Artificial Bee Colony–Based Direct Collocation for Reentry Trajectory Optimization of Hypersonic Vehicle

HAIBIN DUAN, Senior Member, IEEE
SHUANGTIAN LI
Beihang University (formerly Beijing University of Aeronautics and Astronautics, BUAA)
Beijing, China

Reentry trajectory planning has become a key issue because of the rapid development of hypersonic vehicles. In this paper, an artificial bee colony (ABC)–based direct collocation method is proposed for reentry trajectory optimization. First, the reentry trajectory optimization problem is elaborated. The principles of the ABC algorithm are then introduced. Finally, the ABC-based direct collocation method is proposed. The control variables are discretized at a set of Legendre–Gauss collocation points and are optimized with the ABC approach. The parameters in ABC are assigned in terms of both the performance and the time cost of the algorithm. A penalty factor is adopted to modify the fitness value of some particular individuals in the colony so that every individual can be fully used. Series of comparative experimental results verify the feasibility and effectiveness of our proposed method. Further discussions and simulations are given for analyzing the impact of ABC parameters on the performance of the method.

Manuscript received October 17, 2012; revised August 11, 2013, May 14, 2014; released for publication August 12, 2014.

DOI. No. 10.1109/TAES.2014.120654.

Refereeing of this contribution was handled by I. Hwang.

Authors' addresses: H. Duan, State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (formerly Beijing University of Aeronautics and Astronautics, BUAA), No. 37, Xueyuan Road, Beijing 100191, China, E-mail: (hbduan@buaa.edu.cn); S. Li, Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University (formerly BUAA), Beijing 100191, China.

I. INTRODUCTION

With the rapid advance of hypersonic vehicles in the past decades [1], the modeling, guidance, and control of hypersonic reentry vehicles have been widely considered in aerospace engineering fields [2–5]. One kind of hypersonic vehicle with a lifting body exhibits larger lift-to-drag (L/D) ratios than do conventional hypersonic vehicles and has a larger lift for a given angle of attack (AOA) [6]. This kind of hypersonic vehicle, controlled by aerodynamic forces, can take a long-range gliding flight after a reentry maneuver depending on its large L/D ratio. Thus, it is provided with a prompt global reaching capability [7]. The trajectory design for hypersonic reentry vehicles has been extensively studied because of the specific mission requirements and complex constraints that the vehicle should obey during the reentry phase. Trajectory design is an optimal-control problem. Indirect and direct methods are the two main approaches to this problem. The indirect method is based on the Pontryagin maximum principle [8], whereas the direct methods are based on nonlinear programming theory, converting the trajectory optimization problem into a nonlinear programming (NLP) problem and then solving via numerical methods. Among the direct methods, collocation methods are widely used [9–11]. A comparative analysis between global and local interpolation of collocation methods is proposed in [12]. The waypoint and no-fly zone of the reentry vehicle are taken into account in [20]. A genetic algorithm (GA) was first used to generate candidate solutions for initializing the collocation method in [13]. Particle swarm optimization (PSO) was used to optimize unknown factors in a closed-form solution to a reentry problem in [14].

Trajectory optimization for hypersonic reentry vehicles is faced with the following difficulties. Because the Mach number and the flight height of the hypersonic vehicle vary greatly during the reentry phase, the aerodynamic feature of the vehicle has large uncertainty and nonlinearity. Moreover, the trajectory is subject to several constraints, including heating rate, dynamic pressure, overload, and terminal state, and there are a large number of design variables to consider. Optimal algorithms may converge to local optima or even fail to converge if the initial guess is not set appropriately [15]. Therefore, new approaches are required to cope with the trajectory optimization problem for hypersonic reentry vehicles.

The artificial bee colony (ABC) algorithm was originally presented by Dervis Karaboga in 2007 [16] and was inspired by the collective behavior of honey bees. The ABC algorithm has been proven to exhibit a better performance in the function optimization problem in comparison to the GA, the differential evolution algorithm and the PSO algorithm [17, 18]. Compared with other algorithms, the ABC algorithm presents a number of advantages [19]. First, for each iteration of the algorithm, both global search and local search are conducted; consequently, the probability of obtaining the optimum is

increased substantially. A second advantage is that a fitness function is formulated and chosen to be the evaluation standard, with no derivative or gradient information needed. Third, each searching operation of the ABC algorithm obeys probability distribution rather than in a certain way according to the gradient. Thus, the ABC algorithm is capable of coping with problems with large uncertainty and nonlinearity. Fourth, the ABC algorithm has a unique operator called a scout, which can randomly generate new candidate individuals when the swarm converges to local optima. Therefore, the ABC algorithm has better ability to jump out of suboptimal solutions and find the global best one. Because of these advantages, the ABC algorithm is adopted to solve the reentry trajectory optimization problem. In this work, a novel approach combined with the direct collocation method and the ABC algorithm is proposed. The control variables of the hypersonic reentry vehicle are discretized at a set of Legendre–Gauss collocation points and then optimized with the ABC algorithm. A penalty factor is adopted to modify the fitness value of some particular individuals in the colony and use each.

The paper is organized as follows. In Sec. II, the aerodynamic model of the hypersonic reentry vehicle is presented and some basic knowledge about the trajectory planning problem is provided. Sec. III describes the principle of the ABC method. In Sec. IV, the procedure of reentry trajectory planning using the ABC method is elaborated in detail. Then in Sec. V, a series of comparison experiments between the ABC method and the PSO are carried out, and further discussion about the parameters in the ABC algorithm is conducted. The paper is concluded in the last section.

II. REENTRY TRAJECTORY OPTIMIZATION PROBLEM

A. Bolza Optimal Problem

The reentry trajectory optimization problem can be expressed in a general Bolza form as described here [19, 20]. The dynamic equation can be denoted in general as follows:

$$\dot{x} = f(x(t), u(t), t), t \in \left[t_o, t_f\right] \tag{1}$$

where $x(t)$ is the state vector and $u(t)$ is the control vector, as well as the design variable.

The inequality constrains, including path constraints and terminal constraints, are denoted as

$$C(x(t), u(t), t) \leq 0 \tag{2}$$

The equality constraints, including the initial state and midway waypoints, can be expressed as follows:

$$E(x(t), u(t), t) = 0 \tag{3}$$

The cost at the terminal point can be represented as $\Phi(x_f, t_f)$. In this paper, the terminal error $\phi(x_f, t_f)$ is taken as $\Phi(x_f, t_f)$ and denoted as

$$\Phi(x_f, t_f) = \phi\left(x_f, t_f\right) \tag{4}$$

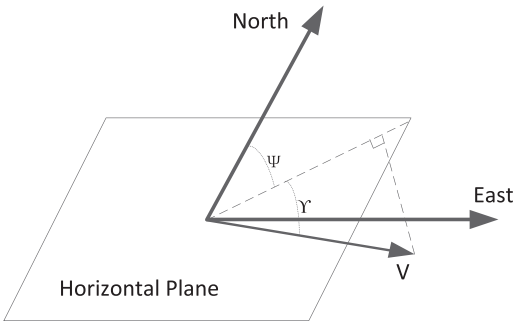

Fig. 1. 3 DOF motion parameters.

The integrand cost that exists throughout the reentry phase, taken from the initial time t_0 to final time t_f, is denoted as $g(x(t), u(t), t)$.

For the reentry task of the hypersonic vehicle studied in this work, the initial states are given; therefore, the cost function J is determined by the control variable $u(t)$. Pick $u(t)$ as the optimization variable, and then the Bolza problem is denoted in the following general form:

$$\begin{aligned} &\min J = \Phi(x_f, t_f) + \int_{t_0}^{t_f} g\,(x(t), u(t), t)\,dt \\ &s.t. \\ &\dot{x} = f(x(t), u(t), t), \quad t \in \left[t_o, t_f\right] \\ &C(x(t), u(t), t) \leq 0 \\ &E(x(t), u(t), t) = 0 \end{aligned} \tag{5}$$

B. 3 Degrees-of-Freedom (3-DOF) Reentry Motion Equations

In the reentry task, the AOA α and bank angle σ are the control variables required to be computed for steering the hypersonic reentry vehicle from a given state to a certain terminal state, satisfying all path constraints. In the reentry problem discussed here, Earth is assumed to be a rotating symmetric sphere. The (3-DOF) motion equations of the point mass model are shown in (6)–(11). In these equations, r is the radial distance between the vehicle and the Earth's core, θ is the vehicle's longitude, and φ is the latitude. In addition, V denotes the velocity vector, γ is the trajectory inclination angle, and ψ is the heading angle. As seen in Fig. 1, the trajectory inclination angle γ represents the angle between the velocity vector and the local horizontal plane, upward positive. Meanwhile, the heading angle ψ is the angle between the projection of the velocity vector in the local horizontal plane and the geographical true north, clockwise positive:

$$\frac{dr}{dt} = V \sin\gamma \tag{6}$$

$$\frac{d\theta}{dt} = \frac{V \cos\gamma \sin\psi}{r \cos\varphi} \tag{7}$$

$$\frac{d\varphi}{dt} = \frac{V \cos\gamma \cos\psi}{r} \tag{8}$$

$$\frac{dV}{dt} = \omega^2 r\cos\varphi(\sin\gamma\cos\varphi - \cos\gamma\sin\varphi\cos\psi) - \frac{D}{m} - g\sin\gamma \tag{9}$$

$$\frac{d\gamma}{dt} = \frac{1}{V}\left[\frac{L\cos\sigma}{m} + \left(\frac{V^2}{r} - g\right)\cos\gamma + 2\omega V\cos\varphi\sin\psi + \omega^2 r\cos\varphi(\cos\gamma\cos\varphi + \sin\gamma\sin\varphi\cos\psi)\right] \tag{10}$$

$$\frac{d\psi}{dt} = \frac{1}{V}\left[\frac{L\sin\sigma}{m\cos\gamma} - 2\omega V\,(\cos\varphi\tan\gamma\cos\psi - \sin\varphi) + \frac{V^2}{r}\cos\gamma\sin\psi\tan\varphi + \frac{\omega^2 r}{\cos\gamma}\sin\gamma\cos\varphi\sin\psi\right] \tag{11}$$

C. Aerodynamic Model

In (6)–(11), ω, g, D, L, σ are, respectively, Earth's rotation angular velocity, the gravitational acceleration, the drag, the lift, and the bank angle. The lift L and drag D are assumed to have standard forms as follows:

$$L = qSC_L, \quad D = qSC_D \tag{12}$$

where q is the dynamic pressure. C_L and C_D are both functions of AOA α and Mach number M.

D. Constraints of Reentry

1) *Path Constraints*: During the reentry phase, to ensure the structural safety of the hypersonic vehicle and to satisfy the demands of thermal protection, the constraints of heating rate, dynamic pressure, and overload must be strictly satisfied, and the reentry trajectory should be as smooth as possible.

a) *Heating rate*: Because the stagnation point is the most seriously heated region, its heating rate is usually chosen as one of the reentry path constraints. An approximate expression of the heating rate of the stagnation point is established in [21]:

$$\dot{Q} = K\left(\frac{\rho}{\rho_0}\right)^n\left(\frac{V}{V_C}\right)^m < \dot{Q}_{\max} \tag{13}$$

For hypersonic vehicles, it is common to take $n = 0.5$, and $m = 3.15$ [22], where K is a constant related to the vehicle's nose radius.

b) *Dynamic pressure*: Dynamic pressure is an important feature variable in aerodynamics. In terms of its impact to the control system and the demand of lateral stability, the dynamic pressure during the reentry phase strictly satisfies the following constraint:

$$q = \frac{\rho V^2}{2} < q_{\max} \tag{14}$$

c) *Overload*: The overload of the hypersonic vehicle should be taken into account in terms of structural safety during the reentry phase. As for the hypersonic vehicle discussed here with a lifting body, both axial and normal overload would increase to a greater extent, threatening the structural safety. Therefore, the total overload is taken as the overload constraint in this paper:

$$n = q\sqrt{C_D^2 + C_L^2}\frac{S_{ref}}{mg} < n_{\max} \tag{15}$$

2) *Terminal Constraint*: To ensure terminal guidance, the terminal state after the reentry phase must satisfy the terminal constraint. The error of longitude, latitude, and radius magnitude should be less than a certain amount:

$$\begin{aligned} e_\theta &= \theta(t_f) - \theta_f \le e_{\theta\max} \\ e_\varphi &= \varphi(t_f) - \varphi_f \le e_{\varphi\max} \\ e_r &= r(t_f)r_f \le e_{r\max} \end{aligned} \tag{16}$$

3) *Constraints of Control Variables*: In the 3 DOF reentry model shown in (6)–(11), the bank angle σ is the explicit control variable, while AOA α is implicit. Because of the physical characteristics of the pneumatic rudder, the angular rate and range of the control variables need to meet certain requirements, given as follows:

$$\begin{cases} |\alpha| \le \alpha_{\max} \\ |\dot{\alpha}| \le \dot{\alpha}_{\max} \end{cases}, \begin{cases} |\sigma| \le \sigma_{\max} \\ |\dot{\sigma}| \le \dot{\sigma}_{\max} \end{cases} \tag{17}$$

E. Cost Function

1) *Total Heat Load During the Reentry Phase*: In reentry trajectory optimization problems with a given target point, an important goal of the optimization is to reduce the aerodynamic heating, thus reducing the burden and the weight of the thermal protection system. Therefore, the integral of the heating rate of the stagnation point is included as part of the cost function, denoted as follows:

$$J_Q = \int_{t_0}^{t_f} \dot{Q}dt \tag{18}$$

2) *Smoothness of the Reentry Trajectory*: To ensure smoothness of the trajectory, a smoothness indicator is introduced. We integrate the square of the changing rate of the trajectory inclining angle and take it as part of the cost function:

$$J_I = \int_{t_0}^{t_f} \dot{\gamma}^2 dt \tag{19}$$

3) *Terminal Error*: For direct collocating methods, because the nodes and terminal points are set in advance, the terminal state of the trajectory it generates is inevitably equal to the target state. In other methods that do not collocate the terminal state, the terminal error is taken as an optimal target, denoted as follows:

$$J_D = \sqrt{\left[\theta_f - \theta(t_f)\right]^2 + \left[\varphi_f - \varphi(t_f)\right]^2} \tag{20}$$

4) *Overall Cost Function*: Taking (18)–(20) into account, the complete cost function is

$$J = J_D + w_1 J_I + w_2 J_Q \tag{21}$$

where w_1 and w_1 are weight coefficients. Accordingly,

$$f = \frac{1}{J+1} \tag{22}$$

is taken as the fitness value in the ABC algorithm, whose theoretic maximum is 1.

III. PRINCIPLES OF THE ABC ALGORITHM

German biologist Karl von Frisch, winner of 1973's Nobel Prize in Psychology or Medicine, found that although each bee can only complete a single task, the entire colony can always find good nectar with the information exchange among the colony via waggle dance and specific odor. Inspired by such self-organized behavior, Karaboga presented the ABC algorithm in 2007 [16].

A. Basic Concepts

Three concepts are defined as follows to elaborate the theory of the ABC algorithm [18].

1) *Food Sources*: Food sources represent possible solutions within certain searching space and are evaluated by a numerical variable called fitness value. In reentry trajectory optimization, a food source represents a solution vector consisting of a series of control variables at discrete nodes, while the fitness value is given in (22), negatively correlated with the cost function J.

2) *Employed Foragers*: The employed foragers are one on one corresponding to the food sources they exploit. Their knowledge of the food sources is shared and evaluated among the colony. Based on the evaluation of the food sources of the employed bees, these bees could perform one of three behaviors as follows [18, 23]:

1) Abandon the current food source and become an unemployed forager if the food source is exhausted and better choices are available; in other words, the solution this bee represents is not good enough for the optimization problem (UF in Fig. 2 [18]).

2) If the current food source is overwhelmingly superior to other food sources, the employed forager would conduct recruitment so that more unemployed foragers could exploit such food source (EF1 in Fig. 2). This meant that the algorithm searches for new solutions in the neighborhood of this solution.

3) It continues to forage without recruitment to the food source (EF2 in Fig. 2).

3) *Unemployed Foragers*: Unemployed foragers signify the rest individuals in the colony without affiliation with the food sources, and they are divided into two categories: scouts and onlookers.

a) *Onlookers (R in Fig. 2)*: The onlookers evaluate the information shared by the employed foragers and search appropriate food sources accordingly. The probability of searching near a certain food source is proportional to its fitness value.

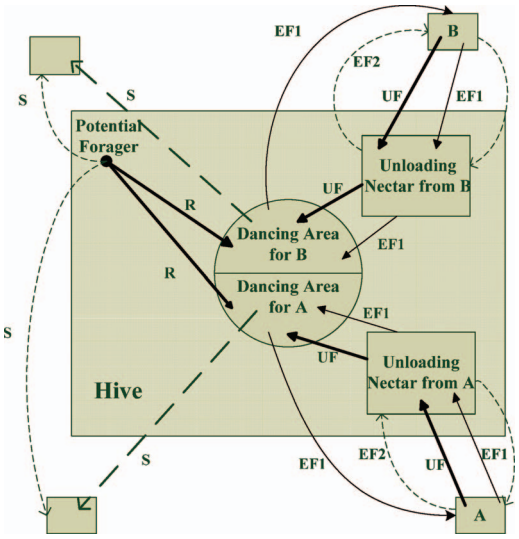

Fig. 2. Behavior of bee foraging for nectar.

b) *Scouts (S in Fig. 2)*: Scouts randomly search for new food sources with no prior knowledge. A scout's essence is partly reinitializing the colony if the algorithm fails to find better solutions for too many generations of iteration. Such a technique assures the algorithm's ability to escape from local optima.

B. General Procedure

The general procedure of the ABC algorithm is described as follows. At the beginning, the bee colony is initialized with the scale of the colony N_c and the number of iteration times *Iter* assigned. Each bee is a vector consisting of D elements, where D denotes the degree of the solution space. The jth element of the ith bee is denoted as X_{ij}. Half of the N_c bees are employed foragers corresponding to several food sources, whose initial value is randomly generated among the solution space, whereas the other bees are unemployed foragers.

In each iteration of the algorithm, the following operations are conducted:

- The employed foragers search for new food sources near the current ones and evaluate the new ones. Denoting the new position as V, the searching principle is defined as

$$V_{ij} = X_{ij} + \varphi_{ij}(X_{ij} - X_{kj}) \tag{23}$$

where $j \in \{1, 2, \ldots, D\}$ and stands for the jth element. In addition, $i, k \in \{1, 2, \ldots, N_c/2\}$, $k \neq i$, and they stand for the index of the bees. Both k and j are randomly generated. The transformation factor φ_{ij} is a random number between $[-1, 1]$. Such a searching principle adaptively reduces the searching step when the search approaches the optimal solution. If the fitness value of the new food source is better than that of the old one, the employed forager would memorize the new one and forget the previous one.

- After all employed foragers finished searching, the food source information is shared among the entire colony. Accordingly, each onlooker would choose an employed

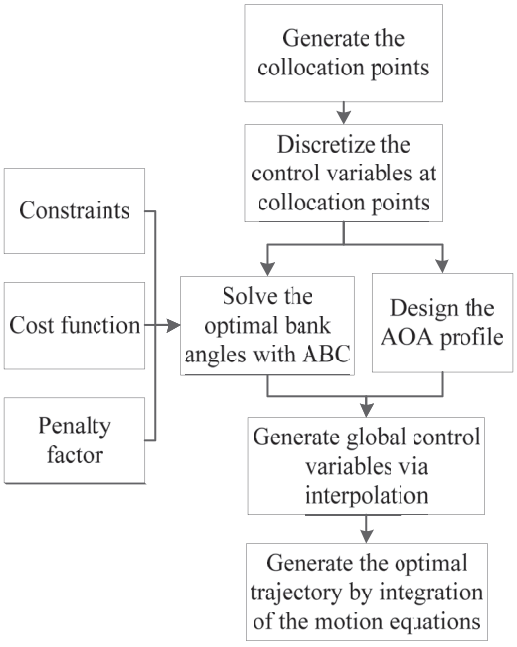

Fig. 3. Schematic diagram of algorithm.

forager to follow. The choosing principle can be defined as

$$P_i = \frac{f(X_i)}{\sum_{i=1}^{N_c/2} f(X_i)} \tag{24}$$

where P_i and $f(X_i)$ are the possibility of an onlooker to follow the ith-employed bee and the fitness value of the ith-employed bee, respectively. After that, each onlooker searches in the neighborhood of the selected food source and would take the place of the current employed forager if it explores a better solution.

- If the searching times *trial* at one food source exceeds a certain threshold *limit*, the scout mechanism is activated; namely, the current food source is abandoned and a new randomly generated food source would take its place.

If the number of iteration times exceeds *Iter*, the optimal solution would be output; otherwise, a new iteration would begin.

IV. THE ABC ALGORITHM–BASED METHOD

In the ABC-based direct collocation method, the control variables are discretized at a set of Legendre–Gauss collocation points. In other words, only the control variables at these points are involved. The control inputs at other time points are calculated via interpolation. Thus, the problem is converted to a finite-dimensional NLP problem. Then the problem can be solved with the ABC algorithm to minimize the value of the cost function. The procedure of trajectory optimization is shown in Fig. 3.

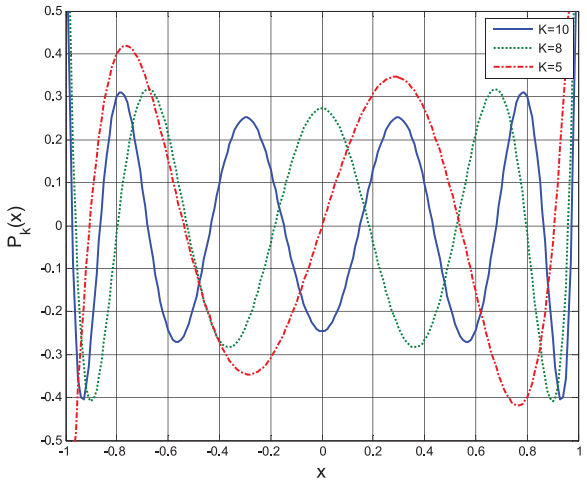

Fig. 4. Legendre polynomials.

A. Distribution of Collocation Points

In the Gauss pseudospectral method [24, 25], the collocation points are distributed according to the root of the K-order Legendre polynomial $P_K(\bar{\tau})$, where

$$P_K(\bar{\tau}) = \frac{1}{2^K K!}\frac{d^K}{d\bar{\tau}^K}\left[(\bar{\tau}^2-1)^K\right],\ \mathrm{K} = 0, 1, 2\cdots \tag{25}$$

As presented in Fig. 4, such distributed collocation points are featured by their density at both ends and scarcity in the middle, meeting the demand of accuracy near the terminal point. Therefore, in our proposed method, the Legendre–Gauss collocation point is where the control variables are discretized and optimized.

For the collocation, the transformation between time domain $t \in [t_o, t_f]$ and $\bar{\tau} \in [-1, 1]$ is necessary, which is given by

$$t(\bar{\tau}) = \frac{t_f - t_0}{2}\bar{\tau} + \frac{t_f + t_0}{2} \tag{26}$$

After the previously described transformation, solve $P_K(\bar{\tau})$ shown in (25). The K roots, together with the starting point $\bar{\tau}_0 = -1$, are chosen as the collocation points. Denote the transformed control variables as $U(\bar{\tau})$, $\bar{\tau} \in [-1, 1]$. Then the control variables at the collocation points can be expressed as $U(\bar{\tau}_i)$, $(i = 0, \ldots, K)$. The global control variables are approximated using $(K+1)$ Lagrange interpolation polynomials $L_i(\bar{\tau})$, namely,

$$u(t(\bar{\tau})) = U(\bar{\tau}) = \sum_{i=1}^{K} L_i(\bar{\tau})U(\bar{\tau}_i) \tag{27}$$

where $\bar{\tau}_i (i = 1, \ldots, K)$ stands for the Gauss collocation points, and

$$L_i(\bar{\tau}) = \prod_{j\neq 0, j=i}^{K} \frac{\bar{\tau} - \bar{\tau}_j}{\bar{\tau}_i - \bar{\tau}_j} \tag{28}$$

B. Operations of Control Variables

1) *Operation of AOA*: As presented in [26], when the AOA is assigned the maximum value, the flight range will

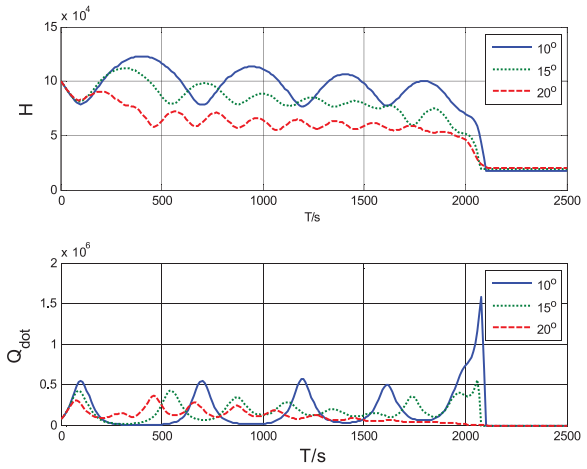

Fig. 5. Heating rate and altitude for different initial AOA values.

reach its peak yet the trajectory will tend to oscillation. However, in terms of structural stability, the trajectory should be relatively smooth. Moreover, because of the burden of the thermal protection system, the total flight time should be reduced at the expense of the flight range.

However, simulations in Fig. 5 illustrate that the peak of the heating rate goes with the minimal value of the altitude and that the larger the initial AOA, the lower the peak heating rate.

In summary, we assign the initial AOA the maximum value. Then after the flight speed threshold, the AOA is set to the value with the maximum L/D ratio.

The L/D feature of the hypersonic vehicle is presented in Fig. 6. When the Mach number is greater than 6, the AOA value with the maximum L/D rate is approximately 4. While it increases up to 12, the Mach number decreases.

2) *Operation of Bank Angle*: As for the bank angle σ, according to the procedure described in (26)–(28), randomly choose the initial iteration values at all collocation points satisfying $|\sigma| \leq \sigma_{\max}$. Then generate the global bank angle values by the Lagrange interpolation. Integrate the 3 DOF motion equation presented in (6)–(11) and get the history of the state variables and constraint indicators, with α and the generated σ as the control variables. Thus, the trajectory optimization problem is converted into an NLP problem described as follows: Find σ_k $(k = 1, \ldots, K)$ to minimize the cost functions in (21), subject to the path constrains in (13)–(15).

C. Optimization by the ABC Algorithm

1) *Choice of Parameters*: Because of the mechanism of the ABC algorithm, each bee in each generation of iteration demands calculation of its fitness value. As for the reentry trajectory optimization problem, each calculation of the fitness value calls for numerical integration of the motion equation throughout the reentry phase, which is time consuming. Denote the time for each calculation as t_{once}. The total calculating time is defined as

$$t_{total} = t_{once} N_c Iter \tag{29}$$

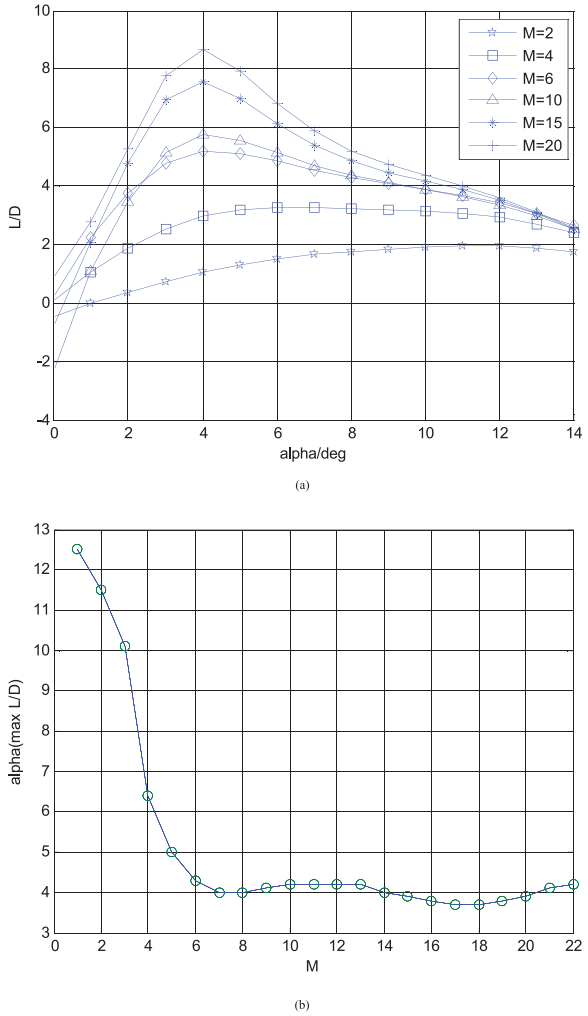

Fig. 6. L/D feature of hypersonic vehicle. (a) L/D rate at different Mach numbers. (b) AOA value with maximum L/D rate at different Mach numbers.

where N_c is the scale of the colony and *Iter* is the number of iteration times. Larger N_c and *Iter* values would undoubtedly result in longer computing time. However, as a population-based algorithm, the ABC algorithm would exhibit better performance with larger N_c and *Iter* values, which is further demonstrated in Sec. V. To balance the searching ability and computation time, N_c and *Iter* are chosen as $N_c = 20$ and $Iter = 15$.

2) *Penalty Factor*: In the iteration process of the colony, an individual may arise that does not strictly satisfy all path constraints, yet its terminal error could be rather small. As discussed earlier, the decrease in the quantity of searching bees would result in poor searching ability of the entire colony. The severity of this problem would increase if all individuals who fail to strictly satisfy all path constraints are directly abandoned without exploitation. To make full advantage of these individuals in the colony, the penalty factor is introduced to evaluate each, taking the aforementioned situation into account. The fitness value of each individual is multiplied by a penalty factor so that the individuals who have rather a

high fitness value but fail to satisfy the path constraints are still preserved in the colony. Because of the penalty factor, they are not the optimal value, yet the colony can search for potentially satisfactory solutions near it.

As for every individual denoted $X_i = (X_i^1, \ldots, X_i^{K+1})$, its fitness value with the penalty factor is defined as

$$F_i = \begin{cases} F_i, f_c \leq f_{c\,\max} \\ F_i \bar{p}, f_c > f_{c\,\max} \end{cases} \tag{30}$$

where f is constraints related the performance indicator and $\bar{p}$ is the penalty factor. Considering the structural safety, the penalty factor should be relatively small. In our simulation, it is chosen as 0.01.

3) *Detailed Process*: The detailed procedure of control variable optimization using the ABC algorithm is described as follows:

1) **Initialization**: The scale of the colony (N_c), the number of iteration times (*Iter*), the maximum number of local search (*limit*), and the number of searching times near one food source (*trial*) are assigned. In addition, a D-dimensional search space is defined, where D denotes the number of design variables, namely, the number of collocation points. The size of each dimension is determined by the range of the bank angle, as shown in (12). The search space can be expressed as

$$S = \bar{\sigma}^{\,D}, \bar{\sigma} \in [-\sigma_{\max}, \sigma_{\max}] \tag{31}$$

such that each point in S signifies a foraging bee and thus represents a set of bank angles at the aforementioned collocation points.

2) **Generation of initial employed foragers**: Half of the colony is defined as employed foragers, with their initial value randomly generated in S. Then using the subalgorithm for calculating the fitness value for each, the bank angle it represents is substituted into the motion equations in (6)–(11) so that the corresponding motion history is obtained. The cost function and fitness value can be calculated from (18)–(22). Then the value of f is modified depending on whether the path constraints in (8)–(10) are violated, as described in (30). The food source with maximum fitness value is noted as *GlobalBest*.

3) **Exploitation of employed foragers**: Each of the employed foragers conducts exploitation at its current food source. In other words, a local search near the current position is carried out, which can be express as in (23). Moreover, the fitness values of new positions are calculated with the subalgorithm mentioned earlier. Then a greedy selection is employed so that the old position can be replaced when a better position is obtained. During the process of greedy selection of the ith forager, if a better position is found, *trial*(i) is reset to 0. Otherwise, *trial*(i) is added by 1.

4) **Exploration of food sources**: Each of the onlookers may choose one employed forager to follow according to (24). Then similar operations to those of Step 3 are conducted, including local search, calculation of fitness value, and greedy selection.

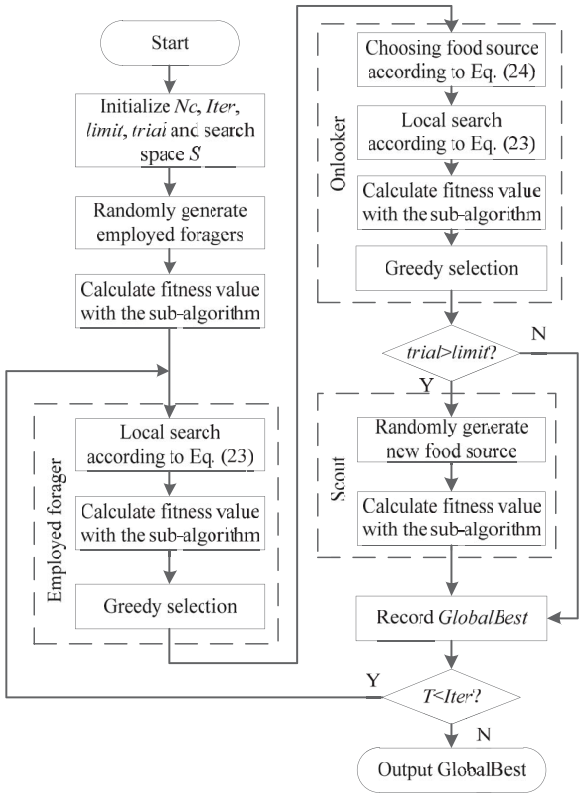

Fig. 7. Flow chart of ABC optimization.

5) **Global search of scouts**: When any *trial*(i) value exceeds the threshold *limit*, the corresponding food source is assumed to be exhausted and is replaced with a randomly generated position, which is called a scout.

6) **Updating *GlobalBest***: *GlobalBest* is updated if its fitness value is less than that of the best food source in this iteration.

7) **End of one iteration**: If the number of iteration times T is less than *Iter*, go back to Step 3. Otherwise, the algorithm is terminated and *GlobalBest* is output as the solution to the optimization problem.

The flow chart of the optimization is presented in Fig. 7, and the subalgorithm is illustrated in Fig. 8.

V. NUMERICAL SIMULATIONS

To investigate the feasibility of the proposed method on the trajectory optimization problem, a series of simulation experiments are conducted. Comparative study with another classical swarm intelligence algorithm, PSO, is presented. Moreover, further experiments are carried out to analyze the impact that the parameters of the ABC algorithm have on the performance of the algorithm.

The terminal condition and path constraints are given in Table I. The initial states of the hypersonic vehicle are described in Table II.

With V_C and ρ_0 known, (8) can be simplified as

$$\dot{Q} = K' \sqrt{\rho} V^{3.15} \tag{32}$$

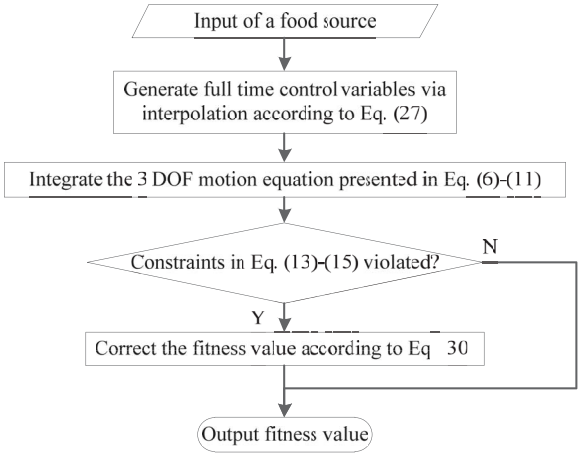

Fig. 8. Subalgorithm calculating fitness value.

TABLE I
Trajectory Constraints

Terminal constraints	H (km)	θ (deg)	φ (deg)		
	20 ± 1	210 ± 0.5	20 ± 0.5		
Path constraints	$\dot{Q}$ (kW/m^2)	q (kPa)	p	γ (deg)	$\dot{\gamma}$ (deg/s)
	≤ 1000	≤ 100	≤ 6	≤ 80	≤ 8

TABLE II
Initial States

No.	H (km)	V (m/s)	γ (deg)	ψ (deg)	φ (deg)	θ (deg)
1	100	7000	-2	68	0	160
2				79	10	
3				111	30	

where $K' = 0.0001$ [27]. The number of collocation points is chosen as 11.

In the simulations, the hypersonic vehicle is set to an initial state as one of the three conditions listed in Table II. Then a constrained (as listed in Table I) optimization problem that minimizes the cost function in (21) is solved with the proposed method. Furthermore, the reentry trajectory is calculated with the optimized control variables and the initial states.

A. Comparison With PSO

A comparative study with the PSO-based method is conducted to investigate the feasibility of the ABC-based approach. Except for the optimization algorithm, other tactics including direct collocation, operations of AOA, and use of penalty factor are the same. The control parameters of the ABC and PSO methods are given in Table III. With these parameters, a series of simulations is conducted in each of the three initial conditions. Figs. 9 and 10 illustrate the best result of both the ABC and the PSO methods under each condition, while the value of constraints, fitness value, and terminal conditions are given in Table IV in detail. Fig. 11 and Table V show the trajectories and bank angle of the three conditions.

TABLE III
Parameters of ABC and PSO

ABC		PSO	
N_c	20	N_c	20
Iter	15	*Iter*	15
limit	10	*Wmax*	0.9
		Wmin	0.1
		C1	2
		C2	2

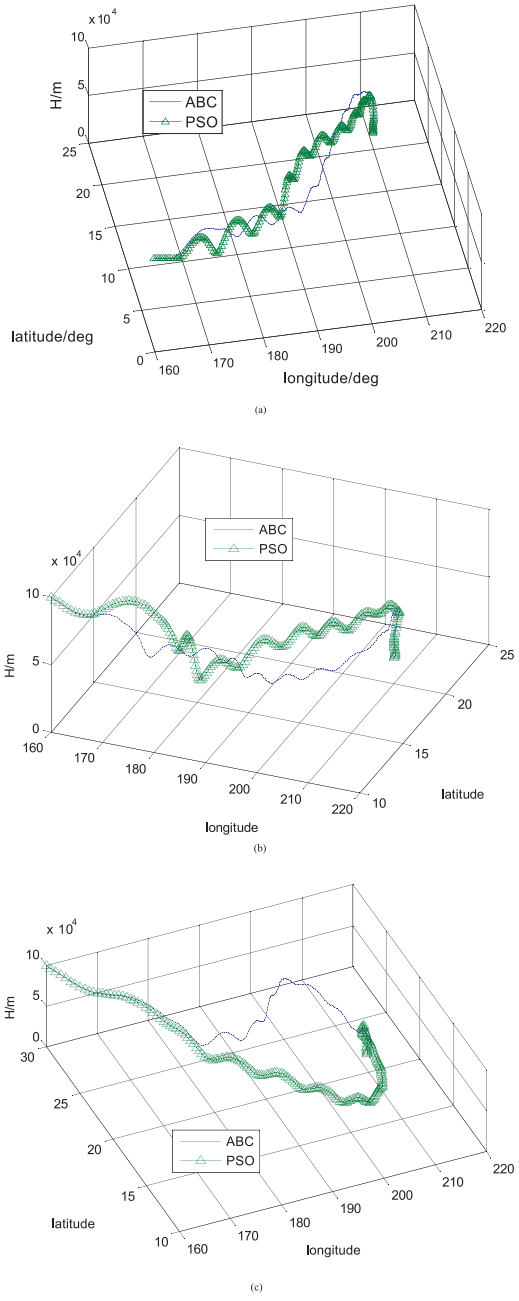

Fig. 9. Three-dimensional view of trajectories. (a) Condition 1. (b) Condition 2. (c) Condition 3.

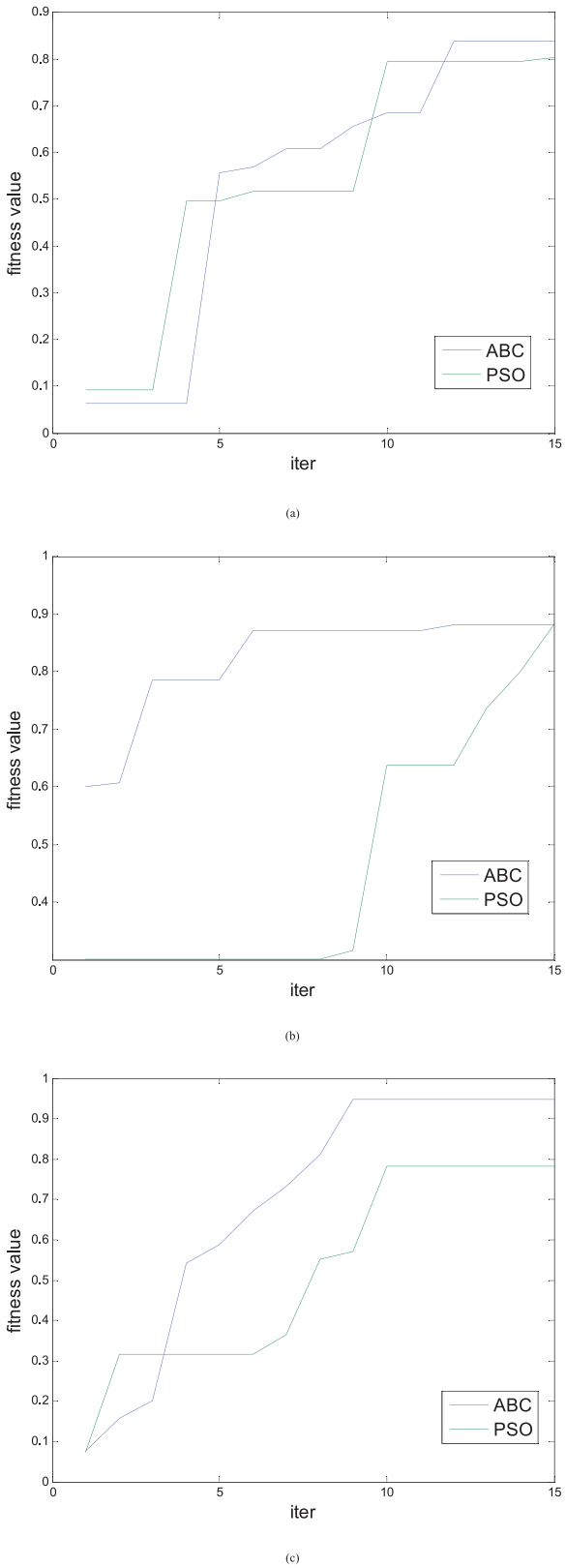

Fig. 10. Fitness values of ABC and PSO methods. (a) Condition 1. (b) Condition 2. (c) Condition 3.

According to the data in Tables IV and V, the trajectories obtained by both ABC- and PSO-based methods satisfy the path constraints and control variable

TABLE IV
Results of Simulations in Detail

No.		1	2	3	Average
f	PSO	0.8	0.88	0.78	0.82
	ABC	0.92	0.88	0.95	0.92
	Improved	15%	0%	22%	12%
$Qmax$ (kW/m^2)	PSO	434	373	343	383
	ABC	380	400	401	394
$Qmax$ (kPa)	PSO	3.88	3.87	2.51	3.42
	ABC	3.85	3.34	3.34	3.51
$nmax$	PSO	4.3	3.4	4.7	4.13
	ABC	4.3	2.7	2.7	3.23
$\lvert e_\theta \rvert$	PSO	0.152	0.013	0.003	0.056
	ABC	0.072	0.017	0.049	0.046
$\lvert e_\varphi \rvert$	PSO	0.198	0.058	0.278	0.176
	ABC	0.046	0.038	0.025	0.037
$\lvert e_r \rvert$	PSO	467	290	856	538
	ABC	263	619	504	462

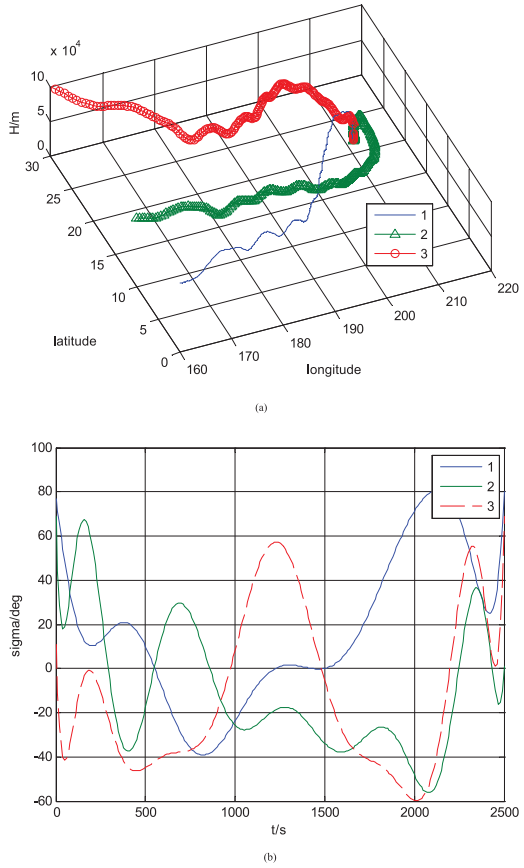

Fig. 11. Three conditions. (a) Trajectories. (b) Bank angle.

TABLE V
Data of Bank Angle

	1	2	3	Constraint
$\gamma_{\max}$ (deg)	79.8	68.8	68.7	80
$\gamma_{\min}$ (deg)	−39.2	−56.2	−59.8	−80
$\dot{\gamma}_{\max}$ (deg/s)	1.55	1.11	2.91	8
$\dot{\gamma}_{\min}$ (deg/s)	−0.57	−2.45	−2.20	−8

constraints, in comparison with Table I. In other words, the structural and thermal safety of the reentry vehicle is guaranteed, which is the prerequisite for the validity of an approach to trajectory optimization.

Furthermore, as shown in Fig. 9, both methods managed to generate reentry trajectories between the terminal position and each of the three initial positions, yet Fig. 10 and the data in Table IV reveal that the ABC-based approach exhibits a quicker convergence speed and better convergence ability than the method based on PSO. Specifically, as expressed in (22), the fitness value f is negatively correlated to the cost function J. In addition, the theoretical maximum value of f is 1. In each of the three conditions, the fitness function obtained by the ABC-based method finally converges to a particular value approximating 1, which is greater than that obtained by the PSO-based method.

In summary, the data and figures presented earlier confirm the feasibility of the proposed ABC-based method in that the reentry vehicle managed to reach the target point with all constraints satisfied and a fitness value close to the theoretical value. In addition, the ABC method has better converging performance than the PSO method does in the current issue.

B. Analysis of the Parameters

To further analyze the performance of the proposed ABC-based method, more discussion and experiments studying the impact of the parameters of the algorithm are conducted.

1) *Number of Iteration Times* Iter: As a population-based and evolutionary algorithm, the ABC algorithm is characterized by its evolution toward better solutions, assured by the mechanism of information sharing and greedy selection. Therefore, the more iterations generated, the more likely the colony would converge to the global optimum. However, the computing time is proportional to *Iter*, as discussed in (29). Thus, the choice of *Iter* is worth considering. It is shown in Fig. 10 that the colony can converges to a fairly preferable value within less than 15 iterations. Thus, *Iter* is chosen as 15 in this paper, and this choice is proven effective by the series of simulations mentioned earlier.

2) *Scale of the Colony* N_c: Generally, the larger scale of the swarm leads to more searches per generation of iteration and a higher level of diversity, which may benefit the algorithm. However, according to (29), computing time increases with N_c. A set of simulation is carried out with different N_c values, where $Iter = 15$ and $limit = 10$. The trajectories generated are presented in Fig. 12. The fitness values of the three conditions shown in Fig. 13 reveal that the larger N_c results in a greater convergent fitness value.

3) *Maximum Number of Local Search* limit: In small bee colonies, the diversity is mainly provided by the mechanism of the scout bees [28]. Such a function is controlled by the maximum number of local search (*limit*). A small *limit* may result in failure to fully exploit a food source, because food sources are more frequently abandoned before it provides useful information. However, if *limit* is too large, the colony may fail to jump out of local optima and lose its diversity. Figs. 14 and 15 are results of experiments conducted with same *Iter* and N_c but different *limit* values. It is shown in Fig. 15 that colonies with *limit* $= 3$ or *limit* $= 5$ fall into local optima, while the other with *limit* $= 10$ exhibits preferable performance.

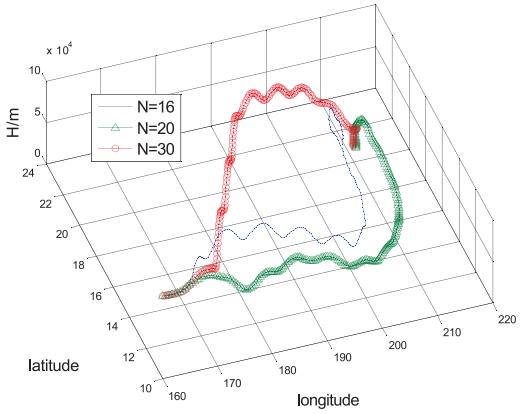

Fig. 12. Trajectories by different N_c.

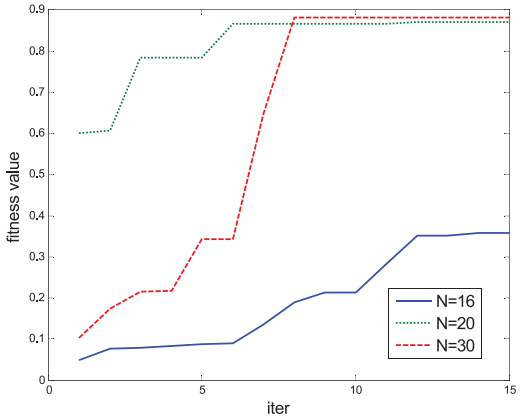

Fig. 13. Fitness values of different N_c.

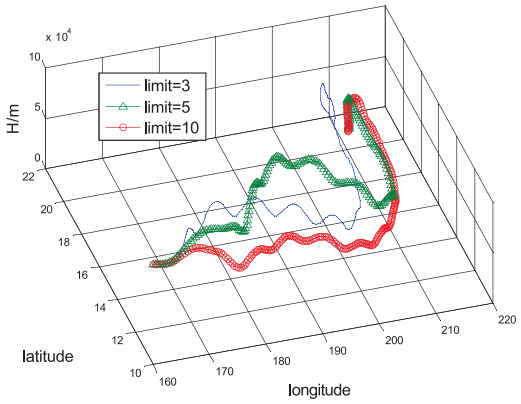

Fig. 14. Trajectories by different *limit*.

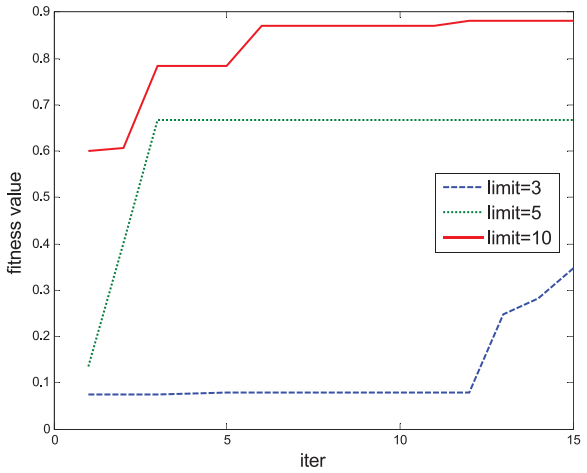

Fig. 15. Fitness values of different *limit*.

VI. CONCLUSION

The paper presents an approach to reentry trajectory planning combining the ABC algorithm and the direct collocation method. The control variables are discretized at collocation points, thus allowing the formulation of an optimization problem. Then the ABC algorithm is used to solve the problem. In terms of the computing time, the artificial colony is limited to a small scale. Under such circumstances, the mechanism of the penalty factor is introduced to modify the fitness value of the foragers in the colony so that every individual in the colony is fully used. Simulations have proved the feasibility, as well as the superiority, of the proposed method. Further discussion and experiments are conducted to investigate the impact of the parameters on the ABC algorithm's performance in trajectory optimization problems. Our future work will focus on improving the real-time performance and robustness of the ABC algorithm [29] and thus enhance the ABC algorithm's competence in dealing with more complex reentry tasks. We will also try to apply a novel bio-inspired algorithm named pigeon-inspired optimization [30] to solve this complicated problem.

ACKNOWLEDGMENTS

The authors are grateful to Prof. Robert C. Michelson for detailed guidance and valuable suggestions in this work. This work was partially supported by Natural Science Foundation of China under Grants No. 61425008, 61333004, No. 61273054, and No. 60975072; the National Key Basic Research Program of China (973 Project) under Grant No. 2014CB046401; the Top-Notch Young Talents Program of China, Aeronautical Foundation of China, under Grant No. 20135851042; and the Open Fund of the State Key Laboratory of Virtual Reality Technology and Systems under Grant No. VR-2013-ZZ-02.

REFERENCES

[1] Bertin, J., and Cummings, R.
Fifty years of hypersonics: Where we've been, where we're going.
Progress in Aerospace Sciences, **39** (2003), 511–536.

[2] Austin, J., and Leondes, C.
Statistically linearized estimation of reentry trajectories.
IEEE Transactions on Aerospace and Electronic Systems, **17**, 1 (1981), 54–61.

[3] Shen, Q., Jiang, B., and Cocquempot, V.
Fault-tolerant control for *T-S* fuzzy systems with application to near-space hypersonic vehicle with actuator faults.
IEEE Transactions on Fuzzy Systems, **20**, 4 (2012), 652–665.

[4] Duan, H. B., and Li, P.
Progress in control approaches for hypersonic vehicle.
Science China Technological Sciences, **55**, 10 (2012), 2965–2970.

[5] Yang, J., Li, S. H., Sun, C. Y., and Guo, L.
Nonlinear-disturbance-observer-based robust flight control for airbreathing hypersonic vehicles.
IEEE Transactions on Aerospace and Electronic Systems, **49**, 1 (2013), 1263–1275.

[6] Rodriguez, A. A., Dickeson, J. J., Cifdaloz, O., McCullen, R., Benavides, J., Sridharan, S., Kelkar, A., Vogel, J. M., and Soloway, D.
Modeling and control of scramjet-powered hypersonic vehicles: Challenges, trends, & tradeoffs.
In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, HI, Aug. 18–21, 2008.

[7] Wolf, A. F.
Conventional Warheads for Long-Range Ballistic Missiles: Background and Issues for Congress.
Washington, DC: Library of Congress, Congressional Research Service, 2009.

[8] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F.
The Mathematical Theory of Optimal Processes. New York: Interscience, 1962.

[9] Ross, I. M., and Fahroo, F.
Legendre pseudospectral approximations of optimal control problems.
Lecture Notes in Control and Information Sciences, Berlin, Germany: Springer-Verlag, 2003, No. 295, pp. 327–342.

[10] Williams, P.
Hermite–Legendre–Gauss–Lobatto direct transcription in trajectory optimization.
Journal of Guidance, Control, and Dynamics, **32**, 4 (2009), 1392–1395.

[11] Louembet, C., Cazaurang, F., Zolghadri, A., Charbonnel, C., and Pittet, C.
Path planning for satellite slew manoeuvres: A combined flatness and collocation-based approach.
IET Control Theory & Applications, **3**, 4 (2009), 481–491.

[12] Huntington, G. T., and Rao, A. V.
Comparison of global and local collocation methods for optimal control.
Journal of Guidance, Control, and Dynamics, **31**, 2 (2008), 432–436.

[13] Subbarao, K., and Shippey, B. M.
Hybrid genetic algorithm collocation method for trajectory optimization.
Journal of Guidance, Control, and Dynamics, **32**, 4 (2009), 1396–1403.

[14] Rahimi, A., Dev Kumar, K., and Alighanbari, H.
Particle swarm optimization applied to spacecraft reentry trajectory.
Journal of Guidance, Control, and Dynamics, **36**, 1 (2013), 307–310.

[15] Yong, E., Chen, L., and Tang, G.
A survey of numerical methods for trajectory optimization of spacecraft.
Journal of Astronautics, **29**, 2 (2008), 397–408.

[16] Karaboga, D., and Basturk, B.
A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm.
Journal of Global Optimization, **39**, 3 (2007), 459–471.

[17] Karaboga, D., and Basturk, B.
On the performance of artificial bee colony (ABC) algorithm.
Applied Soft Computing, **8**, 1 (2008), 687–697.

[18] Karaboga, D., and Akay, B.
A comparative study of artificial bee colony algorithm.
Applied Mathematics and Computation, **214**, 1 (2009), 108–132.

[19] Duan, H. B., Zhang, X. Y., and Xu, C. F.
Bio-Inspired Computing. Beijing, China: Science Press of China, 2011, pp. 90–92.

[20] Jorris, T., and Cobb, R.
Three-dimensional trajectory optimization satisfying waypoint and no-fly zone constraints.
Journal of Guidance, Control, and Dynamics, **32**, 2 (2009), 551–572.

[21] Hull, D. G., and Speyer, J. L.
Optimal reentry and plane-change trajectories.
Journal of the Astronautical Sciences, **2** (1982), 117–130.

[22] Yong, E., Tang, G., and Chen, L.
Rapid trajectory optimization for hypersonic reentry vehicle via Gauss pseudospectral method.
Journal of Astronautics, **29**, 6 (2008), 1766–1772.

[23] Xu, C. F., Duan, H. B., and Liu, F.
Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning.
Aerospace Science and Technology, **14**, 8 (2010), 535–541.

[24] Todd, H. G.
Advancement and analysis of a Gauss pseudospectral transcription for optimal control problems.
Cambridge, MA:
Massachusetts Institute of Technology, 2007.

[25] Benson, A., Thorvaldsen, T., and Rao, V.
Direct trajectory optimization and costate estimation via an orthogonal collocation method.
Journal of Guidance, Control and Dynamics, **29**, 6 (2006), 1435–1440.

[26] Yong, E.
Study on Trajectory Optimization and Guidance Approach for Hypersonic Glide-Reentry Vehicle.
Changsha, China: National University of Defense Technology, 2008.

[27] Shen, Z., Hu, Y., and Ren, Z.
A new on-line planning method for RLV trajectory design.
Journal of Astronautics, **32**, 8 (2011), 1670–1675.

[28] Karaboga, D., Ozturk, C., Karaboga, N., and Gorkemli, B.
Artificial bee colony programming for symbolic regression.
Information Science, **209** (2012), 1–15.

[29] Li, S. T., and Duan, H. B.
Artificial bee colony approach to online parameters identification for hypersonic vehicle.
Scientia Sinica Informationis, **42**, 11 (2012), 1350–1363.

[30] Duan, H. B., and Qiao, P. X.
Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning.
International Journal of Intelligent Computing and Cybernetics, **7**, 1 (2014), 24–37.

Haibin Duan (M'07—SM'08) was born in Shandong, China, in 1976. He received a Ph.D. degree from Nanjing University of Aeronautics and Astronautics in 2005. He was an academic visitor of National University of Singapore in 2007, a senior visiting scholar of the University of Suwon of South Korea in 2011, an engineer of the Aviation Industry Corporation of China (AVIC) in 2006, and a technician of the AVIC Aviation Motor Control System Institute from 1996 to 2000. He is a full professor with the School of Automation Science and Electrical Engineering, Beihang University (formerly Beijing University of Aeronautics and Astronautics, BUAA) and is the head of the BUAA Bio-Inspired Autonomous Flight Systems (BAFS) Research Group. He has written or cowritten more than 70 publications. His research interests are bio-inspired computation, multiple unmanned aerial vehicle (UAV) formation control, and biological computer vision.

Shuangtian Li was born in Shanxi, China, in 1989. He received a B.S. degree in automation from Beihang University (formerly BUAA), Beijing, China, in 2012. He is a M.S. candidate with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University. His research interests are bio-inspired computation and multiple UAV cooperative control.