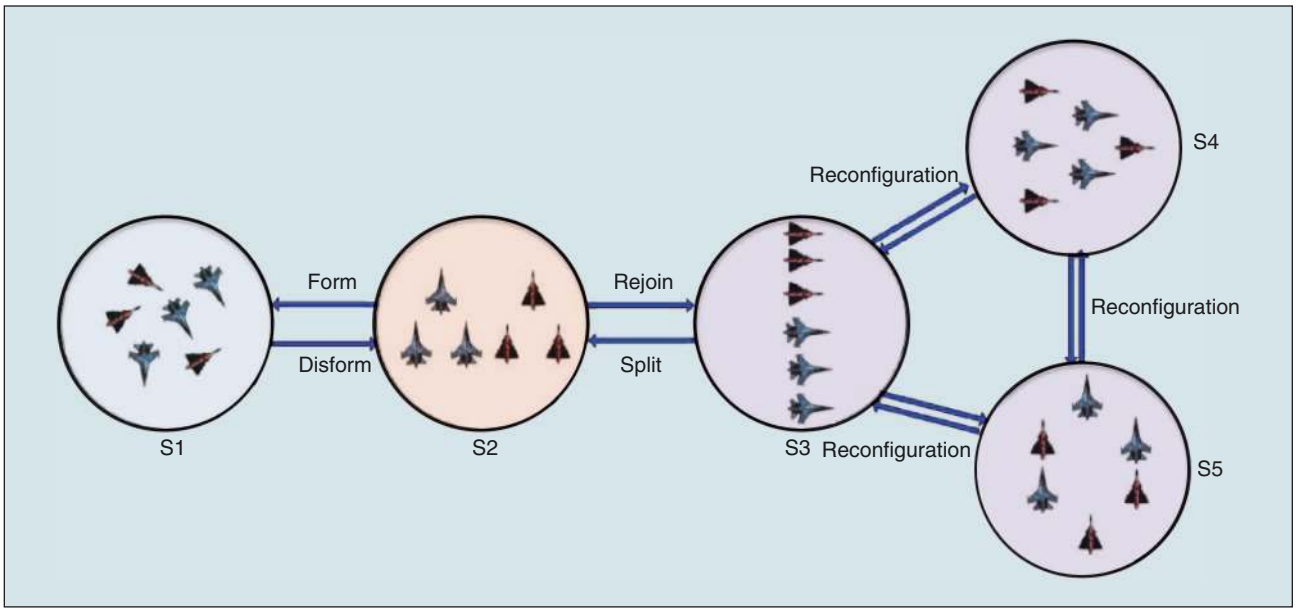

FIGURE 1 Multi-UAV formation.

position in the desired reconfigured formation may be occupied by any aircraft of a particular type.

Several approaches have been applied to the formation reconfiguration problem. By formulating the formation reconfiguration process into a sequence of basic maneuvers, Wang and Hadaegh [19] obtained simple solutions for formation reconfiguration. Fierro and Das [20] considered the singularities and collision constraints to dynamically reconfigure the team of autonomous robots. Wang and Zheng [21] presented a Hierarchical Evolutionary Trajectory Planner (HETP), which has two levels that perform global planning and multiple optimal or near optimal trajectory designs respectively for spacecraft formation reconfiguration in 3-D space. Sauter and Palmer [22] developed a semi-analytic approach and applied it to satellites for a rapid onboard, fuel-minimized, and collision-free path generation, which can significantly increase the responsiveness of the formation to reconfiguration events. Ma et al. [23] designed a general formulation for time optimal trajectory planning of satellites formation reconfiguration using pseudospetral method, and the optimization Nonlinear Programming (NLP) problem was solved by searching the state and control vectors to minimize an objective function. 3-D potential field method was used in [24] to solve formation flight and formation reconfiguration of multi-UAVs, including obstacle and collision avoidance.

By following the inspirational previous works, lots of evolutionary computation methods have been proposed, developed and studied for scientific research and engineering applications [25]. Gueret et al. [26] applied soft computing methods to two of the Semantic Web reasoning tasks; an evolutionary approach to querying, and a swarm algorithm for entailment. Muhleisen and Dentler [27] proposed a novel concept for reasoning within a fully distributed and self-organized storage system based on the collective behavior of swarm individuals. Meng et al. [28] presented a hierarchical mechanochemical model for self-reconfiguration of modular robots in changing environments. Le et al. [29] also presented a theoretic model of symbiotic evolution for the design of water clusters potential model.

The formation reconfiguration problem can be formulated as an optimal control problem with dynamical and algebraic constraints. For this reason, artificial intelligence algorithms and/or other optimization methods can be utilized to find the optimal solution [30], [31].

Furakawa et al. [32] presented a method, where the control strategies are based on the Control Parameterization and Time Discretization (CPTD) method, to solve the time-optimal control of the relative formation of multi-robotic vehicles. However, in their method, time is a fixed value and it is not always possible to obtain the optimal solution at the given value. Xiong, et al. [33] proposed an improved Genetic Algorithm (GA), which incorporates CPTD, for multi-fighter formation reconfiguration optimization. In [33], time is an optimization parameter but this method can only be used in 2-D environments, which limits its application range. The improved GA can be used to transform the problem of time-optimal control for formation reconfiguration into discrete optimization problem with a free terminal state constraint by CPTD, and the experimental results show that it's suitable for obtaining the global optima of time-optimal control of formation reconfiguration in two dimensions.

In this paper, formation reconfiguration in 3-D space is modeled as a parameter optimization problem and a Hybrid Particle Swarm Optimization and Genetic Algorithm (HPSOGA) is proposed. There are two critical issues in the

basic GA, one is its premature convergence, and the other is its weak local searching ability. Furthermore, GA also suffers from a slow convergence speed. Particle Swarm Optimization (PSO) is an efficient optimization algorithm for solving complicated continuous problems. PSO is similar to the continuous GA in that it begins with a random population matrix. Unlike the GA, PSO has no evolution operators such as crossover and mutation. One of the most obvious advantages of PSO over GA is its algorithmic simplicity as it uses a few parameters and is easy to implement. PSO can often locate nearly optimal solutions with a fast convergence speed, but usually fails to adjust its velocity step size for fine tuning in the search space, which often leads to premature convergence. Combining the advantages of PSO and GA, our hybrid approach can find time-optimal solutions simultaneously. Series of comparative results also show that our proposed HPSOGA outperforms PSO.

The rest of this paper is organized as follows. Section II introduces the basic principles of GA and PSO. Section III gives a description of time-optimal control problems for multi-UAV under the free terminal state constraint. Section IV introduces the application of HPSOGA to formation reconfiguration. Section V describes the HPSOGA implementation to solve the optimization problem. The experimental results are given in Section VI, followed by concluding remarks in Section VII.

II. The Standard GA and Standard PSO

A. Standard Genetic Algorithm

GA was first introduced by Holland in the early 1970s [34]. Generally, GA comprises three different phases in the global searching process:

- **Phase 1**: creating an initial population.
- **Phase 2**: evaluating a fitness function.
- **Phase 3**: producing a new population.

A genetic search starts with a randomly generated initial population, within which each individual is evaluated by means of a fitness function. Individuals in this and subsequent generations are duplicated or eliminated according to their fitness values. Individuals are further manipulated by applying GA operators. There are usually three GA operators in a typical genetic algorithm [38]. The first is the production operator (elitism) which makes one or more copies of any individual, with a high probability that possesses a high fitness value, conversely the individual with a low fitness value is eliminated with a high probability from the solution pool. The second operator is the crossover operator. This operator selects two individuals from the population in the current generation and a crossover point (taking the one-point crossover for example) and carries out a swapping operation on the elements to the right hand side of the crossover point of both individuals. The third operator is the mutation operator. This operator acts as a background operator and is used to explore some of the invested points in the search space. Since frequent application of this operator would lead to a completely random search, a very low probability is usually assigned to its operation.

B. Standard PSO

The PSO was formulated in terms of social and cognitive behavior by Kennedy and Eberhart in 1995 [36, 37], and has found wide applications in engineering. The PSO algorithm simulates social behavior among bird individuals (particles) flying through a multi-dimensional search space, each particle representing a point. The particles assess their positions by a fitness function and particles in a local neighborhood share memories of their best position, while using those memories to update their velocities and positions.

Particle updates in basic PSO are accomplished according to (1) and (2) [38]. Equation (1) calculates a new velocity for each particle based on its previous velocity (v_{id}), the particle's location at which the best fitness has been achieved (p_{id}) so far, and the best particle among its neighbors (p_{gd}) at which the best fitness has been achieved so far. Equation (2) updates each particle's position (x_{id}) in the solution hyperspace. The two random numbers r_1 and r_2 are independently generated and c_1 and c_2 are learning factors. The use of the inertia weight w provides improved performance.

$$v_{id} = wv_{id} + c_1 \cdot r_1 \cdot (p_{id} - x_{id}) + c_2 \cdot r_2 \cdot (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id}. \quad (2)$$

There are three parts on the right side of (1). The first part is the velocity part, which represents the influence of the previous velocity of the particle. The second part is the cognition part, which represents the private thinking of the particle. The third part is the social part, which represents the collaboration of the particles.

III. Time-Optimal Control for Multi-UAV

A. Equation of Motion

Assume the number of UAV in formation is N. The terminal time is $t = T$. T is not a given value but a parameter that should be optimized. The ith UAV's control inputs (including thrust, load factor, bank angle) are represented as $u_i = \{u_i(t) \mid \forall t \in [0, T]\} \in \Re^{r_i},\ \forall i \in \{1, \cdots, N\}$. The formation control input vector is $\boldsymbol{U} = (u_1, \cdots, u_N)$, while the continuous control input vector of the formation can be described as $\boldsymbol{U} = (u_1, \cdots, u_N) = \{\boldsymbol{U}(t) \mid \forall t \in [0, T]\}$. The ith UAV's state is $\chi_i = [v_i, \gamma_i, \chi_i, x_i, y_i, z_i]^T \in \Re^6,\ \forall i \in \{1, \cdots, N\}$, where (x_i, y_i, z_i) denotes its coordinates and v_i, γ_i, χ_i denote its airspeed, flight path angle and heading angle respectively. Therefore, the formation system state can be defined as $\boldsymbol{X} = (x_1^T, \cdots, x_N^T)^T \in \Re^{6*N}$. Consider a non-linear system in the standard form [32]:

$$\dot{\boldsymbol{X}}(t) = f(t, \boldsymbol{X}(t), U(t)). \quad (3)$$

Given a set of continuous control inputs $\boldsymbol{U}$ and the initial state $\boldsymbol{X}(0) = \boldsymbol{X}_0$, the state of the system at any time $t \in (0, T]$ can be determined uniquely in the form:

$$\boldsymbol{X}(t) = \boldsymbol{X}(0) + \int_0^{t^-} f(\tau, \boldsymbol{X}(\tau), \boldsymbol{U}(\tau))\, d\tau. \tag{4}$$

This means that given the initial state $\boldsymbol{X}(0)$, the state $\boldsymbol{X}(t)$ can be specified only by the control inputs $\boldsymbol{U}$ in the form $\boldsymbol{X}(t|U)$.

B. Objective Function and Constraints

It is well-known that the canonical form of the objective function can be expressed as [33]:

$$J(\boldsymbol{U}) = \Phi_0(\boldsymbol{X}(T|\boldsymbol{U})) + \int_0^T L_0(t, \boldsymbol{X}(t|\boldsymbol{U}), \boldsymbol{U}(t))\, dt. \tag{5}$$

The problem may also be subject to a variety of other constraints, generally in the form:

$$g_i(\boldsymbol{U}) = \Phi_i(\boldsymbol{X}(\tau_i|\boldsymbol{U})) + \int_0^{\tau_i} L_i(t, \boldsymbol{X}(t|\boldsymbol{U}), \boldsymbol{U}(t))\, dt \le 0 \quad \forall i \in \{1, \cdots, M\}. \tag{6}$$

For a single system, the optimal control problem can be formulated as finding the continuous control inputs $\boldsymbol{U}$ and terminal time T that minimize the objective function $J(\boldsymbol{U})$:

$$\min_{u_1, T} \cdots \min_{u_N, T} J(\boldsymbol{U}) \tag{7}$$

$$\min J(\boldsymbol{U}, \boldsymbol{T}). \tag{8}$$

The function $\boldsymbol{U}$ and time T are normally constrained by the following equation.

$$\boldsymbol{U}_{\min} \le \boldsymbol{U}(t) \le \boldsymbol{U}_{\max}, \forall t \in [0, T), T > 0. \tag{9}$$

Defining the mth UAV as the formation center, the free terminal constraint is given by:

$$g_1(\mathbf{U}, \Delta t) = \sum_{i=1}^{N} \{[(x_i(T) - x_m(T)) - x_i^m]^2 + [(y_i(T) - y_m(T)) - y_i^m]^2 + [(z_i(T) - z_m(T)) - z_i^m]^2\} = 0, \tag{10}$$

where $m \in \{1, \cdots, N\}$, $[x_i^m, y_i^m, z_i^m]^T$ represents the desired relative coordinates of the ith UAV with respect to the mth UAV.

The distance between any two UAVs i and j is defined to be:

$$d^{i,j}(x_i(t), x_j(t)) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2 + (z_i(t) - z_j(t))^2}. \tag{11}$$

In order to avoid collision, $d^{i,j}(x_i(t), x_j(t))$ must be greater than the safety collision distance D_{safe}.

$$d^{i,j}(x_i(t), x_j(t)) \ge D_{\text{safe}}, \quad \forall t \in [0, T], \forall_{i \ne j}\ i,j \in \{1, \cdots, N\}. \tag{12}$$

In order for real-time communication between the UAVs to update one another on the combat situation of the formation, $d^{i,j}(x_i(t), x_j(t))$ must be smaller than the communication distance.

$$d^{i,j}(r(t), m(t)) \ge D_{\text{comm}} \quad \forall t \in [0, T] \quad \forall_{i \ne j}\ i,j \in \{1, \cdots, N\}. \tag{13}$$

IV. HPSOGA Based Formation Reconfiguration Time-Optimal Controller

PSO and GA are global optimization algorithms and are suitable for solving optimization problems with linear or non-linear objective functions; Therefore, they are suitable for solving non-linear formation reconfiguration problem. However, the control inputs of each flight unit are continuous and the HPSOGA cannot solve the continuous control input problem. In order to solve this problem, the control inputs of each flight unit are piecewise linearized, and the approximation piecewise linearization control inputs are used to substitute the continuous inputs, then HPSOGA is used to find the global optimal solution. Based on the above ideas, this paper adopts the CPTD method, obtaining the approximate objective function and constraints condition, simplifying the problem in description and handling, and then using HPSOGA to find the approximate solution $\hat{\mathbf{U}}(t; n_p, \boldsymbol{\Omega})$ until satisfying the constraints of Eq. (12), (13), (18), (19) and (20).

A. Formation Reconfiguration Time-Optimal Control Discrete Based on CPTD Method

The continuous control inputs u_i are approximated by a piecewise function with a set of static parameters (in practice these static parameters are constants). The terminal time T is first partitioned into n_p time intervals. Partitioning is conducted to introduce a piecewise function with n_p constants that substitute the continuous control inputs.

The terminal time T is formulated as a function of time interval Δt_p, which is used for numerical integration.

The static control parameter is set and the time intervals are found by minimizing the objective function with a standard non-linear parametric optimization method.

The proposed method takes three steps to derive this approximate solution of the problem. The following subsections describe these steps.

1) The division of the terminal time T: The terminal time T is partitioned into $n_p \in \{1, 2, \cdots\}$ intervals, each $\Delta t_p \in \Re^+$, so

$$T = n_p \Delta t_p. \tag{14}$$

At each time interval Δt_p, according to the corresponding control inputs, equation (3) does numerical integration.

2) The piecewise linearization of control inputs: For the n_p intervals, define $r_i \times n_p$ constants for the ith UAV as $\boldsymbol{\Omega}_i = \{\sigma_j^i \in \Re^{r_i} | \forall j \in \{1, \cdots, n_p\}\}, \forall i \in \{1, \cdots, N\}$. Then,

each of the continuous control inputs for the ith UAV can be approximated by a piecewise function with constant as follows [3]:

$$\hat{u}_i(t; n_p, \mathbf{\Omega}_i) = \sum_{j=1}^{n_p} \sigma_j^i \chi_j(t) \cong u_i(t), \tag{15}$$

where $\chi_j(t)$ is given by

$$\chi_j(t) = \begin{cases} 1 & (j-1)\Delta t_p \le t \le j\Delta t_p \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Define the set of all piecewise constants for all UAVs as $\mathbf{\Omega} = \{\mathbf{\Omega}_1, \cdots, \mathbf{\Omega}_N\}$. The set of approximated control inputs for all the UAVs can be written as $\hat{\mathbf{U}}(t; n_p, \mathbf{\Omega}) = \{\hat{u}_1(t; n_p, \mathbf{\Omega}_1), \cdots, \hat{u}_N(t; n_p, \mathbf{\Omega}_N)\}$. Finding $\hat{\mathbf{U}}(t; n_p, \mathbf{\Omega})$ therefore results in finding the parameter set $\mathbf{\Omega}$. The most important thing for this approximation in practical implementations is an appropriate choice for n_p. Increasing n_p results in an exponentially increase in computation time, while reducing n_p results in loss of accuracy.

3) Approximation of control inputs: The approximation $\hat{\mathbf{U}}(t; n_p, \mathbf{\Omega}) = \{\hat{u}_1(t; n_p, \mathbf{\Omega}_1), \cdots, \hat{u}_N(t; n_p, \mathbf{\Omega}_N)\}$ can be derived from $\mathbf{\Omega}$ and Δt_p. In fact, finding $\hat{\mathbf{U}}(t; n_p, \mathbf{\Omega})$ and T is equivalent to finding $\mathbf{\Omega}$ and Δt_p introduces an approximate objective function and constraint function J. As a result, the dynamic optimization problem $\dot{\mathbf{X}}(t) \cong f(t, \mathbf{X}(t), \hat{\mathbf{U}}(t; n_p, \mathbf{\Omega}))$ can be transformed into the static optimization problem:

$$J \cong \min_{\mathbf{\Omega}, \Delta t_p} (n_p \Delta t_p). \tag{17}$$

Subject to bounds

$$\begin{aligned} (u_{\min})_i \le \sigma_j^i \le (u_{\max})_i \qquad & \forall i \in \{1, \cdots, N\}, \\ & \forall j \in \{1, \cdots, n_p\},\ 0 < \Delta t_p. \end{aligned} \tag{18}$$

And the free terminal constraint:

$$\begin{aligned} \hat{g}_1(\mathbf{\Omega}, \Delta t) = \sum_{i=1}^{N} \{ & [(x_i(T) - x_m(T)) - x_i^m]^2 \\ & + [(y_i(T) - y_m(T)) - y_i^m]^2 \\ & + [(z_i(T) - z_m(T)) - z_i^m]^2 \} = 0. \end{aligned} \tag{19}$$

The state of the system can be appropriately written as follows:

$$\dot{X}(t) = f(t, X(t), \hat{U}(t; n_p, W)). \tag{20}$$

Numerically, the formation reconfiguration problem is formulated as a parameter optimization problem with a nonlinear objective function and constraints. It can now be solved with the proposed HPSOGA. However, the solution found by HPSOGA will be near-optimal due to the CPTD method.

B. Time-Optimal Control of Formation Reconfiguration Based on HPSOGA

The construction of a particle's position is as follows: $\mathbf{\Omega} = \{\mathbf{\Omega}_1, \cdots, \mathbf{\Omega}_N\}$ combines with Δt_p as the particle's position. Thus, the position of each particle can be expressed as $P = [\mathbf{\Omega}_1, \mathbf{\Omega}_2, \cdots, \mathbf{\Omega}_N, \Delta t_p]$. Control parameter $\mathbf{\Omega}_i$ is a constant array, that is $\mathbf{\Omega}_i = \begin{bmatrix} \sigma_{11}^i & \sigma_{21}^i & \cdots & \sigma_{n_p 1}^i \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1r_i}^i & \sigma_{2r_i}^i & \cdots & \sigma_{n_p r_i}^i \end{bmatrix}, \forall i \in \{1, 2, \cdots N\}, \forall j \in \{1, 2, \cdots n_p\},\ \forall k \in \{1, 2, \cdots r_i\}$. σ_{jk}^i is the kth component of $\hat{u}_i(t; n_p, \mathbf{\Omega}_i)$ at the jth time interval. Because each column of $\mathbf{\Omega}_i$ represents the control parameter of the ith UAV at a particular time interval, we expand $\mathbf{\Omega}_i$ by column and combine it with Δt_p, eventually straightening it into a floating point code series of length $N \times n_p \times r_i + 1$. Finally, the particle's position can be expressed as:

$$\begin{aligned} X = [& ((\sigma_{11}^1, \sigma_{12}^1, \cdots, \sigma_{1r_i}^1), \cdots, (\sigma_{n_p 1}^1, \sigma_{n_p 2}^1 \cdots, \sigma_{n_p r_i}^1)), \cdots, \\ & ((\sigma_{11}^N, \sigma_{12}^N, \cdots, \sigma_{1r_i}^N), \cdots, (\sigma_{n_p 1}^N, \sigma_{n_p 2}^N \cdots, \sigma_{n_p r_i}^N)), \Delta t_p]. \end{aligned} \tag{21}$$

In HPSOGA, the position and velocity are randomly initialized.

The objective function can be calculated as follows: Considering the time-optimal control constraints, the extended objective function is defined as:

$$\begin{aligned} J_{\text{extend}} = \min_{\mathbf{\Omega}, \Delta t_p} \{ & (n_p \Delta t_p) + \sigma^* \hat{g}_1(\mathbf{\Omega}, \Delta t) \\ & + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} [\sigma_{ij} \max(0, D_{\text{safe}} - d^{i,j}(x_i(t), x_j(t))) \\ & + \sigma_{ij}^{'} \max(0, d^{i,j}(x_i(t), x_j(t)) - D_{\text{comm}})] \}, \end{aligned} \tag{22}$$

where σ_{ij} and $\sigma_{ij}^{'}$ are the safety distance punishment coefficient and communication distance punishment coefficient, respectively. σ^* is the punishment coefficient of the terminal constraint. As long as σ_{ij}, $\sigma_{ij}^{'}$, and σ^* are large enough (must be a positive number), the primitive objective function expression (17) and the constraint conditions (12) (13) (19) will be equivalent to (22). The fitness function in GA is $f = J_{\text{extend}}$.

Therefore, the formation reconfiguration is classified as a constrained parametric optimization problem with non-linear objective function and constraint functions. This can be solved with a standard non-linear programming method HPSOGA, though the global optimality of the solution depends upon the convexity of the objective function and constraint functions.

V. Hybrid Algorithm Description

Based on the above description, HPSOGA can solve formation reconfiguration problem. The algorithm can be divided into two stages, the PSO stage and the GA stage. The solutions can be found by the following steps:

Step 1: Initialize M particles randomly, the max iteration time *Ncmax*, and the parameters in HPSOGA. The crossover probability and mutation probability are 0.9 and 0.05 respectively.

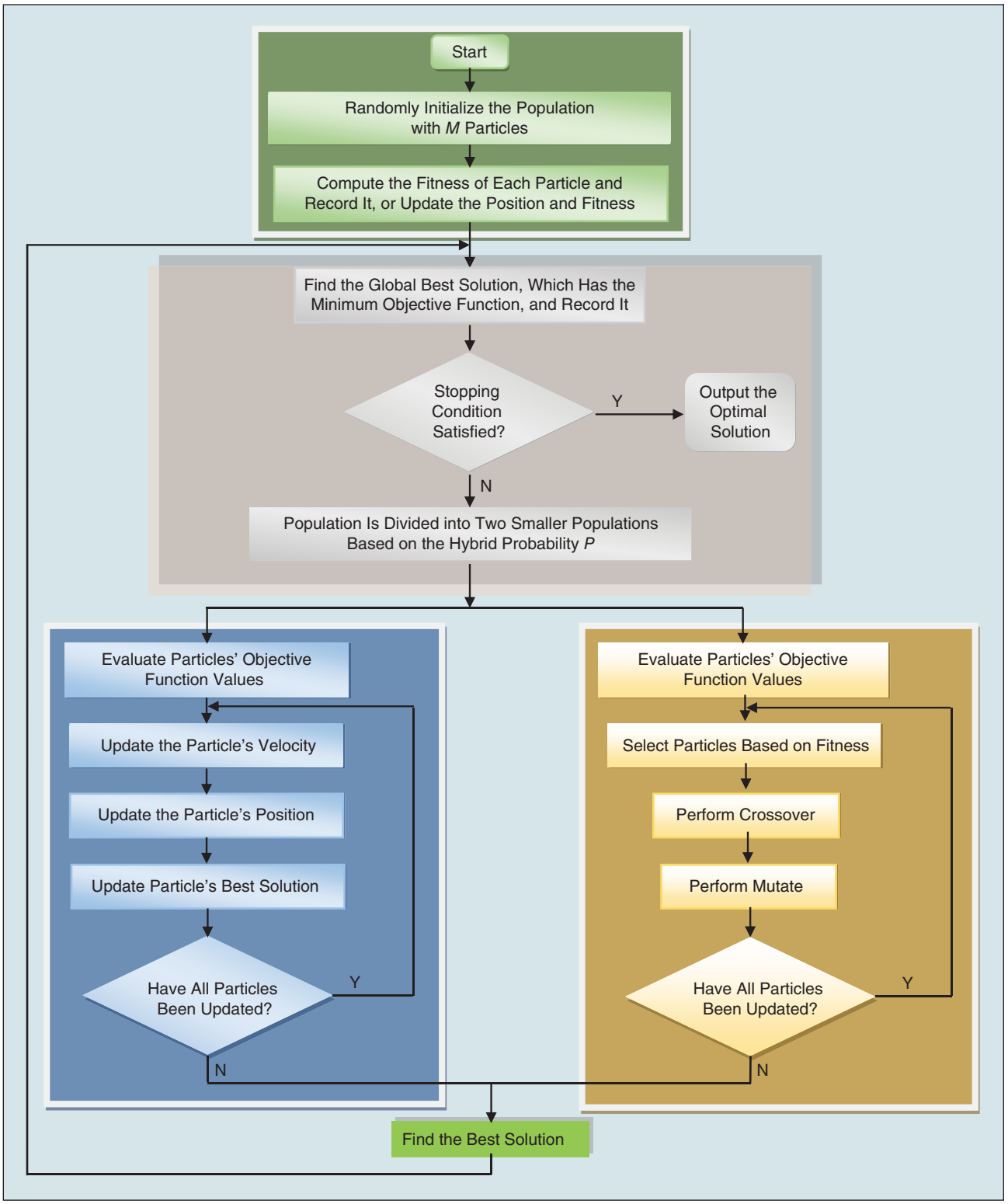

FIGURE 2 The detailed flow chart of HPSOGA.

Step 2: Calculate the objective function values of all particles, store the position of the particle with the minimum objective function value as the global best particle.

Step 3: Divide the particles into two groups based on the hybrid probability P, one group, with an expected size of $M * P$, uses PSO to update their positions and the other group uses GA.

Step 4: PSO stage. Particles update their velocities and positions according to (23) and (24). Two pseudo-random sequences, $r_1 \sim U(0,1)$ and $r_2 \sim U(0,1)$ are used to effect the

TABLE 1 State at initial and terminal time.

STATE VARIABLE	UAV1	UAV2	UAV3	UAV4	UAV5
INITIAL STATE XI (KM)	−12	−10	5	20	20
INITIAL STATE YI (KM)	10	30	50	30	10
INITIAL STATE ZI (KM)	2	4	6	8	10
RELATIVE STATE XT(KM)	−20	−10	0	−10	−20
RELATIVE STATE YT(KM)	20	10	0	−10	−20
RELATIVE STATE ZT(KM)	0	0	0	0	0

stochastic algorithm nature. For all dimensions $j \in 1....n$, let $x_{i,j}, v_{i,j}, \text{pbest}_{i,j}$ be the jth dimension of the current position, current velocity and current personal best position of the ith particle and $\text{Gbest}_{\text{pso}}$ is the global best position of the $M * P$ particles. F_i is the current personal best particle's objective function value and G_{pso} is the global best particle's objective function value. The velocity update step is:

$$v_{i,j} = wv_{i,j} + c_1 \cdot r_1 \cdot (\text{pbest}_{i,j} - x_{i,j}) + c_2 \cdot r_2 \cdot (\text{Gbest}_{\text{pso},j} - x_{i,j}), \quad (23)$$

where [39]

$$w = \left| \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \right|$$
$$\varphi = c_1 + c_2, \quad \varphi > 4.$$

The updated velocity is then added to the current position of the particle to obtain the new position:

$$x_{i,j} = x_{i,j} + v_{i,j}. \quad (24)$$

Then, we can get the new objective function value of x_i and record it as F_i'. If F_i' is less than F_i, the current personal best particle's objective function is F_i', and the current personal best position is the new position. If F_i' is less than G_{pso}, the global best particle's objective function value is F_i' and the global best position is the new position.

Step 5: GA stage. GA has three operators, namely selection, crossover, and mutation, described as follows:

1) Selection Operator

Roulette wheel selection strategy is widely used in GA because it can ensure that the selection probability of each particle is proportional to its fitness, i.e. the better a particle's fitness, the more likely it will be selected.

2) Crossover Operator

Crossover happens between two parents which are independently selected from the population. Children are created by the single-point crossover operation. It can be defined as follows:

$$P_1^{\text{new}} = \omega \cdot P_1 + (1 - \omega) \cdot P_2 \quad (25)$$
$$P_2^{\text{new}} = \omega \cdot P_2 + (1 - \omega) \cdot P_1, \quad (26)$$

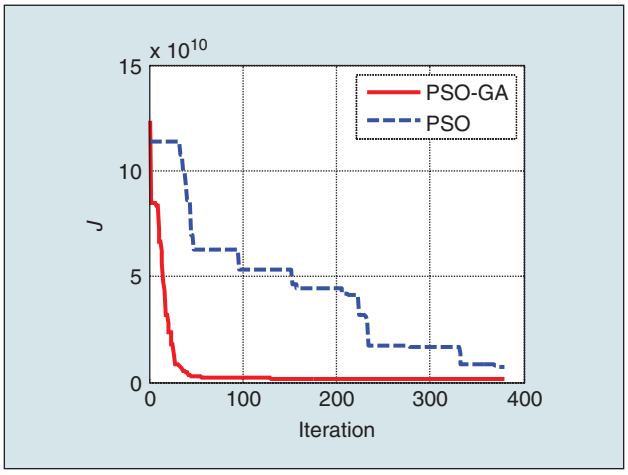

FIGURE 3 Comparison evolution cure of PSO and HPSOGA.

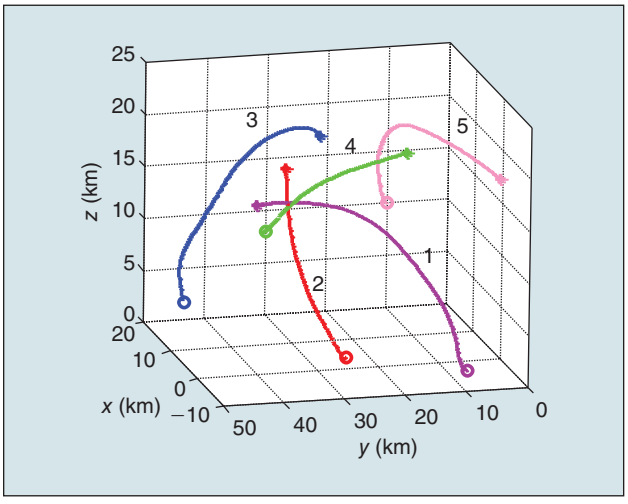

FIGURE 4 The reconfiguration trajectory obtained by HPSOGA.

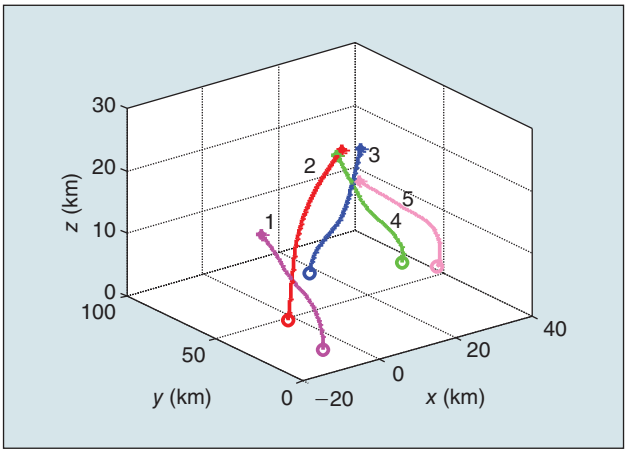

FIGURE 5 The reconfiguration trajectory obtained by PSO.

where P_1 and P_2 are parent particles, P_1^{new} and P_2^{new} are child particles, ω is a random number such that $\omega \in [0,1]$.

3) Mutation Operator

Mutation operator can maintain particle diversity and avoid premature convergence. It is executed on a particle which is selected based on its fitness. We adopt the adaptive acceleration mutation operator which can be defined as follows:

$$P_i^j(k+1) = P_i^j(k) + \beta \cdot \Delta P_i^j(k) + \rho \cdot sP_i^j(k), \tag{27}$$

where

$$\Delta P_i^j(k) = (P_i^{best}(k) - P_i^j(k)) \cdot |N(0,1)|$$
$$sP_i^j(k+1) = \beta \cdot \mathrm{acc}^j(k) \cdot \Delta P_i^j(k) + \rho \cdot sP_i^j(k).$$

$P_i^j(k)$ is the ith dimension of the jth particle in the kth generation, $P^{best}(k)$ is the best individual in the kth generation. ρ and β are the learning speed and inertia constant respectively, and they are set as 0.6 and 0.4 based on trial experiments. $N(0,1)$ is the normal random distribution function, and $\mathrm{acc}^j(k)$ is defined as follows:

$$\mathrm{acc}^j(k) = \begin{cases} 1, & \text{if new fitness greater than before} \\ 0, & \text{else.} \end{cases} \tag{28}$$

The position Gbest_{ga} and objective function value G_{ga} of the best particle that the GA can find are stored. The best objective function value is the reciprocal of the maximum fitness.

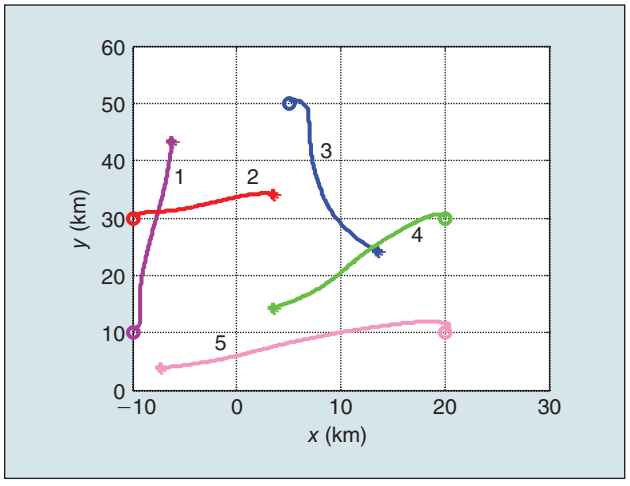

FIGURE 6 The horizontal trajectory obtained by HPSOGA.

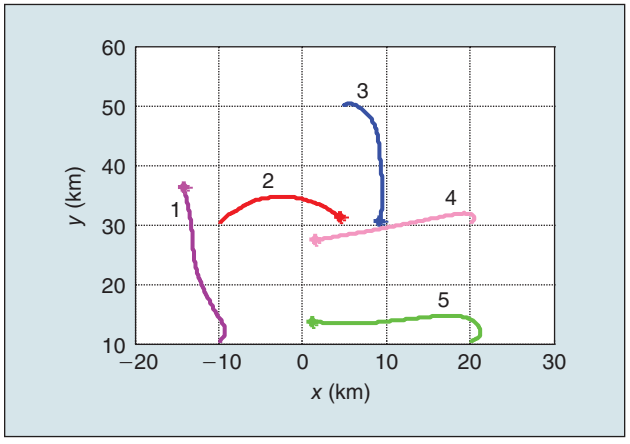

FIGURE 7 The horizontal trajectory obtained by PSO.

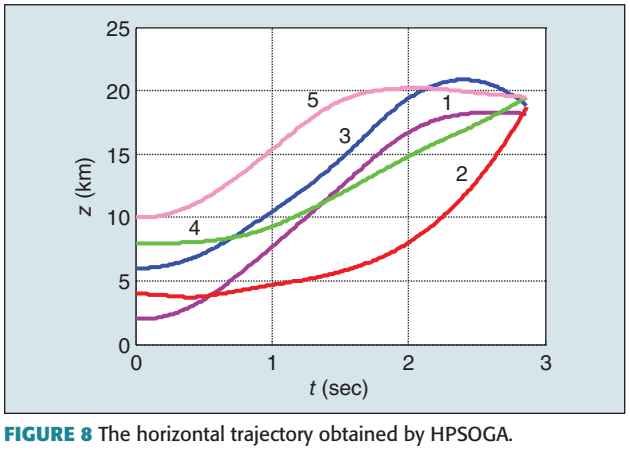

FIGURE 8 The horizontal trajectory obtained by HPSOGA.

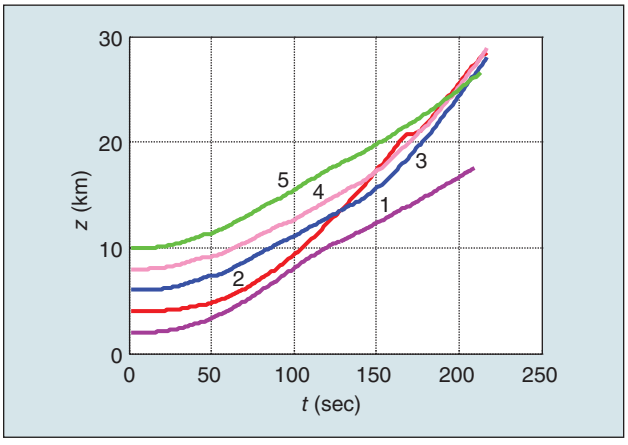

FIGURE 9 The horizontal trajectory obtained by PSO.

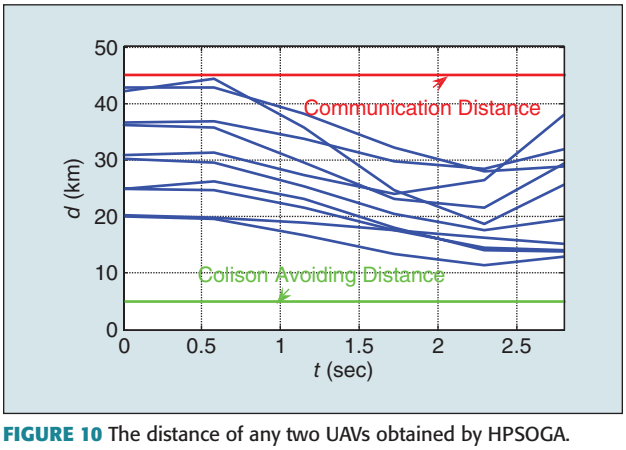

FIGURE 10 The distance of any two UAVs obtained by HPSOGA.

Step 6: Comparing G_{pso} and G_{ga}, if G_{pso} is less than G_{ga}, the global best position is $Gbest_{pso}$ and $Gbest_{ga}$ is replaced by $Gbest_{pso}$ and G_{ga} by G_{pso}. Else, the global best position is $Gbest_{ga}$, replace $Gbest_{pso}$ by $Gbest_{ga}$ and G_{pso} by G_{ga}.

Step 7: Repeat step 2 to step 6 until the ending condition is met.

The flow chart of our proposed HPSOGA is shown in Fig. 2.

VI. Experiments and Analysis

Considering N UAVs, the flight altitude of the ith UAV is denoted as $\chi_i = [v_i, \gamma_i, \chi_i, x_i, y_i, z_i]^T \in \Re^6, \quad \forall i \in \{1, \cdots, N\}$, where x_i, y_i, and z_i are the coordinates of the center of the ith UAV. The UAV model is simplified in this formation reconfiguration problem, and the outer loop variables such as thrust T_i, load factor n_i, and bank angle ϕ_i are chosen as control inputs (u_i) to each UAV. The equations of motion for the ith UAV are as follows [33]:

$$\dot{v}_i = g[(T_i - D_i)/W_i - \sin\gamma_i] \tag{29}$$

$$\dot{\gamma}_i = (g/v_i)(n_i\cos\phi_i - \cos\gamma_i) \tag{30}$$

$$\dot{\chi}_i = (gn_i\sin\phi_i)/(v_i\cos\gamma_i) \tag{31}$$

$$\dot{x}_i = v_i\cos\gamma_i\cos\chi_i \tag{32}$$

$$\dot{y}_i = v_i\cos\gamma_i\sin\chi_i \tag{33}$$

$$\dot{z}_i = -v_i\sin\gamma_i, \tag{34}$$

where D_i is the aerodynamic drag, γ_i is the flight path angle χ_i, is the heading angle, and W_i is the weight of ith vehicle, $i = 1, 2, \cdots, N$.

The goal of this problem is to find the continuous control inputs such that the UAVs, starting from an arbitrary initial state

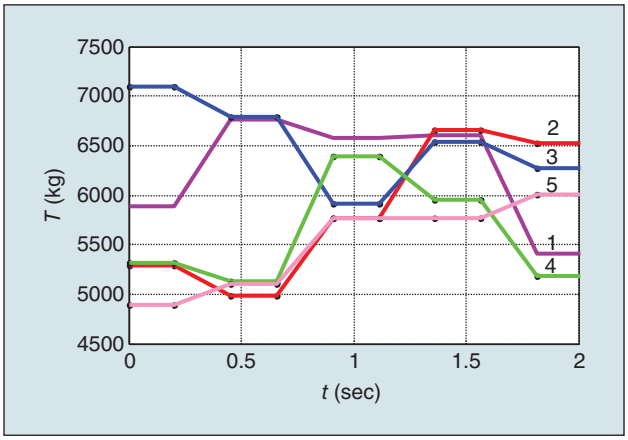

FIGURE 13 Computed optimal thrust force obtained by PSO.

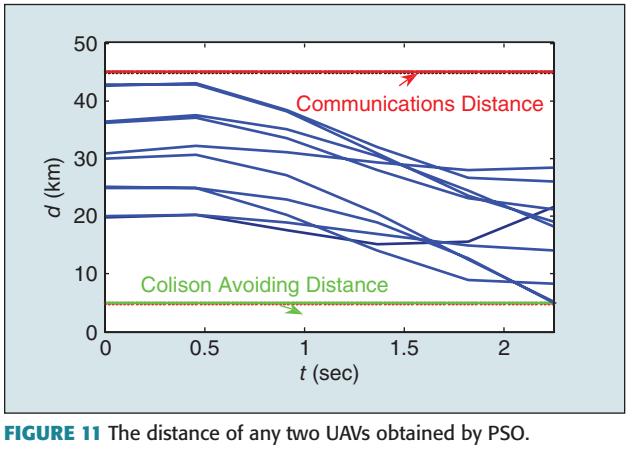

FIGURE 11 The distance of any two UAVs obtained by PSO.

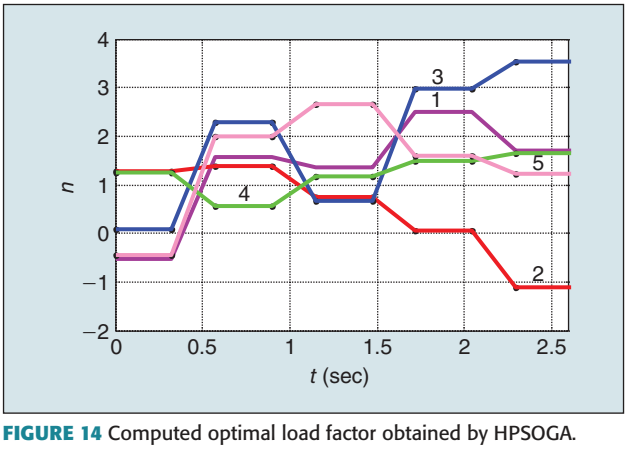

FIGURE 14 Computed optimal load factor obtained by HPSOGA.

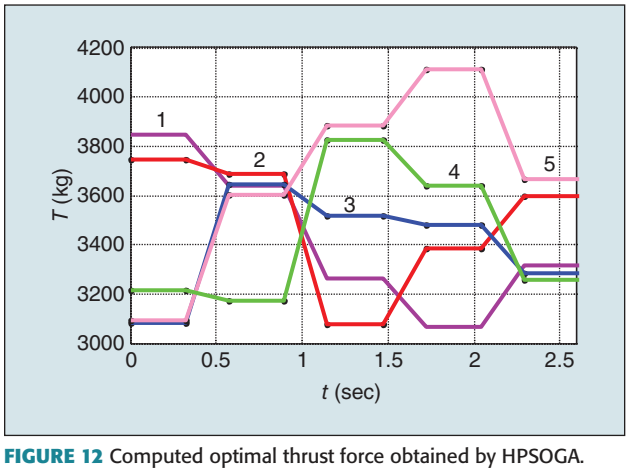

FIGURE 12 Computed optimal thrust force obtained by HPSOGA.

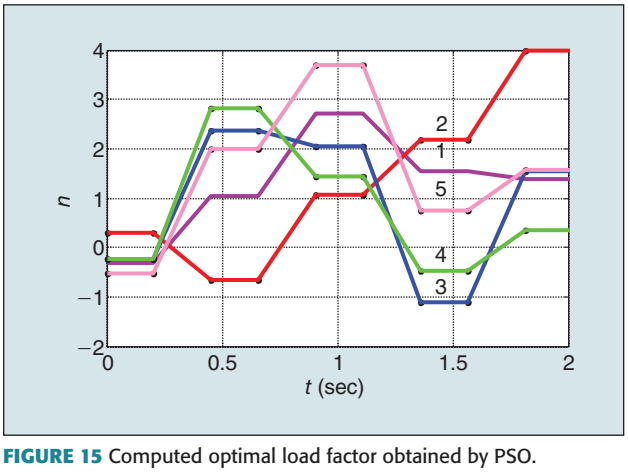

FIGURE 15 Computed optimal load factor obtained by PSO.

PSO and GA are global optimization algorithms and are suitable for solving optimization problems with linear or non-linear objective functions; Therefore, they are suitable for solving non-linear formation reconfiguration problem.

$(x_1(0) \neq, \ldots, \neq x_N(0))$, terminate with a relative formation at the optimal time T. The desired relative coordinates for each UAV are given with respect to those of a UAV located in the center. The relative coordinates of the central UAV are thus $[0, 0, 0]$.

In our experiment, $N = 5$. It means that there are five UAVs. $M = 400, Nc\max = 370, P = 0.5, c_1 = c_2 = 2.01$, the initial value of $w = 0.8, P_c = 0.9, P_m = 0.05, \rho = 1, \beta = 0.2$. Assume that the third vehicle is the center of the formation. $D_{\text{safe}} = 5\,(\text{km}), D_{\text{comm}} = 45\,(\text{km})$. The initial states of the UAVs are set randomly and the relative states at time $t = T$ are shown in Table 1. After the optimal control, the UAVs can move to the desired relative of V-shape formation at the same altitude.

Compared with the standard PSO, the experimental results show that our proposed HPSOGA can obtain better solutions.

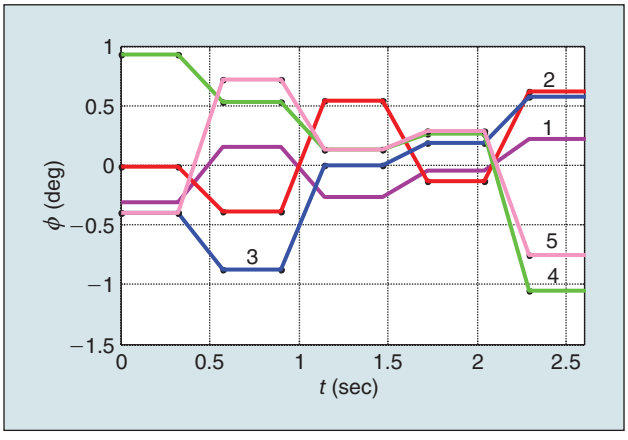

FIGURE 16 Computed optimal heading angle obtained by HPSOGA.

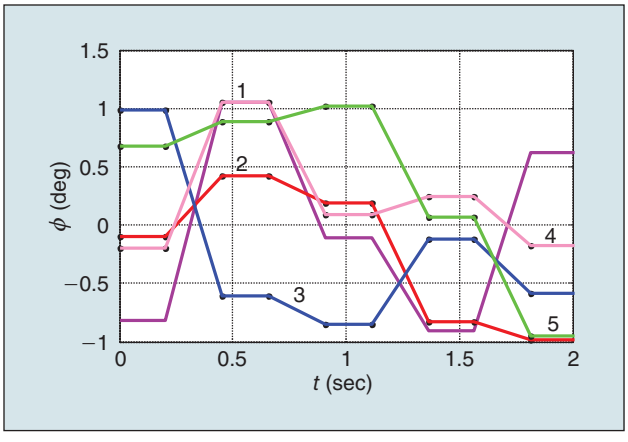

FIGURE 17 Computed optimal heading angle obtained by PSO.

Fig. 3 describes the relationship of the objective function and iteration count of PSO and HPSOGA. From Fig. 3, we can conclude that the HPSOGA performs better than PSO.

Fig. 4 shows the formation reconfiguration trajectory of the solution obtained by HPSOGA. The numbers indicate the flight trajectories of the respective UAVs. "o" is the initial state and "*" is the terminal state. Fig. 5 is the formation reconfiguration trajectory of the solution obtained by PSO.

Fig. 6 displays the horizontal trajectory of the solution obtained by HPSOGA, Fig. 7 gives the horizontal trajectory of the solution obtained by PSO. From Fig. 6, we can see that the vehicles have successfully moved to the desired relative V-shape formation. However, the UAVs failed to move to the desired relative V-shape formation in Fig. 7.

Fig. 8 shows the vertical trajectory of the solution obtained by HPSOGA, while Fig. 9 gives the vertical trajectory of the solution obtained by PSO. From Fig. 8, it can be seen that the UAVs are at almost the same altitude (although there are errors) at the end of the trajectory reconfiguration process but from Fig. 9, it is obvious that the UAVs are not at the same altitude.

The distance between any two UAVs for the solution obtained by HPSOGA is given in Fig. 10, and the corresponding results for the solution obtained by PSO are shown in Fig. 11. Fig. 10 illustrates that the distance between any two UAVs always satisfies the constraints (12) and (13), while from Fig. 11, it can be concluded that the PSO solution does not always satisfy the constraints (12) and (13).

Fig. 12 shows the computed optimal thrust forces for all the UAVs for the solution obtained by HPSOGA, while the computed optimal thrust forces for all the UAVs for the PSO solution are shown in Fig. 13.

Fig. 14 and Fig. 15 show the computed optimal load factors for all the UAVs of the solutions obtained by HPSOGA and PSO, respectively.

The computed optimal heading angles for all the UAVs of the solution obtained by HPSOGA are shown in Fig. 16. Fig. 17 gives the computed optimal heading angles for all the UAVs of the solution obtained by PSO.

From the simulation results, we can conclude that the UAVs based on the HPSOGA successfully move to the desired relative V-shape formation in 3-D space, while the PSO solution fails. This result indicates that the HPSOGA has higher search veracity, more rapid convergence speed, and stronger ability against precocity than PSO. The hybrid algorithm utilizes the advantages of both GA and PSO in solving optimization problems.

VII. Conclusions

Multi-UAV is often required to change its relative formation from one to another in the battlefield environment. In addition to the terminal state constraint and control action energy constraints, the constraints of safe distance to avoid collision and dependable distance to guarantee normal communication between each pair of UAVs are also considered. On the premise

of satisfying all the above constraint requirements, different optimal objective functions fit in with different multi-UAV formation reconfiguration optimization problems. A method to solve the multi-UAV formation reconfiguration problem in 3-D space has been presented. The problem was formulated as an optimization problem involving the minimization for a specified objective function with state relative constraints. The CPTD method has been used to solve this problem with a free terminal state constraint. Formation reconfiguration problem was focused on determining optimal control inputs for each UAV such that the group can start from the initial state and reach its final configuration at the optimal time while satisfying the set of constraints.

The experiments were also conducted to show that the proposed HPSOGA can successfully solve the optimal control for the multi-UAV formation reconfiguration problem in 3-D space. Given a random initial state and the target relative state at the terminal time, the HPSOGA can find the optimal solution to meet the objective function requirements and various formation system constraints to achieve formation reconfiguration in 3-D space. The HPSOGA presented in this paper is able to, not only solve the single-formation reconfiguration problem, the minimum energy control, and the shortest time and minimum energy integrated control problems, but also solve the centralized control of complex systems, such as multi-formation reconfiguration, multi-vehicle coordinate problems.

Acknowledgments

This work was partially supported by Natural Science Foundation of China (NSFC) under grant #61273054, 61273367, 60975072, and 60975080, National Key Basic Research Program of China under grant #2013CB035503, Program for New Century Excellent Talents in University of China under grant #NCET-10-0021, Top-Notch Young Talents Program of China, Fundamental Research Funds for the Central Universities of China, and Aeronautical Foundation of China under grant #20115151019.

References

[1] S. C. Felter and N. E. Wu, "A relative navigation system for formation flight," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 3, pp. 958–967, July 1997.

[2] J. R. Riehl, G. E. Collins, and J. P. Hespanha, "Cooperative search by UAV teams: A model predictive approach using dynamic graphs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 4, pp. 2637–2656, Oct. 2011.

[3] A. Franchi, C. Masone, H. H. Bulthoff, and P. R. Giordano, "Bilateral teleoperation of multiple UAVs with decentralized bearing-only formation control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, San Francisco, CA, 2011, pp. 2215–2222.

[4] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 5, pp. 1236–1251, Sept. 2012.

[5] M. Pachter, J. J. D'Azzo, and A. W. Proud, "Tight formation flight control," *J. Guid. Control Dyn.*, vol. 24, no. 2, pp. 246–255, Apr. 2001.

[6] H. B. Duan, Y. P. Zhang, and S. Q. Liu, "Multiple UAVs/UGVs heterogeneous coordinated technique based on receding horizon control (RHC) and velocity vector control," *Sci. China Technol. Sci.*, vol. 54, no. 4, pp. 869–876, Apr. 2011.

[7] M. Pachter, J. J. D'Azzo, and J. Dargan, "Automatic formation flight control," *J. Guid. Control Dyn.*, vol. 17, no. 6, pp. 1380–1383, Nov. 1994.

[8] H. B. Duan and S. Q. Liu, "Nonlinear dual-mode receding horizon control for multiple UAVs formation flight based on chaotic particle swarm optimization," *IET Control Theory Applicat.*, vol. 4, no. 11, pp. 2565–2578, Nov. 2010.

[9] S. S. Stankovic, M. J. Stanojevic, and D. D. Siljak, "Decentralized overlapping control of a platoon of vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 5, pp. 816–833, Sept. 2000.

[10] M. R. Anderson and A. C. Robbins, "Formation flight as a cooperative game," in *Proc. AIAA Guidance, Navigation Control Conf.*, Philadelphia, PA, 1998, pp. 244–251.

[11] R. Beard and F. Hadaegh, "Constellation templates: An approach to autonomous formation flying," in *Proc. World Automation Congr.*, Anchorage, AK, 1998, pp. 177.1–177.6.

[12] F. Giulietti, L. Pollini, and M. Innocenti, "Formation flight control: A behavioral approach," in *Proc. AIAA Guidance, Navigation Control Conf.*, Montreal, Canada, 2001, pp. 1–6.

[13] M. Asada, E. Uchibe, and K. Hosoda, "Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development," *Artif. Intell.*, vol. 110, no. 2, pp. 275–292, June 1999.

[14] T. Balch and R. C. Arkin, "Behavior-based formation control for multi robot teams," *IEEE Trans. Robot Autom.*, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[15] R. Dougherty, V. Ochoa, Z. Randles, and C. Kitts, "A behavioral control approach to formation-keeping through an obstacle field," in *Proc. IEEE Aerospace Conf.*, Big Sky, MT, 2004, pp. 168–175.

[16] L. Pollini, F. Giuletti, and M. Innocenti, "Robustness to communication failures within formation flight," in *Proc. American Control Conf.*, Anchorage, AK, 2002, pp. 2860–2866.

[17] S. Thomas, B. C. Chang, and H. G. Kwatny, "Reconfigurable control of aircraft with partial elevator," in *Proc. AIAA Guidance, Navigation, Control Conf. Exhibit*, Austin, TX, 2003, pp. 1–6.

[18] P. K. C. Wang and F. Y. Hadaegh, "Minimum-fuel formation reconfiguration of multiple free-flying spacecraft," *J. Astronaut. Sci.*, vol. 47, no. 2, pp. 77–102, 1999.

[19] P. K. C. Wang and F. Y. Hadaegh, "Optimal formation-reconfiguration for multiple spacecraft," in *Proc. AIAA Guidance, Navigation Control Conf.*, Boston, MA, 1998, pp. 686–696.

[20] R. Fierro and A. K. Das, "Hybrid control of reconfigurable robot formations," in *Proc. American Control Conf.*, Denver, CO, 2003, pp. 4607–4613.

[21] S. Y. Wang and C. W. Zheng, "A hierarchical evolutionary trajectory planner for spacecraft formation reconfiguration," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 279–289, Jan. 2012.

[22] L. Sauter and P. Palmer, "Onboard semianalytic approach to collision-free formation reconfiguration," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2638–2652, July 2012.

[23] G. F. Ma, H. B. Huang, and Y. F. Zhuang, "Time optimal trajectory planning for reconfiguration of satellite formation with collision avoidance," in *Proc. 8th IEEE Int. Conf. Control Automation*, Xiamen, China, 2010, pp. 476–479.

[24] T. Paul, T. R. Krogstad, and J. T. Gravdahl, "UAV formation flight using 3D potential field," in *Proc. 16th Mediterranean Conf. Control Automation*, Ajaccio, France, 2008, pp. 1240–1245.

[25] P. J. Werbos, "Computational intelligence for the smart grid-history, challenges, and opportunities," *IEEE Comput. Intell. Mag.*, vol. 6, no. 3, pp. 14–21, Aug. 2011.

[26] C. Gueret, S. Schlobach, K. Dentler, M. Schut, and G. Eiben, "Evolutionary and swarm computing for the semantic web," *IEEE Comput. Intell. Mag.*, vol. 7, no. 2, pp. 16–31, May 2012.

[27] H. Muhleisen and K. Dentler, "Large-scale storage and reasoning for semantic data using swarms," *IEEE Comput. Intell. Mag.*, vol. 7, no. 2, pp. 32–44, May 2012.

[28] Y. Meng, Y. Y. Zhang, and Y. C. Jin, "Autonomous self-reconfiguration of modular robots by evolving a hierarchical mechanochemical model," *IEEE Comput. Intell. Mag.*, vol. 6, no. 1, pp. 43–54, Feb. 2011.

[29] M. N. Le, Y. S. Ong, Y. C. Jin, and B. Sendhoff, "A unified framework for symbiosis of evolutionary mechanisms with application to water clusters potential model design," *IEEE Comput. Intell. Mag.*, vol. 7, no. 1, pp. 20–35, Feb. 2012.

[30] Y. M. B. Ali, "An augmented particle swarm model based bi-acceleration factor," *Int. J. Intell. Comput. Cybern.*, vol. 4, no. 2, pp. 187–206, July 2011.

[31] D. S. Guo and Y. N. Zhang, "Novel recurrent neural network for time-varying problems solving," *IEEE Comput. Intell. Mag.*, vol. 7, no. 4, pp. 61–65, Nov. 2012.

[32] T. Furukawa, H. F. Durrant-Whyte, F. Bourganlt, and G. Dissanayake, "Time-optimal coordinated control of the relative formation of multiple vehicles," in *Proc. IEEE Int. Symp. Computational Intelligence Robotics Automation*, Kobe, Japan, 2003, pp. 259–264.

[33] W. Xiong, Z. J. Chen, and R. Zhou, "Optimization for multiple flight vehicles formation reconfiguration using hybrid genetic algorithm," in *Proc. 1st Chinese Guidance, Navigation Control Conf.*, Beijing, China, 2007, pp. 501–506.

[34] J. H. Holland, *Adaptation in Natural and Artificial System*. Ann Arbor, MI: Univ. of Michigan Press, 1975, pp. 20–60.

[35] J. Nanbo and Y. Rahmat-Samii, "Hybrid real-binary particle swarm optimization (HPSO) in engineering electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 58, no. 12, pp. 3786–3794, Dec. 2010.

[36] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. 1995 IEEE Int. Conf. Neural Networks*, Perth, Australia, 1995, pp. 1942–1948.

[37] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro-Machine Human Science*, Nagoya, Japan, 1995, pp. 39–43.

[38] Y. H. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. 1998 IEEE Int. Conf. Evolutionary Computation*, Anchorage, AK, 1998, pp. 69–73.

[39] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.