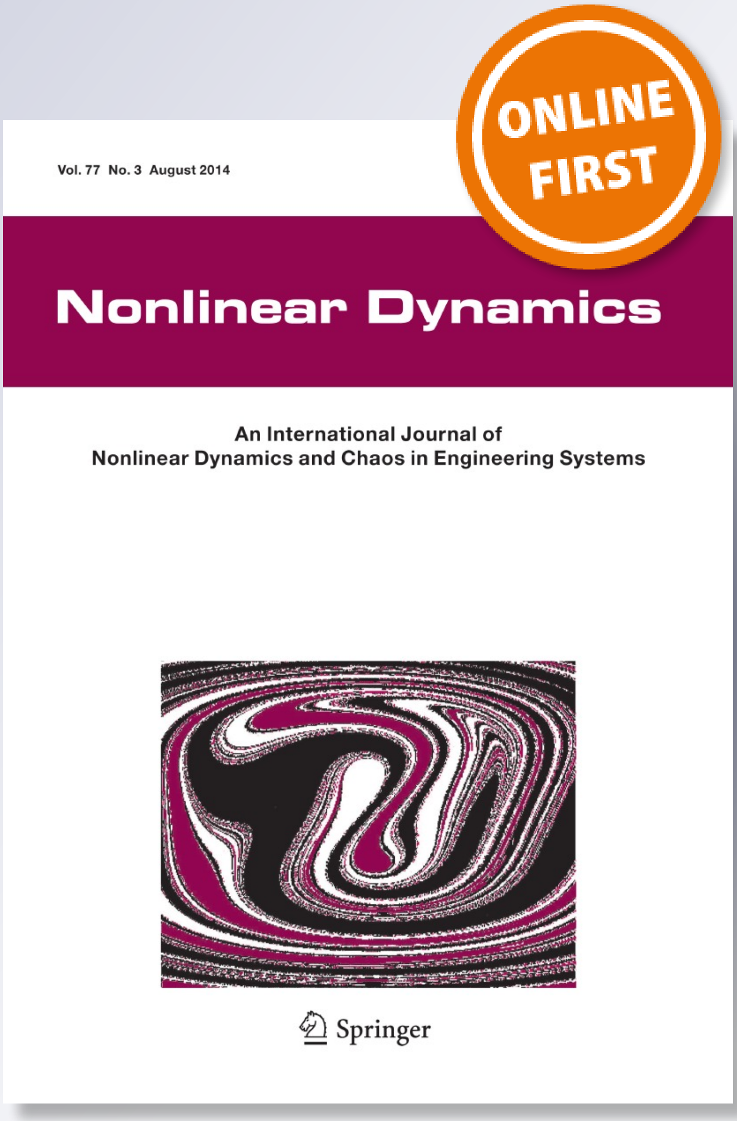
Receding horizon control for multiple UAV formation flight based on modified brain storm optimization

Huaxin Qiu & Haibin Duan

Nonlinear Dynamics
An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems

ISSN 0924-090X

Nonlinear Dyn
DOI 10.1007/s11071-014-1579-7


ONLINE FIRST
Vol. 77 No. 3 August 2014
Nonlinear Dynamics
An International Journal of
Nonlinear Dynamics and Chaos in Engineering Systems
Springer

Springer

Springer

Nonlinear Dyn
DOI 10.1007/s11071-014-1579-7

ORIGINAL PAPER

Receding horizon control for multiple UAV formation flight based on modified brain storm optimization

Huaxin Qiu · Haibin Duan

Received: 29 January 2014 / Accepted: 3 July 2014

Abstract Formation flight for unmanned aerial vehicles (UAVs) is a rather complicated global optimum problem. In the global optimum problem, the complex relationship between the controller parameters and the performance index, and the different kinds of constrains under complex combat field environment are taken into account. Brain storm optimization (BSO) is a brand-new swarm intelligence optimization algorithm inspired by a human being's behavior of brainstorming. In this paper, in allusion to the drawbacks that the basic BSO algorithm traps into local optimum easily and has a slow convergent speed, some novel designs are proposed to enhance the performance of the optimization algorithm. The modified BSO is applied to solve the optimization problem based on the nonlinear Receding horizon control (RHC) mode of UAVs to seek the RHC control parameters for UAV formation flight. Series of comparative experimental results are presented to show the feasibility, validity, and superiority of our proposed method.

Keywords Unmanned aerial vehicles (UAVs) · Formation · Receding horizon control (RHC) · Brain storm optimization (BSO)

H. Qiu · H. Duan (✉)
State Key Laboratory of Virtual Reality Technology and Systems, School of Automation Science and Electrical Engineering, Beihang University (BUAA),
Beijing 100191, People's Republic of China
e-mail: hbduan@buaa.edu.cn

1 Introduction

Unmanned aerial vehicles (UAVs) are attracting the interest of many researchers all over the world [1]. This popularity may be attributed to deep studies in theoretical analysis [2,3] and potential use in many applications such as search and rescue missions, surveillance, law enforcement, inspection, mapping, and aerial cinematography.

Compared with a single UAV, formation of the UAVs can leverage the capabilities of the team to have more effective performance in missions such as cooperative simultaneous localization and mapping (SLAM), coverage and recognizance, and security patrol [4]. Hence, recent years have seen an increasing interest in the study of UAV formation control from both theoretical and experimental points of view [5–7].

In the literature, many methods have been applied to the formation keeping control. Zhang and Liu [8] combined Kalman filter with PID controller to design UAV formation flight control system. Xie et al. [9] proposed two nonlinear robust formation control algorithms to solve the problem of designing nonlinear robust formation controllers on a team of UAV using off-the-shelf autopilots. Paul et al. [10] presented a solution for formation flight and formation reconfiguration of UAVs. It is based on a virtual leader approach and combined with an extended local potential field. However, these approaches all have their own strength and weakness, and are more suitable for some applications and less for others [11].

Published online: 22 July 2014

In this paper, we adopt the RHC approach to solve the formation control problem for fixed-wing UAVs. RHC is an optimization-based control method originating in process industry in the early 1970s [12]. Recently, RHC is utilized to achieve formation fight and other cooperative tasks. The main thought of RHC is online receding/moving optimization, which is based on the simple idea of repetitive solution of an optimal control problem and state updating after the first input of the optimal command sequence. It breaks the global control problem into several local optimization problems of smaller sizes, and thus can decrease the computing complexity and computational expense significantly. In the formation control problem, RHC is an effective method of solving constrained optimization with the following advantages [13]:

(1) Different control objectives can be achieved by changing appropriate terms in the cost function (e.g., formation keeping, formation joining, and flying) [14];
(2) The RHC strategy can adapt to the change of the conditions (e.g., topography, threat source, and internal instructions);
(3) The RHC control has ability to deal with control input constraints and system state constraints, such as the flight state constraints of UAVs.

The selection of the RHC control parameters is a very tough problem in the formation control, on account of the existence of strong coupling among the inputs, and the nonexistence of mapping relationship between the performance index and the controller parameters [15]. Some researchers try to use intelligent algorithms to solve this problem, for instance, particle swarm optimization (PSO) [16], artificial bee colony (ABC) algorithm, differential evolution (DE) [17], and so on. Unfortunately, as the complexity of optimization problem, these intelligent algorithms become slightly prohibitive under the block on local optimum and dissatisfactory convergence rate.

A novel brainstorm optimization algorithm named brain storm optimization (BSO) was first introduced by Shi in [18] in recent years, which was inspired by the human brain storming process other than the optimization algorithms inspired by collective behavior of insects like ants, bee, etc. BSO generally uses the grouping, replacing, and creating operators to produce ideas as many as possible to approach the problem global optimum generation by generation [19].

In this paper, owning to its better performance of global exploration than others, the thoughts of BSO are applied to the control field to optimize the RHC control parameters in UAV formation control problem, to minimize the value of the cost function. Some modified designs taking aim at the problem of local optimum and slow convergence rate are proposed to enhance the conventional BSO performance.

The rest of the paper is organized as follows. Section 2 proposes the formation control scheme based on RHC, covering the selection of UAV model, the design of the cost function, and the overview of RHC formation controller. Section 3 focuses on three aspects of the specific improvement measures in the modified BSO. Simulation validations, together with comparison against the basic BSO and PSO, are presented in Sect. 4, and our concluding remarks are drawn in Sect. 5.

2 UAV formation control problem based on RHC

2.1 Principle of RHC

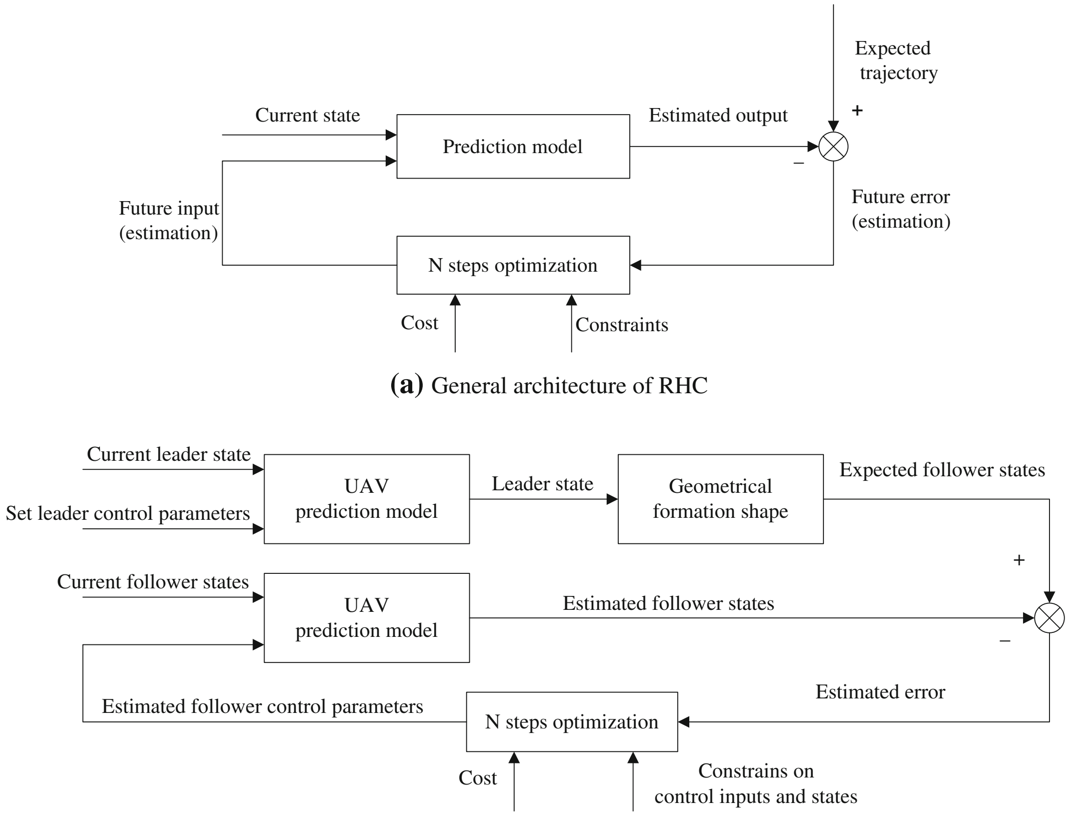
RHC, which has the advantage of online processing constraints on control input and output, describes the control problem as a constrained optimization problem of finite time [13]. In a sampling time, by seeking to minimize the estimated objective over a fixed time interval, subject to the estimated dynamics and constraints, the optimal control input sequence in estimated time domain can be obtained; the first item in the sequence is chosen as the RHC input, while the rest items discarded;

then, the system states change under the action of the control inputs; at the next time step, the process is repeated, with updated estimates of the current state and future quantities; these repeated, and the system recording horizon control is implemented. General architecture of RHC is shown in Fig. 1a.

2.2 UAV model

2.2.1 Kinematic and dynamic models

In this paper, point-mass aircraft model [20] is used to describe the motion of formation flying UAVs. The related variables are defined with respect to the inertial coordinate frame $(\hat{x}, \hat{y}, \hat{h})$ and are shown in Fig. 2.

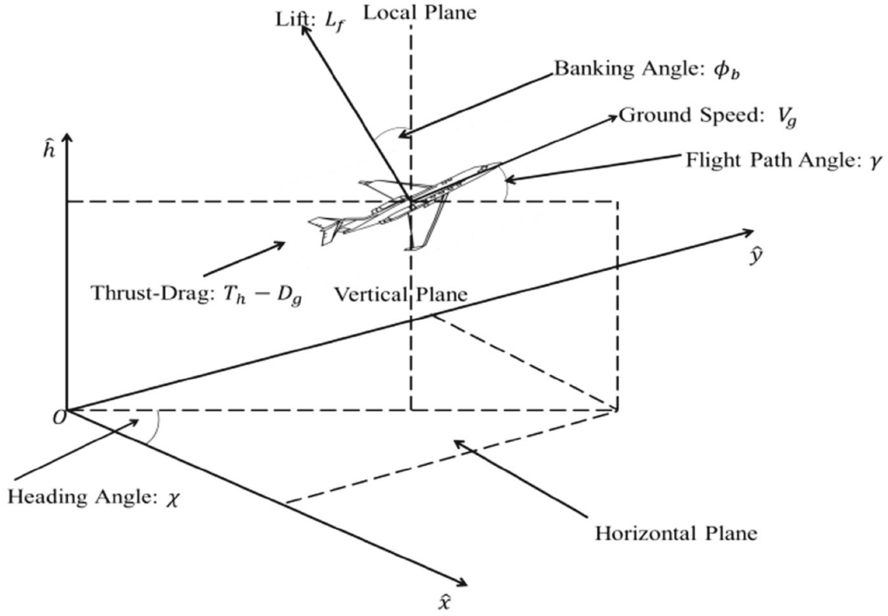


(a) General architecture of RHC

(b) RHC formation controller process

Fig. 1 Process of RHC

Fig. 2 UAV model



In what follows, the model assumes that the aircraft thrust is directed along the velocity vector, and that the aircraft always performs coordinated maneuvers. It is also assumed that the Earth is flat, and the fuel expenditure is negligible (i.e., the center of mass is time-invariant). Under these assumptions, the UAV equations of motion can be described by the following equations:

$$\begin{aligned} \dot{x}_i &= V_{gi} \cos\gamma_i \cos\chi_i \\ \dot{y}_i &= V_{gi} \cos\gamma_i \sin\chi_i \\ \dot{h}_i &= V_{gi} \sin\gamma_i \end{aligned} \tag{1}$$

where $i = 1, \ldots, n$ is the index of multiple UAVs and n is the number of UAVs. For UAV i, x_i is the downrange displacement, y_i is the cross-range displacement, h_i is the altitude, V_{gi} is the ground speed, γ_i is the flight-path angle, and χ_i is the heading angle. The UAV dynamics are given by the following equations:

$$\begin{aligned} \dot{V}_{gi} &= \tfrac{T_{hi}-D_{gi}}{m_i} - g_a \sin\gamma_i \\ \dot{\gamma}_i &= \tfrac{g_a}{V_{gi}}(n_{gi}\cos\phi_{bi} - \cos\gamma_i) \\ \dot{\chi}_i &= \tfrac{L_{fi}\sin\phi_{bi}}{m_i V_{gi}\cos\gamma_i} \end{aligned} \tag{2}$$

where T_{hi} is the engine thrust,D_{gi} is the drag, m$_i$ is the mass,g_a is the acceleration due to gravity, L_{fi} is the vehicle lift, and ϕ_{bi} is the banking angle. The control variables in the UAVs are the g-load $n_{gi} = L_{fi}/(g_a m_i)$ controlled by the elevator, the banking angle ϕ_{bi} controlled by the combination of rudder and ailerons, and the engine thrust T_{hi} controlled by the throttle. Throughout the formation control process, the control variables will be constrained to remain within their respective limits.

The reduced UAV models can be recast in a matrix form:

$$\dot{X} = AX + BU \tag{3}$$

with $X = [x_1, \ldots, x_n], U = [u_1, \ldots, u_n], u_i = [T_{hi}, n_{gi}, \phi_{bi}]^T, x_i = [x_i,\ y_i, h_i, V_{gi}, \gamma_i, \chi_i]^T, i=1, \ldots, n$, where$X \in R^{6\times n}$ and $U \in R^{3\times n}$ are, respectively, the aggregate states and aggregate control inputs of all UAVs.

2.2.2 *Estimated model*

Under the assumption of bounds on control variables and flight speed [13], the above-mentioned UAV model (3) can be also expressed as the following equations:

$$x_{k+1} = Ax_k + Bu_k \tag{4}$$

Estimated value for the system states can be obtained by iterating Eq. (4):

$$x_{k+i|k} = A^i x_{k|\text{k}} + \sum_{j=0}^{i-1} A^{i-1-j} B u_{k+j|\text{k}}, \quad i = 1, 2, \ldots, N \tag{5}$$

where $x_{k+i|k}$ is the estimated state over the time interval $k + I$ at time k. When $i = 0$, $x_{k|\text{k}} = x_k$ is the current state estimate apparently. In order to facilitate the study, the estimated states and respective control inputs over the time interval $k + 1, \ldots, k + N$ at time k can be recast in a vector form:

$$\tilde{x} = \left(x_{k+1|\text{k}}, x_{k+2|\text{k}}, \ldots, x_{k+N|\text{k}}\right)^T,$$
$$\tilde{u} = \left(u_{k|\text{k}}, u_{k+1|\text{k}}, \ldots, u_{k+N-1|\text{k}}\right)^T.$$

Thus, an expression only using current state x_k and control input u can be obtained.

$$\tilde{x} = H_x x_k + H_u \tilde{u} \tag{6}$$

with $H_x = \left(A, A^2, \ldots, A^N\right)^T$ and

$$H_u = \begin{bmatrix} B & 0 & \cdots 0 \\ AB & B & \cdots \vdots \\ \vdots & \vdots & B \\ A^{N-1}B & A^{N-2}B & \cdots AB \end{bmatrix}.$$

2.3 Cost function design

The target of formation control is to maintain the relationship among unmanned machines in a fixed position shape, meanwhile to make the required control input smaller; therefore, the cost terms of system states and inputs should be included in the cost function [13]. The cost function needs to be designed in current time domain in order to coordinate RHC with optimal solution over a fixed time interval from current time. Choose 2-norm as the basic item of cost function; assume that the length of fixed time interval is N, and then the cost function of follower RHC controllers is the following equation:

$$\begin{aligned} \hat{J}_{QP} = \sum_{i=0}^{N-1} &((x_{k+i+1|k} - x_{\text{ref},k+i+1})^T Q(x_{k+i+1|k} \\ &-x_{\text{ref},k+i+1}) + (u_{k+i|k} - u_{\text{ref},k+i})^T \\ &R(u_{k+i|k} - u_{\text{ref},k+i})) \end{aligned} \tag{7}$$

where $x_{\text{ref},k+i+1}$ is the expected state of system,$u_{\text{ref},k+i}$ is expected control input, and $u_{k+i|k}$ is the estimated control input in future time $k+I$ at current time k.

The weighting matrixes P and Q are positive matrix. That is to say that $P = P^T > 0$ and $Q = Q^T > 0$. In order to get smaller control, set the desired control input to be zero. Substituting Eq. (6) in Eq. (7), we can obtain the following equations:

$$\begin{aligned}\widehat{J_{QP}} &= \tilde{x}^T\tilde{Q}\tilde{x} + \tilde{u}^T\tilde{R}\tilde{u} - 2\tilde{x}^T\tilde{Q}\tilde{x}_{\text{ref}} + \tilde{x}_{\text{ref}}^T\tilde{Q}\tilde{x}_{\text{ref}} \\ &= \tilde{u}^T(H_u^T\tilde{Q}H_u + \tilde{R})\tilde{u} + 2(H_x x_k - \tilde{x}_{\text{ref}})\tilde{Q}H_u\tilde{u} \\ &\quad + x_k^T H_x^T\tilde{Q}H_x x_k - 2x_k^T H_x^T\tilde{Q}\tilde{x}_{\text{ref}} + \tilde{x}_{\text{ref}}^T\tilde{Q}\tilde{x}_{\text{ref}}\end{aligned} \tag{8}$$

where $\tilde{Q} = \text{diag}\{Q, \ldots, Q\}$, $\tilde{R} = \text{diag}\{R, \ldots, R\}$ and $\tilde{x}_{\text{ref}} = (x_{\text{ref},k+1}, x_{\text{ref},k+2}, \ldots, x_{\text{ref},k+N})^T$.

The first item of the Eq. (8) is the quadratic term of $\tilde{u}$, the second item is the monomial term of $\tilde{u}$, and the last three items are the absolute term. The absolute terms can be ignored when making optimal solution. Then, we can get the standard quadric form:

$$\begin{aligned}J_{QP} &= \tilde{u}^T(H_u^T\tilde{Q}H_u + \tilde{R})\tilde{u} \\ &\quad + 2(H_x x_k - \tilde{x}_{\text{ref}})\tilde{Q}H_u\tilde{u}.\end{aligned} \tag{9}$$

From Eq. (9), the cost function J_{QP} is the function of the initial state x_k and the control input sequence $\tilde{u}$ which is the decision variable. For every certain initial state x_k, by solving the quadratic programming problem, we may get a group of optimal control input sequence.

2.4 RHC formation controller

The elementary process of RHC formation controller is presented in the following exposition [13]:

Step 1 At sampling time k, the follower state is x_0. To solve the optimization problem described in Eq. (9), obtain the optimal control input sequence in future N steps.

Step 2 The first item in the sequence is chosen as the follower RHC inputs, while the rest N-1 items discarded.

Step 3 The system of follower reaches a new state x_1 at sampling time $k+1$ under the action of the control inputs.

Step 4 Regard current time and follower state as k and x_0 respective. Return **Step 1**.

For each sampling time, the optimal decision variable of the optimal problems $\tilde{u}$ is only concerned with the initial state x_k, i.e., there is a state feedback control law $\tilde{u}^* = f(x_k)$, but there is no analysis relationship between the system state and the control input. The RHC formation controller process can be illustrated by Fig. 1b.

3 Modified BSO

3.1 Basic BSO

BSO has been successfully applied to generate ideas to solve very difficult and challenging problems [18]. As presented in [18] and [21], the process of BSO can be described as follows [22]. First, N ideas are randomly generated within the searching space, denoted as $X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$, where $i = 1, 2, \ldots, N$, and D is the dimension of the optimization problem to be solved. Each dimension signifies one design variable. Then, each idea is evaluated, and its fitness value $J(X_i)$ is obtained. The process of iteration begins afterwards.

During each generation, the N ideas are clustered into K cluster according to the positions, and the best idea in each cluster is recorded as the cluster. Then, a cluster is randomly selected with a probability of p_{6a}, and the cluster center is replaced with a randomly generated idea.

In the creating operation, BSO first randomly choose one cluster or two. After that, the selected idea(s) is updated according to the following equation:

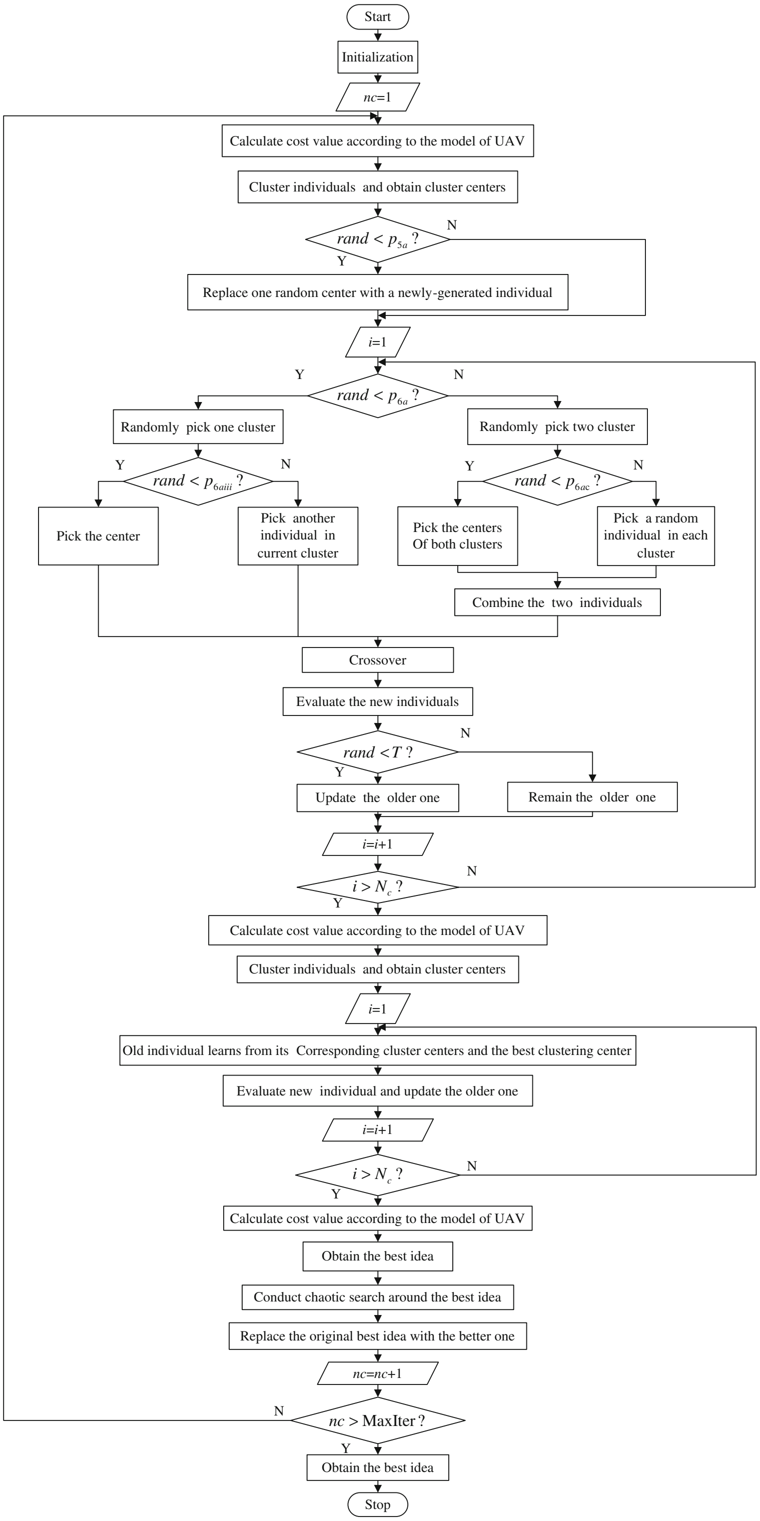
$$\begin{aligned}&x_{\text{new}} = x_{\text{old}} + \xi N(\mu, \sigma) \\ &x_{\text{old}} = \begin{cases} x_{ij}, & \text{one cluster} \\ \omega_1 x_{i1,j} + \omega_2 x_{i2,j}, & \text{two cluster}\end{cases}\end{aligned} \tag{10}$$

where $N(\mu, \sigma)$ is the Gaussian random value with mean μ and variance σ.ω_1 and ω_2 are weight values of the two ideas. ξ is an adjusting factor slowing the convergence speed down as the evolution goes, which can be expressed as the following equation:

$$\xi = \log\text{sig}\left(\frac{0.5 \times Nc_{\max} - Nc}{K}\right) \times \text{random}(0, 1), \tag{11}$$

where r is a random value between 0 and 1.$Nc_{\max}$ and Nc denote the maximum number of iteration and current number of iteration, respectively, whereas K

Fig. 3 Modified BSO process


Start
Initialization
nc=1
Calculate cost value according to the model of UAV
Cluster individuals and obtain cluster centers
rand < p_{5a} ?
N
Y
Replace one random center with a newly-generated individual
i=1
Y
N
rand < p_{6a} ?
Randomly pick one cluster
Randomly pick two cluster
Y
N
rand < p_{6aiii} ?
Y
N
rand < p_{6ac} ?
Pick the center
Pick another individual in current cluster
Pick the centers Of both clusters
Pick a random individual in each cluster
Combine the two individuals
Crossover
Evaluate the new individuals
N
rand < T ?
Y
Update the older one
Remain the older one
i=i+1
N
i > N_c ?
Y
Calculate cost value according to the model of UAV
Cluster individuals and obtain cluster centers
i=1
Old individual learns from its Corresponding cluster centers and the best clustering center
Evaluate new individual and update the older one
i=i+1
i > N_c ?
N
Y
Calculate cost value according to the model of UAV
Obtain the best idea
Conduct chaotic search around the best idea
Replace the original best idea with the better one
nc=nc+1
N
nc > MaxIter ?
Y
Obtain the best idea
Stop

adjusts the slope of the log sig function. Such form of ξ facilitates global searching ability at the beginning of the evolution and enhances local searching ability when the process is approaching to the end.

After the new idea is created, a crossover between the new one and the old one is conducted. The two ideas generated by crossover, together with the old one and the created one, are evaluated, and the old one is replaced with the best of the four.

The process above repeats until N ideas are updated. Thus, one generation is finished. The iteration goes until terminal requirement is met. Then, the best idea is output as the optimal solution to the problem.

3.2 Modified BSO

Taking a deep sight into the process of BSO described above, several improvements deserve to be taken into consideration.

3.2.1 New clustering method

First, the clustering method can be improved by sorting fitness values. In basic BSO, during each generation, the N ideas are clustered into K cluster completely according to the positions. Two clustering methods have been proposed, which are k-means clustering method and SGM [19]. There is no denying that these methods give expression to prodigious randomness thus give security to diversity. However, it makes no difference in generating ideas. Hence, a brand-new clustering standard can be tried. In this paper, fitness value is taken into account as a new clustering standard. In the Modified BSO, the facilitator is implemented by fitness value clustering method as the following steps:

Step 1 Randomly produce K random positive integers with a constraint that the sum of these integers is constant N .These K random positive integers are denoted as the number of individuals in each cluster $nr_i (1 \leq i \leq K)$.

Step 2 Calculate the fitness value $J(X_i)$ for each idea $X_i (1 \leq i \leq N)$ and sort values in ascending or descending order (It depends on the expected fitness value, maximum or minimum) to obtain J_{order} and Ind_{order}, where $J_{\text{order}} \in R1 \times N,$ is the form of $J(X_i)$ in ascending or descending order, and $Ind_{\text{order}} \in R1 \times N$, is the sequence of original index for each idea X_i when X_i is sorted by $J(X_i)$ in order.

Step 3 Assign the idea $X_{Ind_{\text{order}}(m)}$ into the i group, where $m = \left(\sum_{j=1}^{i-1} nr_j + 1\right), \ldots, \left(\sum_{j=1}^{i} nr_j\right)$, $i = 1, \ldots, K$, and set $nr_0 = 0$. Meanwhile, choose the idea $X_{Ind_{\text{order}}\left(\sum_{j=1}^{i-1} nr_j + 1\right)}$ as the center of i group.

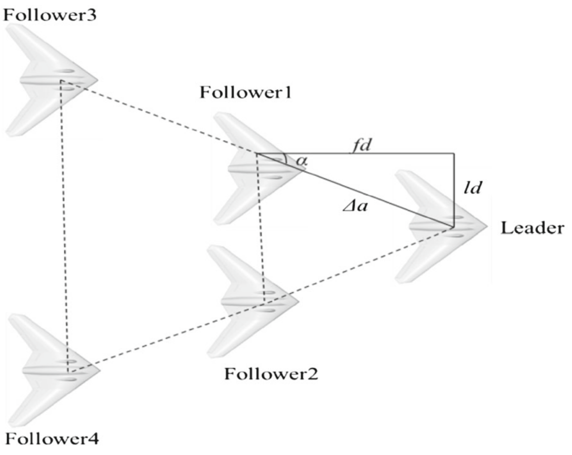


Fig. 4 Multiple UAV formation

Generally speaking, above-mentioned clustering method is to classify ideas by fitness value in essence. In the actual situation, this clustering method has relevance to the behavior of human beings. People with the same thoughts level easily reach a consensus to gather together. And in their respective groups, the owner of better idea has more possibility to become the clustering center.

It is worthwhile to specify that this method does not show any contrariness to Osborn's original rules for idea generation in a Brainstorming process. In other words, the diversity of basic algorithm is not destroyed. Specifically, it is just to provide some convenience for later generating steps.

3.3 Local optimal solution

In the process of simulation, BSO is trapped in local optimal problems, the same as PSO and ABC. In this paper, two kinds of solutions to this problem are proposed.

The first method is based on the principle of probability updating. In basic BSO algorithm process, the two ideas generated by cross, together with the old one and the created one, are evaluated and the old one is replaced with the best of the four [22]. For the following exposition, we assume that the smaller value of the cost function is better. Probability updating is to replace the old one with the best one in a probability [23]:

$$T(X_i \to X_j) = \frac{1}{1 + \exp[(\omega_j - \omega_i)/\kappa]} \tag{12}$$

where κ characterizes the intensity of updating, ω_i represents the cost value of the old one, and ω_j represents the cost value of best one. For $\kappa = 0$, X_i will be replaced with X_j deterministically when $\omega_j < \omega_i$. For $\kappa > 0$, ideas performing worse are also remained with a certain probability. In most cases, better ideas prefer to replace the worse one. In line with most previous studies, we set κ to be 0.1 (strong updating).

Specific replacement formula is as follows:

$$X_i = \begin{cases} X_j & \text{rand} \le T \\ X_i & \text{rand} > T \end{cases} . \tag{13}$$

This method fully reflects inclusive of group and appropriate humanistic cares to worse ideas; therefore, it meets the need of the required diversity of algorithm.

In addition, slightly traditional method is chaos. Chaos is the highly unstable motion of deterministic systems in finite phase space which often exists in nonlinear systems [24]. In the well-known logistic equation,

$$x_{n+1} = 4x_n(1 - x_n) \tag{14}$$

where $0 < x_n < 1$, a very small difference in the initial value of x would give rise to large difference in its long-time behavior, which is the basic characteristic of chaos. The track of chaotic variable can travel ergodically over the whole space of interest. The variation of the chaotic variable has a delicate inherent rule in spite of the fact that its variation looks like in disorder. Therefore, after each search round, we can conduct the chaotic search in the neighborhood of the current optimal parameters by listing a certain number of new generated parameters through chaotic process. In this way, we can make use of the ergodicity and irregularity of the chaotic variable to help the algorithm to jump out of the local optimum as well as finding

Table 1 Optimization parameters

		Variables	Description	Value
Objective variable		J	Cost function value	–
Design variables	Control variables	T_{hi} (N)	Engine thrust	[10,100]
		n_{gi}	g-load ($n_{gi} = L_{fi}/(g_a m_i)$)	[−5,5]
		ϕ_{bi} (rad)	Banking angle	[$-\pi$,π]
Fixed variables	Simulation parameters	Countmax	RHC simulation cycles	30
		dt (s)	Simulation sampling time	0.1
	RHC parameters	P_rhc	The length of prediction horizon	3
		M_rhc	The length of control horizon	1
	UAV number	leader_num	Leader number	1
		fol_num	Follower number	4
	Single UAV parameters	D_{gi}(N)	Drag	50
		m_i (kg)	Mass	30
		g$_a$ (m/s^2)	The acceleration due to gravity	9.8
	Leader control parameters	T_{h1}(N)	The leader engine thrust	50
		n_g	The leader g-load	1
		ϕ_b (rad)	The leader banking angle	0
	Formation	*fd* (m)	The forward distance	90
		ld (m)	The lateral distance	120
		α (rad)	The formation angle arctan (*ld*/*fd*)	0.6435
		Δa (m)	The distance between two UAVs$\sqrt{ld^2 + fd^2}$	150

Table 2 Constraints

Variables	Description	Value
V_{gi} (m/s)	The ground speed	V_{gi} ≤80
T_{hi} (N)	Engine thrust	[10,100]
n_{gi}	$g-\text{load}(n_{gi}=L_{fi}/(g_a m_i))$	[−5,5]
ϕ_{bi} (rad)	Banking angle	$[-\pi,\pi]$

Table 3 Initial states of UAVs

Variables	Description	Leader	Follower1	Follower2	Follower3	Follower4
x_i (m)	The down-range displacement	0	600	−200	−600	−100
y_i (m)	The cross-range displacement	0	−800	0	150	400
h_i (m)	The altitude	300	450	500	200	100
V_{gi} (m/s)	The ground speed	50	50	50	50	50
γ_i (rad)	The flight-path angle	0	0	0	0	0
χ_i (rad)	The heading angle	0	0	0	0	0

Table 4 Control parameters of BSO and the modified BSO

Parameter	Description	Value	Used in
Maxiter	Maximum times of iteration	100	BSO & modified BSO
N	Number of ideas	100	
K	Number of clusters	5	
p_{5a}	Probability to directly update a cluster center	0.2	
p_{6b}	Probability to choose one cluster	0.8	
p_{6biii}	Probability to select the center of the selected clusters	0.4	
p_{6c}	Probability to select the center of the two selected clusters	0.5	
k	Parameter to change logsig() function's slope	20	
μ	The mean of the Gaussian random function	0	
σ	The mean of the Gaussian random function	1	
c_1	Respective cluster center learning factor	2	Modified BSO
c_2	Best cluster center learning factor	2	
$c(1)$	the first basic characteristic of chaos	0.3	

Table 5 Control parameters of PSO

Parameter	Description	Value
Maxiter	Maximum times of iteration	100
N	Number of particles	100
ω	Inertia weight	0.5
c_1	Self best factor	2
c_2	Global best factor	2

Table 6 Simulation results of leader

	x_i (m)	y_i (m)	h_i (m)	V_{gi} (m/s)	γ_i (rad)	χ_i (rad)
Leader	1500	0	300	50	0	0

Table 7 Simulation results of J

	Reference	BSO	Modified BSO	PSO
J	0	16.6325585968166	1.36484003451963	31.4662159746113

Table 8 Simulation results of x_i, y_i and h_i by different methods

Variables	x_i (m)				y_i (m)				h_i (m)			
Followers	1	2	3	4	1	2	3	4	1	2	3	4
Reference	1380	1380	1260	1260	90	−90	180	−180	300	300	300	300
Modified BSO	1379.822024	1379.876048	1260.659155	1259.438421	90.861822221	−90.242945499	180.0300319	−180.1586782	299.2351017	298.9679791	299.1533295	300.5887949
BSO	1381.976176	1380.635933	1258.770519	1259.805868	91.97348217	−90.97077408	179.3915757	−180.9137312	299.2994533	301.5668045	298.8565829	298.6239736
PSO	1384.801249	1383.222658	1254.027895	1260.84047	88.54320153	−89.81670471	175.3830038	−178.1692543	300.9664408	302.4490125	302.4061132	287.1061563

Table 9 Simulation results of V_{gi}, γ_i and χ_i by different methods

Variables	V_{gi} (m/s)				γ_i (rad)				χ_i (rad)			
Followers	1	2	3	4	1	2	3	4	1	2	3	4
Reference	50	50	50	50	0	0	0	0	0	0	0	0
Modified BSO	49.66773873	49.08415388	50.49143031	49.63391434	−0.354467982	−0.026852112	−0.398504634	−0.152252768	0.0690046361	−0.027463961	−0.014137389	−0.132536469
BSO	49.9003334	48.0263383	49.8173607	49.9051872	−0.305631494	−0.124023716	−0.294716526	−0.114041354	0.46450392	−0.079428875	−0.173055545	0.0564277824
PSO	57.64190063	47.35137817	53.70761734	51.88572922	−0.211470435	−0.269143292	−0.212847832	−0.012105491	−0.028435018	−0.609230002	−0.327915937	−0.230367961

the optimal parameters. The specific procedure is as follows:

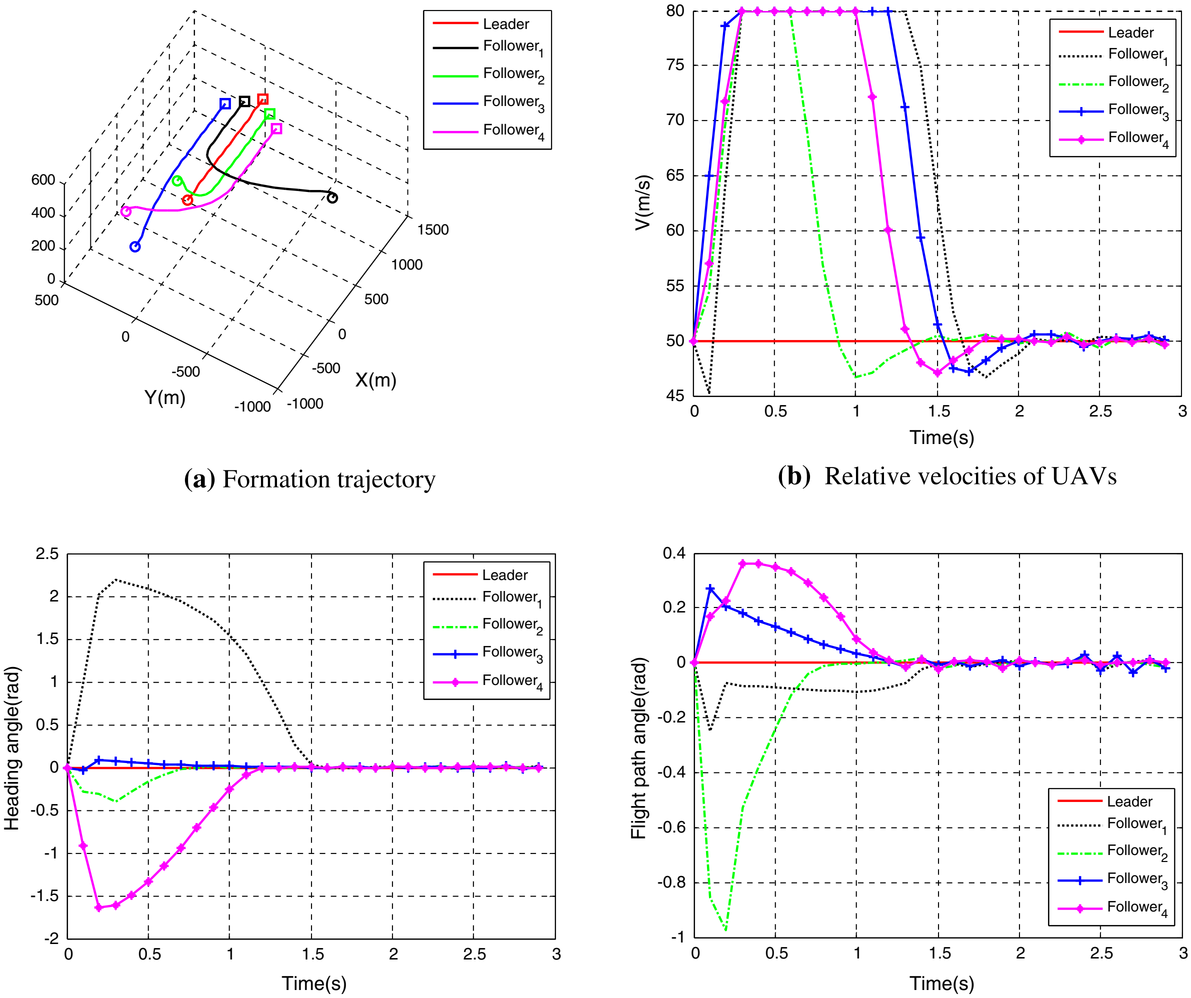
Step 1 Conduct the chaotic search around the best idea X_i parameters based on Eq. (13) after transforming the parameters ranges into (0, 1).

Step 2 Among the engendered series of solutions, select the best one and use it to replace the former best ideas.

3.3.1 Learning behavior

Gazing at the entire idea of BSO algorithm, we will notice that the algorithm tries to pursue diversity while it loses sight of the purpose to find the optimal solution as soon as possible.

During each generation, lots of ideas can and should be based on ideas already generated [18]. Any generated idea can and should serve as a clue to generate more ideas. Picking up several good ideas from ideas



(a) Formation trajectory

(b) Relative velocities of UAVs

(c) Relative heading angles

(d) Relative flight path angles

Fig. 5 Detailed results generated by the control sequence optimized by the modified BSO

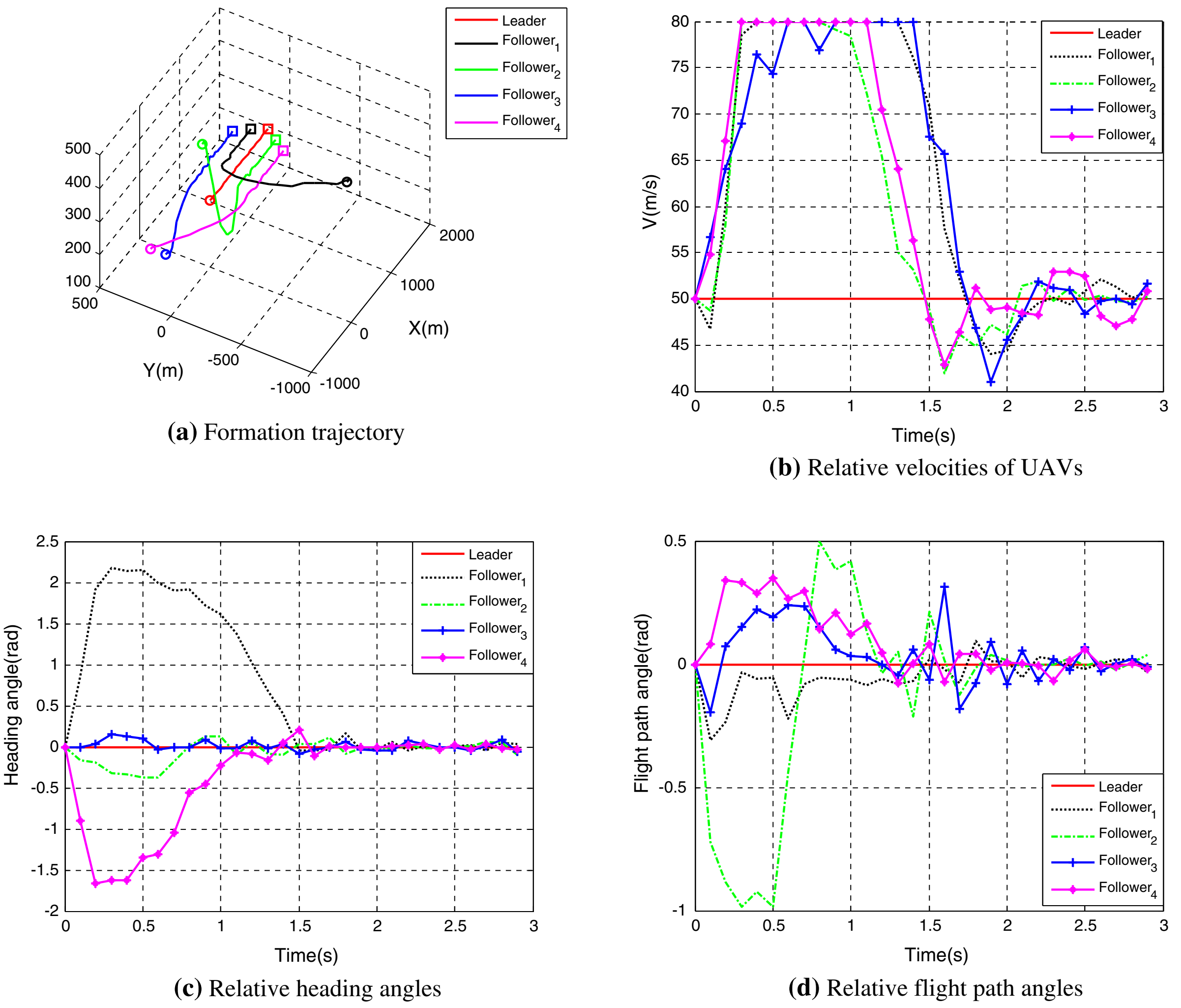
generated so far is to cause the brainstorming group to pay more attention to the better ideas which the brainstorming group to pay more attention to the better ideas which the brainstorming group believes to be. The ideas picked up work like point attractors for the idea generation process. In the specific process of the algorithm, generally, cluster centers are selected with higher probability than other ideas in the creating and updating section; therefore, the cluster centers are treated with higher priority [22].

In a nutshell, the cluster centers with better ideas just cause enough attention but not play guiding roles. In a group, the center not only deserves to be paid closer attention to, but also is duty bound to play the role of appropriately guiding other individuals. Namely, other individuals should learn to the center of the group. Taking example by PSO, the individual updates its ideas according to the following equations [25]:

$$\begin{aligned} X_i^{\text{new}} &= X_i^{\text{old}} + V_i \\ V_i &= \omega V_i + c_1 r_1 \left(P_{id} - X_i^{\text{old}}\right) + c_2 r_2 \left(P_{gd} - X_i^{\text{old}}\right) \end{aligned} \quad (15)$$

where P_{id} is the idea of the m group center, m is the group with i, P_{gd} is the best idea among K centers at present, ω is the inertia weight, c_1 and c_2 are positive constants, and r_1 and r_2 are two random numbers in the range [0,1], $1 \leq i \leq N_c$. In this paper, we define $\omega = 0.5$, $c_1 = c_2 = 2$, and we restrict X_i in the range $[0, 1]^D$ and V_i in the range $[0, 0.1]^D$.

The modified BSO process can be illustrated by Fig. 3.



(a) Formation trajectory

(b) Relative velocities of UAVs

(c) Relative heading angles

(d) Relative flight path angles

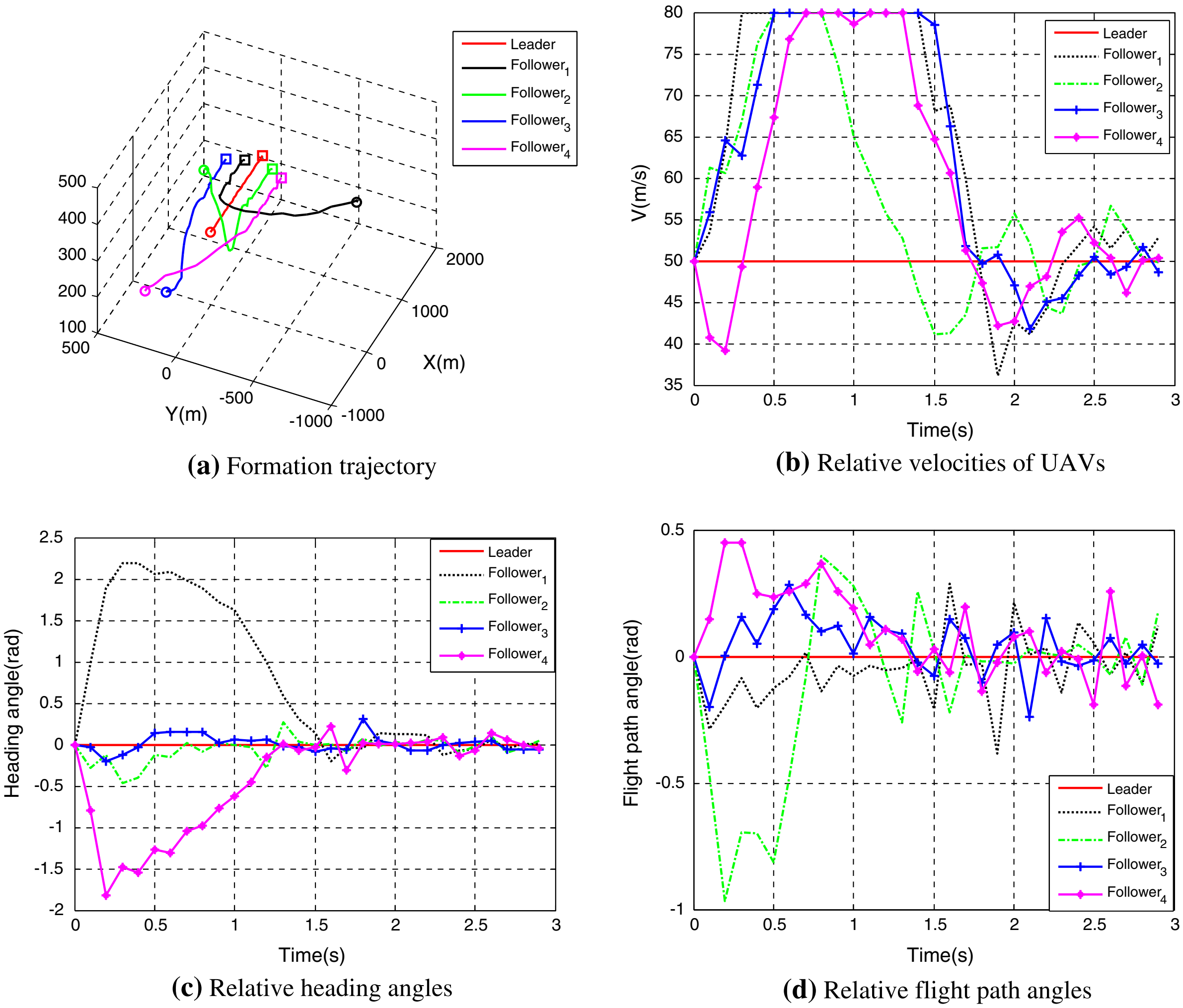
Fig. 6 Detailed results generated by the control sequence optimized by BSO

4 Comparative experimental results

In a typical multiple UAV formation flight, the followers follow the trajectory of the leader UAV, taking other aircrafts as reference to keep its own position inside the formation [16]. In a large formation, intra-aircraft distances must be kept constant. The formation model in this paper adopts leader-mode strategy (as shown in Fig. 4), which means each follower UAV takes its trajectory references from the leader UAV, while the altitude is the same for all. The leader UAV takes charge of formation trajectory.

To test the effectiveness of the modified algorithm, the optimization for a model of UAV formation is chosen as the benchmark [22]. 30 RHC circles were conducted in the process of each simulation. In a single RHC circle, algorithm optimization loops 100 times. As shown in Table 1, this benchmark problem has three design variables and one cost function value. Our object is to minimize the cost function value indirectly determined by these design variables with the constraints shown in Table 2 at the initial states shown in Table 3. The performances of the modified algorithm are compared with the basic BSO and a popular population-based algorithm called PSO. The control parameters of BSO and the Modified BSO are given in Table 4. The control parameters of PSO modeled after [25] are given in Table 5.

In order to clearly illustrate the performance of the modified BSO in single RHC process, the evolution



(a) Formation trajectory

(b) Relative velocities of UAVs

(c) Relative heading angles

(d) Relative flight path angles

Fig. 7 Detailed results generated by the control sequence optimized by PSO

curve after 100 generations in the first round of the RHC simulation cycle compared with basic BSO and PSO is displayed in Fig. 8a. As we shall see, due to probability updating and chaos optimization, the modified BSO exhibits better capability to jump out of local optimum [22]. When the individuals converge to a local optimum, indicated by the horizontal retention parts of the evolution curve, the modified BSO takes much less generations to jump out of the local optimum and find a position with smaller cost function value.

From the below location of the modified BSO always, it also gives a satisfactory performance on the final optimization result in single RHC simulation process. It is obvious that the learning behavior toward to their respective cluster center and the best cluster center is crucially involved in speeding up to seek the optimal solution within the global scope.

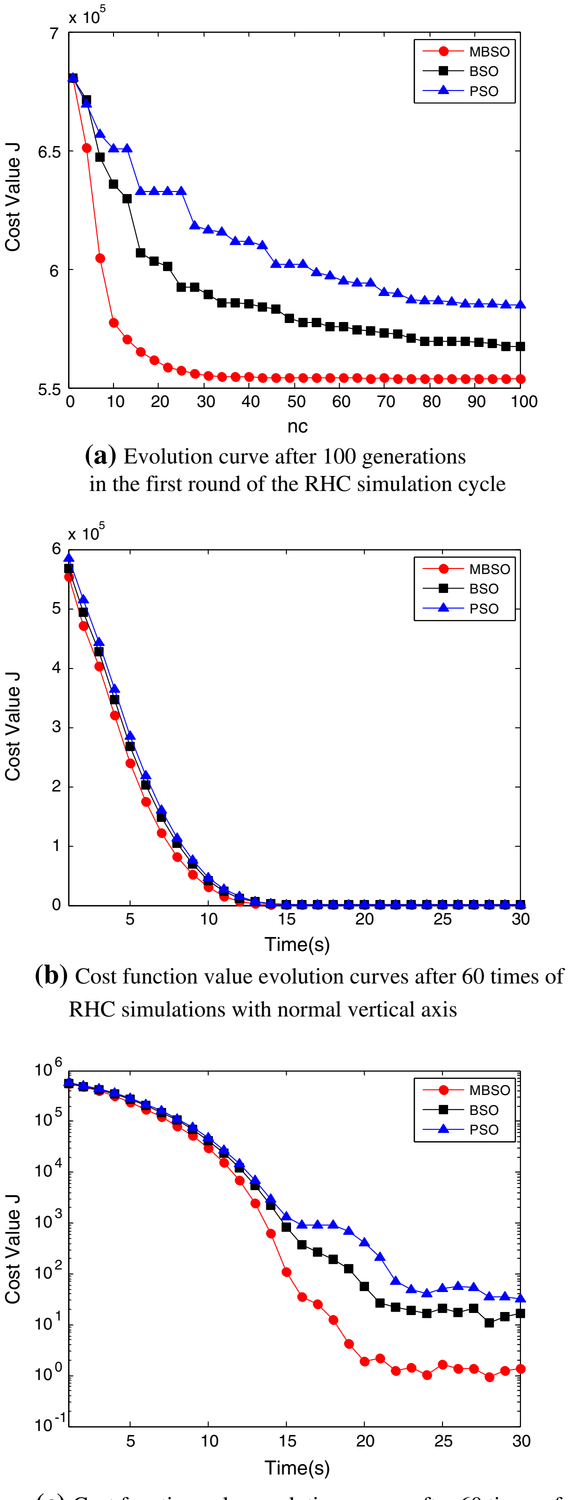
The final results after 30 RHC simulations are presented in Tables 6–9. Without loss of generality, in the simulation, we set the leader to level off at an even speed in a straight line. From data in the table, we can conclude that either the modified BSO or basic BSO or PSO can complete the optimization goal to form a comparative stable flight formation. The down-range displacement x_i, the cross-range displacement y_i, the altitude h_i, the ground speed V_{gi}, the flight-path angle γ_i, and the heading angle χ_i of four followers are in line with reference values relatively.

Figures 5–7 show the detailed results generated by the control sequence optimized by three algorithms, respectively, in which (a)–(d) describe three-dimensional formation trajectories, relative velocities, heading angles, and flight-path angles of UAVs.

As shown in (c) of Figs. 5–7, each RHC control parameters solved by three algorithms can make the relative heading angles χ_i of followers tend to coincide with leader at about 1.5 s with minor fluctuations. However, the performance of three algorithms on velocities V_{gi} and flight-path angles γ_i is quite different. Observed from Fig. 5b, d, in the presence of the modified BSO, the relative velocities and flight-path angles of UAVs are close to accordance at about 2 and 1.5 s, respectively. The performance of basic BSO is inferior to the modified algorithm. The velocities of followers in Fig. 6b fluctuate irregularly within 5 m/s amplitude, and the flight-path angles of followers in Fig. 6d can track the sates of leader in stable at 2 s, 0.5 s behind the modified BSO. PSO, by contrast, is defeated due to fluctuation shown in Fig. 7b, d with larger amplitude and higher frequency.

From the three-dimensional formation trajectories of UAVs in (a) of Figs. 5–7, the aggregate performance of three algorithms can be observed directly. It is evident that the follower 2 in (a) of Figs. 6 and 7 selected a more awkward path before converging to the designated position, while at the help of the modified algorithm, it can find a relatively better way shown in Fig. 5a to reach a desired position and keep formation. The formation trajectories in Fig. 5a have better stability, faster convergence rate, and better tracking ability, indicated by more smooth curve and longer stage of keeping formation, than the other two.

According to the above analysis, the modified BSO's optimization purposes to form stable formation has a basic implementation. In order to see the difference among three algorithms clearly and straightforward, the cost function value evolution curves after 30 times of RHC simulations (as shown in Fig. 8b, c) is presented to compare the performance measured by numerical values. From Fig. 8b, the evolution curve of the modified BSO is always below the other two algorithms. In other words, the modified algorithm can find better control parameters in each RHC simulation cycle; therefore, the result of last RHC simulation cycle will provide a good starting point for the next cycle. In Fig. 8c, the logarithmic axes are chosen as the vertical axis to facilitate the analysis of formation stability and tracking ability. The evolution curve of the mod-



(a) Evolution curve after 100 generations in the first round of the RHC simulation cycle

(b) Cost function value evolution curves after 60 times of RHC simulations with normal vertical axis

(c) Cost function value evolution curves after 60 times of RHC simulations with logarithmic vertical axis

Fig. 8 Evolution curve

ified algorithm converges to a smallest steady value with the fastest speed than the others, which means that UAVs get in a relatively stable formation at a rapid speed, and a better consistency exists in UAVs group.

To sum up, given credit to the aforementioned improvement measures, the modified BSO does embody certain superiority in jumping out of local optimum and speeding up the optimization process in the set off of basic BSO and PSO.

5 Conclusions

This paper presents a modified BSO approach for formation flight optimization problem based on the nonlinear RHC mode of UAVs. The proposed modified algorithm enhances the basic BSO performance in jumping out of local optimum and speeding up the rate of convergence from the following aspects of improvement:

(1) A new clustering method by sorting cost function value;
(2) Probability updating which means to replace the old one with the best one in a probability;
(3) Conduct chaotic search around the best one utilizing the ergodicity and irregularity of the chaotic variables;
(4) Learning behavior toward to their respective cluster center and the best cluster center.

From the comparative results in simulation, we can conclude that either the modified BSO or basic BSO or PSO can complete the optimization goal to form a comparative stable flight formation. It also demonstrates that the modified BSO has certain superiority in jumping out of local optimum and speeding up the optimization process. Our future work will focus on a novel bio-inspired algorithm named Pigeon-Inspired Optimization (PIO) [26] to provide new methods for multi-UAV control problem.

Acknowledgments This work was partially supported by the Natural Science Foundation of China (NSFC) under grant # 61333004, #61273054 and #61175109, National Key Basic Research Program of China(973 Project) under grant #2013CB 035503 and #2014CB046401, Top-Notch Young Talents Program of China, and Aeronautical Foundation of China under grant #20135851042.

References

1. Derafa, L., Benallegue, A., Fridman, L.: Super twisting control algorithm for the attitude tracking of a four rotors UAV. J. Frankl. Inst. **349**(2), 685–699 (2012)
2. Wang, Y.Q., Wu, Q.H., Wang, Y.: Distributed cooperative control for multiple quadrotor systems via dynamic surface control. Nonlinear Dyn. **75**(3), 513–527 (2014)
3. Maqsood, A., Go, T.H.: Multiple time scale analysis of aircraft longitudinal dynamics with aerodynamic vectoring. Nonlinear Dyn. **69**(3), 731–742 (2012)
4. Karimoddini, A., Hai, L., Chen, B.M., Lee, T.H.: A bumpless hybrid supervisory control algorithm for the formation of unmanned helicopters. Mechatronics **23**(6), 677–688 (2013)
5. Chang, B.L.: A dynamic virtual structure formation control for fixed-wing UAVs. In: IEEE International Conference on Control and Automation, pp. 627–632. Santiago (2011)
6. Giulietti, F., Pollini, L., Innocenti, M.: Autonomous formation flight. IEEE Control Syst. Mag. **20**(6), 34–44 (2000)
7. Binetti, P., Ariyur, K.B., Krstic, M., Bernelli, F.: Formation flight optimization using extremum seeking feedback. J. Guid. Control. Dyn. **26**(1), 132–142 (2003)
8. Peng, Z., Jikai, L.: On new UAV flight control system based on Kalman & PID. In: 2nd International Conference on Intelligent Control and Information Processing, pp. 819–823. Harbin (2011)
9. Xie, F., Zhang, X., Fierro, R., et al.: Autopilot-based nonlinear UAV formation controller with extremum-seeking. In: 44th IEEE Conference on Decision and Control, pp. 4933–4938. Sevilla (2005)
10. Paul, T., Krogstad, T.R., Gravdahl, J T.: UAV formation flight using 3D potential field. In: 16th Mediterranean Conference on Control and Automation, pp. 1240–1245. Ajaccio Corsica (2008)
11. Chang, B.L., Quee, S.N.: A flexible virtual structure formation keeping control for fixed-wing UAVs. In: Proceeding of IEEE International Conference on Control and Automation, pp. 621–626. Santiago (2011)
12. Zhang, X.Y., Duan, H.B., Yu, Y.X.: Receding horizon control for multi-UAVs close formation control based on differential evolution. Sci. China Inf. Sci. **53**(2), 223–235 (2010)
13. Hua, S.L., You, Y., Zhang, H., Song, H.: Receding horizon control of UAV formations. Electron. Optics Control **249**(1), 1–5 (2012)
14. Francesco, B., Tamás, K., Gary, J.B.: Collision-free UAV formation flight using decentralized optimization and invariant sets. In: Proceeding of IEEE International Conference on Decision and Control, vol. 1, pp. 1099–1104. Nassau (2004)
15. Duan, H.B., Yu, Y.X., Zhao, Z.Y.: Parameters identification of UCAV flight control system based on predator-prey particle swarm optimization. Sci. China Inf. Sci. **56**(1), 1–12 (2013)
16. Duan, H.B., Liu, S.Q.: Non-linear dual-mode receding horizon control for multiple unmanned air vehicles formation flight based on chaotic particle swarm optimization. IET Control Theory Appl. **4**(11), 2565–2578 (2010)
17. Duan, H.B., Li, P.: Bio-inspired computation in unmanned aerial vehicle. Springer, Berlin (2013)

18. Shi, Y.H.: Brain storm optimization algorithm. In: Proceeding of 2nd International Conference Swarm Intelligence, pp. 303–309. Chongqing (2011)
19. Zhan, Z.H., Zhang, J., Shi, Y.H., Liu, H.L., A modified brain storm optimization. In: Proceeding of 2012 IEEE World Congress Computational Intelligence, pp. 1–8. Brisbane (2012)
20. Wang, J., Xin, M.: Integrated optimal formation control of multiple unmanned aerial vehicles. IEEE Trans. Control Syst. Technol. **21**(5), 1731–1744 (2013)
21. Shi, Y.H.: An optimization algorithm based on brainstorming process. Int. J. Swarm Intell. Res. **2**(4), 35–62 (2011)
22. Duan, H.B., Li, S.T., Shi, Y.H.: Predator-prey brain storm optimization for DC brushless motor. IEEE Trans. Magn. **49**(10), 5336–5340 (2013)
23. Gao, J., Zhi, L., Wu, T., Wang, L.: Diversity of contribution promotes cooperation in public goods games. Phys. A Stat. Mech. Appl. **389**(16), 3166–3171 (2010)
24. Xu, C.F., Duan, H.B., Liu, F.: Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path. Aerosp. Sci. Technol. **14**(8), 535–541 (2010)
25. Yang, B., Li, W.Z., Yang, F.: A new PSO-PID tuning method for time-delay processes. In: Proceeding of 2nd International Symposium on Systems and Control in Aerospace and Astronautics, pp. 1–6. Shenzhen (2008)
26. Duan, H.B., Qiao, P.X.: Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. Int. J. Intell. Comput. Cybern. **7**(1), 24–37 (2014)