Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 8135274, 13 pages
https://doi.org/10.1155/2018/8135274

Research Article

Improved Chicken Swarm Optimization Method for Reentry Trajectory Optimization

Yu Wu,[1] **Bo Yan**,[1] **and Xiangju Qu**[2]

[1] *College of Aerospace Engineering, Chongqing University, Chongqing 400044, China*
[2] *School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Yu Wu; cquwuyu@cqu.edu.cn

Received 18 September 2017; Revised 17 December 2017; Accepted 2 January 2018; Published 31 January 2018

Academic Editor: Antonino Laudani

Copyright © 2018 Yu Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Reentry trajectory optimization has been researched as a popular topic because of its wide applications in both military and civilian use. It is a challenging problem owing to its strong nonlinearity in motion equations and constraints. Besides, it is a high-dimensional optimization problem. In this paper, an improved chicken swarm optimization (ICSO) method is proposed considering that the chicken swarm optimization (CSO) method is easy to fall into local optimum when solving high-dimensional optimization problem. Firstly, the model used in this study is described, including its characteristic, the nonlinear constraints, and cost function. Then, by introducing the crossover operator, the principles and the advantages of the ICSO algorithm are explained. Finally, the ICSO method solving the reentry trajectory optimization problem is proposed. The control variables are discretized at a set of Chebyshev collocation points, and the angle of attack is set to fit with the flight velocity to make the optimization efficient. Based on those operations, the process of ICSO method is depicted. Experiments are conducted using five algorithms under different indexes, and the superiority of the proposed ICSO algorithm is demonstrated. Another group of experiments are carried out to investigate the influence of hen percentage on the algorithm's performance.

1. Introduction

In recent years, the study on hypersonic vehicle has a rapid development with human's exploration in space [1]. The hypersonic vehicle has a wide range of applications in both military and civilian use, such as global strike and space transportation. To realize an effective and reliable flight, generation of an optimal trajectory is one of the key technologies to enhance the global reach capability of hypersonic vehicle. Reentry trajectory optimization has always been considered as a difficult problem due to its strong nonlinearity in motion equations and constraints [2]. Reentry trajectory optimization is an optimal control problem, and there are two categories of methods to solve such problems, that is, indirect methods and direct methods [3]. In indirect methods, the solution model is transformed into a Hamiltonian boundary-value problem through variation method or Pontryagin maximum principle, and then the problem can be solved with the Newton method, conjugate gradient method, and so on [4]. A high accuracy of solution is the main advantage of the indirect method, while a relatively accurate guess of initial solution limits its application. The main idea of the direct methods is to discretize the dynamic process of system and convert the optimal control problem to a nonlinear programming problem [5]. Compared to the indirect methods, the constraints and the objective function are not necessarily continuous and differentiable in the direct methods. However, the direct methods have a slower convergence rate and cannot guarantee finding the global optimal solution. Among the direct methods, the metaheuristic algorithms are widely applied in solving the combinatorial optimization problem. Those algorithms are a combination of stochastic algorithm and local search algorithm, and the deviation of the optimal solution from the feasible solution usually cannot be estimated.

Swarm intelligence algorithms are a class of metaheuristic algorithms which have been paid more and more attention over the past decade. These algorithms mimic the mechanism of information exchange, information sharing, and learning among swarm animals in natural environment [6]. What these algorithms have in common is that they get the solution

with good quality after iteration [7]. Swarm intelligence algorithms have attracted great research interest because they can make a balance between the convergence rate and the solution quality [8]. For example, the artificial bee colony (ABC) algorithm was originally developed by Karaboga and Basturk in 2007 and was inspired by the collective behavior of honey bees [9]. The ABC algorithm has shown a better performance in the function optimization problem compared to the genetic algorithm (GA), the differential evolution (DE) algorithm, and the particle swarm optimization (PSO) algorithm [10]. Inspired by the behavior of homing pigeons, pigeon-inspired optimization (PIO) algorithm was first proposed by Duan and Qiao in 2014 [11]. Based on the elements for pigeons to find their homes, the algorithm consists of two operators, that is, map and compass operator and landmark operator. The feasibility and rapidness of the algorithm have been proven in controlling parameters optimization problem [12].

New algorithms are still being studied. The chicken swarm optimization (CSO) algorithm was originally proposed by Meng et al. in 2014 [13]. It is a stochastic optimization algorithm and mimics the hierarchical order and the behaviors of the chicken swarm. The whole chicken swarm is divided into several subgroups, and there are one rooster, some hens, and several chicks in each subgroup. The three kinds of chicken search for food, respectively, according to their different movement principles. Besides, hierarchy exists in each subgroup, and there is competition between different subgroups. The above characteristics of chicken swarm make the algorithm search for the global optimum. In [13], CSO algorithm was used to solve the benchmark function optimization problem, and the experimental results demonstrate that CSO algorithm performs better in terms of convergence rate and optimization accuracy compared to PSO, DE algorithm, and bat algorithm (BA). However, the benchmark functions were tested only in a low-dimensional case. To verify the validity of CSO algorithm in high-dimensional case, standard test functions are used in the experiment of [14]. The results indicate that the CSO algorithm is easy to fall into local optimum, and premature convergence occurs [14].

As far as we know, the CSO algorithm has not been applied in solving the reentry trajectory optimization problem in the reported studies. Other swarm intelligence algorithms, such as PSO algorithm [15], ant colony optimization (ACO) algorithm [16], and artificial bee colony (ABC) algorithm [17] have been used to obtain the reentry trajectories for hypersonic vehicle. In these studies, the problem is formulated into a high-dimensional optimization problem with strong nonlinearity and multiple constraints, and the control variables of vehicle in different moments are optimized by various swarm intelligence algorithms, respectively. Compared to the low-dimensional cases, the solution space is expanded rapidly in the reentry trajectory optimization problem. As the scale of the chicken swarm is fixed in the original CSO, it is more likely to fail to obtain the global optimum. Therefore, the CSO algorithm is required to be modified to improve its performance in solving the high-dimensional optimization problem. Also, the improved version of CSO is expected to show its superiority in solution quality and convergence rate over other swarm intelligence algorithms.

To this end, an improved chicken swarm optimization (ICSO) method solving the reentry trajectory optimization problem is proposed for the first time in this paper. To be specific, the movement principle of hen is modified based on that in original CSO algorithm. The reasons are as follows. Firstly, the number of hens accounts for the majority of the chicken swarm, so the movement of hen has a great influence on the algorithm's performance. Secondly, there are more parameters in the equation of updating the hen's movement, which also affects the algorithm's performance. A crossover operator is introduced in the ICSO algorithm after the hens finish updating their positions. With the crossover operator, the hens with the worst fitness values are substituted for the new offspring, and the new offspring have certain probability to be better solutions compared to those in the original CSO algorithm. Thus, the local optimal solution and the premature convergence are avoided. Besides, the proposed ICSO algorithm can speed up the convergence rate.

The remainder of the paper is organized as follows. Section 2 presents the dynamic equations and the aerodynamic model of the hypersonic vehicle studied in this paper. The constraints and the cost function are also formulated in this section. Section 3 introduces the basic principle of CSO algorithm, and how the CSO algorithm is improved is also highlighted. In Section 4, the details of ICSO method solving the reentry trajectory optimization problem are elaborated. Comparison experiments among several swarm intelligence algorithms are conducted in Section 5, and the influence of the hen percentage on the algorithm's performance is further explained. The concluding remarks are contained in the last section.

2. Description of Reentry Trajectory Optimization Problem

In this section, the reentry trajectory optimization problem is formulated into the optimal control problem. Then the model used in this study, namely, the common aero vehicle (CAV), is described, including the dynamic model and the aerodynamic model. Besides, the constraints and the cost function are proposed to establish the optimization model of the problem.

2.1. Characteristics of CAV. There are two kinds of CAV, that is, CAV-L and CAV-H. CAV-L means low-performance CAV and has a hypersonic lift to drag ratio (L/D) in the 2.0–2.5 range. CAV-H is a lifting body design applicable to high performance with hypersonic L/Ds in the 3.5–5.0 range [18]. In this study, the model of CAV-H is used. Compared to CAV-L, CAV-H has greater down-range and cross-range gliding ability.

In the reentry trajectory optimization problem discussed in this paper, the CAV-H is assumed to fly over aspherical rotating Earth. The 3-DOF motion equations of CAV-H are as follows [19]:

$$\frac{dr}{dt} = V \cdot \sin\gamma,$$

$$\frac{d\theta}{dt} = \frac{V\cos\gamma\sin\Psi}{r\cos\varphi},$$

$$\begin{aligned}
\frac{d\varphi}{dt} &= \frac{V\cos\gamma\cos\Psi}{r},\\
\frac{dV}{dt} &= \omega^2 r\cos\varphi\left(\sin\gamma\cos\varphi-\cos\gamma\sin\varphi\cos\Psi\right)-\frac{D}{m}\\
&\quad -g\sin\gamma,\\
\frac{d\gamma}{dt} &= \frac{1}{V}\left[\frac{L\cos\sigma}{m}+\left(\frac{V^2}{r}-g\right)\cos\gamma\right.\\
&\quad +2\omega V\cos\varphi\sin\Psi\\
&\quad \left.+\omega^2 r\cos\varphi\left(\cos\gamma\cos\varphi+\sin\gamma\sin\varphi\cos\Psi\right)\right],\\
\frac{d\Psi}{dt} &= \frac{1}{V}\left[\frac{L\sin\sigma}{m\cos\gamma}-2\omega V\left(\cos\varphi\tan\gamma\cos\Psi-\sin\varphi\right)\right.\\
&\quad \left.+\frac{V^2}{r}\cos\gamma\sin\Psi\tan\varphi+\frac{\omega^2 r}{\cos\gamma}\sin\gamma\cos\varphi\sin\Psi\right],
\end{aligned} \tag{1}$$

where r is the radial distance from the center of the Earth to the vehicle, θ is the vehicle's longitude, and φ is the latitude. In addition, V is the flight velocity, γ is the flight path angle, and Ψ is the heading angle. g and ω represent the gravity constant and rotational angular velocity of the Earth. According to (1), the state vector is defined as $\mathbf{x}(t) = [r\ \ \theta\ \ \varphi\ \ V\ \ \gamma\ \ \Psi]$. The terms L and D are aerodynamic lift and drag and have forms as follows:

$$\begin{aligned}
L &= C_L\cdot q\cdot S,\\
D &= C_D\cdot q\cdot S,
\end{aligned} \tag{2}$$

where q is the dynamic pressure and can be further expressed as $q = 0.5\cdot\rho\cdot V^2$; S is the reference area, and ρ is the density of the atmosphere. C_L and C_D are both nonlinear functions of angle of attack α and Mach number (Ma). Aerodynamic data can be obtained by consulting [20].

From (1) to (2), when the values of α and the bank angle σ at every moment are known, the state of vehicle can be calculated with numerical method under a given initial state vector $\mathbf{x}$. In the reentry trajectory optimization problem, α and σ are the control variables required to be optimized for making the vehicle from a given initial state to a specified terminal state.

2.2. Constraints of Reentry. During the reentry phase, the constraints of trajectory, terminal state, and control variables must be obeyed to ensure a safe and accurate flight. These constraints are described as follows.

(1) Trajectory Constraints. This category of constraints includes those on dynamic pressure q, overload n, and aerodynamic heating rate $\dot{Q}$. They must be satisfied every moment to ensure the flight safety and structural safety of vehicle. The mathematical expressions of these constraints are listed as follows:

$$\begin{aligned}
q &= 0.5\cdot\rho\cdot V^2 < q_{\max},\\
n &= \frac{\sqrt{D^2+L^2}}{(m\cdot g)} \le n_{\max},\\
\dot{Q} &= K\cdot\sqrt{\rho}\cdot V^3 \le \dot{Q}_{\max}.
\end{aligned} \tag{3}$$

The dynamic pressure q is in Newton per square meter, and the overload n is in g. The aerodynamic heating rate $\dot{Q}$, in watt per square meter, is defined at the stagnation point on the surface of vehicle because the stagnation point is the most seriously heated region. In (11), K is a constant and it is common to take $K = 5.21\times10^{-8}$ [21].

(2) Terminal Constraints. To guarantee precise guide in terminal stage of landing, the terminal constraints must be met after reentry phase. The error of radius, longitude, and latitude must satisfy the following constraints:

$$\begin{aligned}
e_r &= \left|r\left(t_f\right)-r_f\right| \le e_{r\max},\\
e_\theta &= \left|\theta\left(t_f\right)-\theta_f\right| \le e_{\theta\max},\\
e_\varphi &= \left|\varphi\left(t_f\right)-\varphi_f\right| \le e_{\varphi\max}.
\end{aligned} \tag{4}$$

(3) Constraints of Control Variables. The control variables α and σ must be kept within a certain range to meet the physical characteristics of the actuator. The above constraints are expressed in the following equation:

$$\begin{aligned}
\alpha_{\min} &\le \alpha \le \alpha_{\max},\\
\sigma_{\min} &\le \sigma \le \sigma_{\max}.
\end{aligned} \tag{5}$$

2.3. Cost Function in Reentry Trajectory Optimization Problem. The goal of reentry tasks is not always the same. It should be determined based on the demand and preference of decision-makers. Usually, to make a better performance of vehicle and an efficient flight, the following three cost functions are proposed:

$$\begin{aligned}
J_Q &= \min\int_{t_0}^{t_f}\dot{Q}\,dt,\\
J_t &= \min t_f,\\
J_\theta &= \max\left|\theta\left(t_f\right)-\theta\left(t_0\right)\right|.
\end{aligned} \tag{6}$$

In (6), the integral of the aerodynamic heating rate of the stagnation point is the optimization goal to reduce the burden and the weight of the thermal protection system. J_t ensures that the reentry phase is finished within the shortest time in (18). To develop the down-range gliding ability of vehicle, maximum down-range in longitude profile is regarded as the cost function in (19). In experimental studies, any of the above three cost functions can be selected to further analyze the results.

3. Principles of the ICSO Algorithm

Chickens are one kind of social animals and have their unique way to search for food. Although one chicken has a low efficiency of searching for food, the chicken swarm can always find food quickly by cooperation [22]. Motivated by such self-organized social behavior, Meng proposed the CSO algorithm in 2014. In this section, the basics of CSO algorithm are introduced, and then a detailed description of how the CSO algorithm is improved will be presented.

3.1. Basics of CSO Algorithm. To make a better understanding of CSO algorithm, the basics of CSO algorithm will be introduced from three aspects, that is, the structure of chicken swarm, the rules of updating positions, and the change of hierarchical order.

(1) The Structure of Chicken Swarm. There are several subgroups in the chicken swarm, and, in each subgroup, one rooster, some hens, and several chicks are included. The chickens with the best several fitness values would be selected as roosters; that is to say, the number of roosters equals the number of subgroups. The chickens with the worst several fitness values would be assumed as chicks, and other chickens are the hens. The number of hens is usually the largest. In a subgroup, hens and chicks are randomly assigned, and a certain number of hens are chosen to establish mother-child relationship with the chicks randomly.

(2) The Rules of Updating Positions. Chickens have different behaviors when they are searching for food according to their roles in the subgroup. Roosters are the strongest in each subgroup, so they search for food for their subgroups selflessly. Besides, they have duty on the safety of chickens and territory in the subgroup. Hens follow the rooster in the same subgroup to search for food, and they can also randomly steal the food from other chickens and prevent others from eating their own food. Chicks are the weakest and are raised by their mother, so they imitate their mother's behavior to search for food. Since the chickens' behaviors vary in the subgroup, the rules of updating their positions are also different according to their roles, as seen in the following equations:

$$x_{i,j}^{t+1} = x_{i,j}^{t} \cdot (1 + \text{Rand1}), \tag{7}$$

$$\sigma^2 = \begin{cases} 1 & f_i \le f_k, \\ \exp\left(\dfrac{f_k - f_i}{|f_i| + \varepsilon}\right) & \text{otherwise}, \end{cases} \tag{8}$$

$$k \in [1, N], \ k \ne i,$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + c_1 \cdot \text{Rand2} \cdot \left(x_{r1,j}^{t} - x_{i,j}^{t}\right) + c_2 \cdot \text{Rand3} \cdot \left(x_{r2,j}^{t} - x_{i,j}^{t}\right) \quad (r_1 \ne r_2), \tag{9}$$

$$c_1 = \exp\left(\frac{f_i - f_{r1}}{|f_i| + \varepsilon}\right), \tag{10}$$

$$c_2 = \exp\left(f_{r2} - f_i\right), \tag{11}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \text{Rand4} \cdot \left(x_{m,j}^{t} - x_{i,j}^{t}\right). \tag{12}$$

In the above equations, $x_{ij}^{(t)}$ ($i \in [1, 2, \ldots, N]$, $j \in [1, 2, \ldots, D]$, $t \in [1, 2, \ldots, \text{Iter}]$) denotes the jth element of the ith chicken in the tth iteration, where N is the number of chickens in the swarm, D is the dimension of the solution space, and Iter is the maximum iteration times. Equations (7) and (8) are the updating rules of rooster's position. Rand1 is a random number obeying normal distribution with mean 0 and standard deviation σ^2. In (8), k is a randomly selected rooster's index; f_i is the fitness value of the corresponding chicken i; ε is the smallest constant in the computer and makes the equation meaningful. Such a position update rule enables the roosters to search for food in a wider range of places. Equations (9)–(11) are the updating rules of hen's position. Rand2 and Rand3 are two random numbers in the interval $[0, 1]$. $r1$ is the index of a rooster, which is in the same subgroup as the ith hen. $r2$ is the index of a randomly selected chicken (rooster or hen). This rule shows two behaviors of hens when searching for food, that is, following the rooster in the same subgroup to search for food and stealing the food from other chickens. As for the chicks, they move around their mother to search for food, and this behavior can be expressed as (12), where m is the index of chick i's mother and Rand4 is a parameter that usually takes the value in the interval $(0, 2)$.

(3) The Change of Hierarchical Order. In the chicken swarm, hierarchical order will not be kept unchanged all the time. To reflect the change of each chicken's fitness value timely, parameter G is introduced. The roosters, hens, chicks, and mothers will be redefined according to their fitness values every G iterations. This mechanism can avoid the solution trapping into the local optimum to some extent and make the search move towards the global optimum.

3.2. Description of ICSO Algorithm. Reentry trajectory optimization is a high-dimensional optimization problem, and the results in [14] indicate that the CSO algorithm is easy to fall into local optimum. Therefore, the CSO algorithm needs to be modified to qualify for solving the reentry trajectory optimization problem.

In CSO algorithm, the number of hens is the largest in the chicken swarm, and there are more variable parameters in hen's position updating rule. The hens' fitness values have great influence on the algorithm performance. In hen's position updating rule, the rooster's position is referred. Compared to other intelligent optimization algorithms, for example, PSO algorithm, this operation can make the hens keep away from the local optimum, and ultimately the algorithm has less possibility to be trapped into local optimum. However, when more roosters are trapped into the local optimum, the hens will also face the same situation, which results in missing the global optimum.

To overcome the weakness of CSO algorithm, a crossover operator is introduced in ICSO algorithm. After the hens

update their position, a random number $a \in (0,1)$ is generated. If a is smaller than the default crossover probability, the two hens with the best fitness values are selected to do the crossover operation. The information of new offspring can be expressed as the following equations:

$$\begin{aligned} \text{offspring1} &= p * \text{hen1} + (1-p) * \text{hen2}, \\ \text{offspring2} &= (1-p) * \text{hen1} + p * \text{hen2}. \end{aligned} \tag{13}$$

In (13), parameter p represents the information percentage that the offspring can inherit the corresponding hen. After the crossover operation, the two new offspring will substitute the two hens with the worst fitness values and continue the iterative process with other chickens. When there are hens trapped into the local optimum, the ICSO algorithm can change the position of hens with the crossover operator. Therefore, the fitness values of the new offspring are better than those of roosters at a certain probability. With the mechanism of hierarchical order change in CSO algorithm, the roosters will be replaced by the hens with better fitness values. This mechanism can not only substitute the roosters which may be trapped into the local optimum but also improve the search speed. Ultimately, the roosters have faster speed towards the global optimum, which accelerate the convergence rate.

4. The ICSO Method Solving the Reentry Trajectory Optimization Problem

When using the ICSO algorithm to solve the reentry trajectory optimization problem, the control variable is discretized at a set of Chebyshev collocation points based on the flight time t_f. Only the control variable at these moments is optimized, and the control variable at other moments can be obtained by interpolation. Thus, the reentry trajectory optimization problem is transformed into a high-dimensional control variable optimization problem and can be solved with the ICSO algorithm.

4.1. Determination of Collocation Points. In the Chebyshev pseudospectral method [23], the collocation points are distributed according to the values of T_k, as shown in the following equation:

$$T_k = \cos\left(\frac{(n-k)\cdot\pi}{n}\right) \quad (k = 0, 1, 2, \ldots, n). \tag{14}$$

The number of collocation points is $n+1$, and T_k is the value of collocation point k. The Chebyshev points are dense at both ends and sparse in the middle. This feature exactly meets the demand of precise control near the destination. It is obvious that $T_k \in [-1, 1]$, and T_k should be converted to the time interval $[t_0, t_f]$ using the following equation:

$$t_k = \frac{t_f - t_0}{2} \cdot T_k + \frac{t_f + t_0}{2}. \tag{15}$$

Ultimately, the control variable at the moments t_k ($k = 0, 1, \ldots, n$) needs to be optimized.

4.2. Operation of Control Variables. There are two control variables in the reentry trajectory optimization problem, that is, angle of attack α and bank angle σ. To have a lower peak heating rate, α is usually set to the value with the maximum L/D ratio. A nominal angle of attack profile corresponding to the maximum L/D ratio of CAV-H [24] is shown in Figure 1.

With the relationship between α and Mach number in Figure 1, the value of control variable α can be determined by the flight velocity of vehicle. Thus, there are only two control variables to be optimized in this problem.

As mentioned before, only the control variable σ at moment t_k is optimized, and the values of σ at other time points can be obtained by interpolation. Here linear interpolation is adopted, and σ_{t_i} ($t_k < t_i < t_{k+1}$) can be calculated by the following equation:

$$\sigma_{t_i} = \frac{\sigma_{t_{k+1}} - \sigma_{t_k}}{t_{k+1} - t_k} \cdot (t_i - t_k) + \sigma_{t_k}. \tag{16}$$

The values of σ_{t_k} and $\sigma_{t_{k+1}}$ are known, so σ_{t_i} can be obtained by (16).

4.3. Process of ICSO Method. The procedure of ICSO method solving reentry trajectory optimization problem is depicted as follows:

(1) Parameter initialization: the total number of chickens (N), the maximum iterations (Iter), the frequency of change of hierarchical order in the swarm (G), and the numbers of roosters, hens, chicks, and mothers (N_r, N_h, N_c and N_m) are initialized, respectively. Besides, the initialization of solution dimension (D, is also the number of collocation points) and the flight time t_{f_i} should be conducted before the determination of collocation points. Finally, the initial values of bank angle σ are calculated by the following equation:

$$\sigma_{i,j} = \sigma_{\min} + (\sigma_{\max} - \sigma_{\min}) \cdot \text{rand}, \tag{17}$$

where $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, D$, and rand is a random number in the interval $[0, 1]$. Then the initial values of bank angle σ at other time points can be obtained using (16). Under a given initial state of vehicle $\mathbf{x}_0$, the state at every moment can be calculated with (1), and the fitness value of each chicken is also known

(2) Generation of hierarchical order in the swarm: the initial hierarchical order is defined according to the fitness value of chicken. If the number of iteration times t is divisible by G, the hierarchical order will be updated

(3) Update the positions of roosters, hens, and chicks and recalculate the fitness values, respectively: roosters, hens, and chicks update their positions according to (7), (9), and (12), respectively. Then their fitness values are recalculated using (1)

(4) Select the two hens with the best fitness values to do the crossover operation: new offspring are generated using (13). They replace the two hens with the worst fitness values

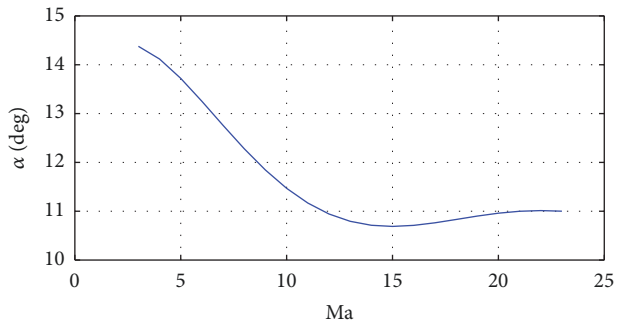

Figure 1: Nominal angle of attack profiles for CAV-H.

(5) Update individual's best fitness value and the global best solution: in this iteration, if the chicken's fitness value is better than the previous best one, the chicken's best fitness value or the global best solution will be updated

(6) End of one iteration: if the number of iteration times t is less than Iter, return to Step (2). Otherwise, terminate the algorithm and output the global best solution. The control variables corresponding to the global best solution are the optimal control variables of the reentry trajectory optimization problem.

The flow chart of the ICSO method is presented in Figure 2, and the blue box denotes the contents modified based on CSO algorithm.

5. Experimental Studies

To investigate the feasibility of the ICSO method solving the reentry trajectory optimization problem, three groups of experiments are conducted. In the first two groups of experiments, the minimum aerodynamic heating rate and the minimum flight time are the cost functions. Comparisons among five swarm intelligence algorithms are made, and the statistical significance tests are also carried out to show the improved performance of ICSO algorithm. In the third group of experiments, the impacts of hen percentage on the algorithm's performance are explained.

The parameters used in both CSO and ICSO algorithms are set in Table 1. Besides, in ICSO algorithm, random number $a = 0.25$ and information percentage $p = 0.5$. The parameters settings of ACO, PSO, and ABC use the data in [16], [25], and [17], respectively. In the experiments, each algorithm is run for 100 independent trials to obtain the statistical results. For a fair comparison, all of the common parameters of these algorithms, such as the population size, dimensions, and maximum number of iterations, are set to be the same. The initial states of CAV-H and the constraints are given in Tables 2 and 3.

5.1. Reentry Trajectory with the Minimum Aerodynamic Heating Rate. Here a comparative study is conducted, and the cost function in (6) is taken as the fitness function in each algorithm. Figure 3 shows the best results in 100 independent trials with different algorithms. The trajectories of CAV-H and the trajectory constraints during flight corresponding to the best results are illustrated in Figures 4 and 5.

In Figure 3, the fitness values of the five algorithms are all kept at 10^{10} in the first few iterations because no solution satisfies all constraints, and fitness value is set to be a larger value (10^{10}) to denote the abandoned solution. The ICSO algorithm has the smallest fitness value after 100 times of iterations. Besides, the fitness values can all converge within 100 iterations with the five algorithms. The ICSO algorithm has the fastest convergence rate (about 50 iterations), while the ACO and PSO algorithms perform the worst (about 90 iterations). These results demonstrate that the ICSO algorithm is better than the other four algorithms both in minimizing the fitness value and in improving the convergence rate. The reason why the ICSO algorithm results in the best performance is that the PSO algorithm is a special case of the CSO algorithm under some simplification. When $N_r = 0$ and $N_h = 0$, the parameters c_1 and c_2 in CSO algorithm are similar to those in PSO algorithm. Therefore, the CSO algorithm can give full play to its advantages based on inheriting the advantages of PSO algorithm. Compared to the CSO algorithm, the crossover operator is introduced to the ICSO algorithm, and the solution qualities corresponding to hens are improved. The hens with the best solution qualities will replace the roosters when the hierarchical order of the chicken swarm is updated. This mechanism will accelerate the search speed of rooster, thus making the roosters move towards the global optimum faster. Those advantages make the ICSO algorithm perform the best.

In Table 4, the fitness value, the terminal error of position, and the flight time of vehicle corresponding to the best results of each algorithm are given.

In Figure 5 and Table 4, all of the results satisfy the constraints set in Table 3. That is, the flight safety and structural safety of CAV-H are guaranteed, and the terminal position of vehicle is taken within the error range, which guarantees a precise guide in the terminal stage of landing. Note that the values of aerodynamic heating rate in Figures 3 and 5 show different meanings. In Figure 3, the total value of aerodynamic heating rate after a complete reentry flight is presented

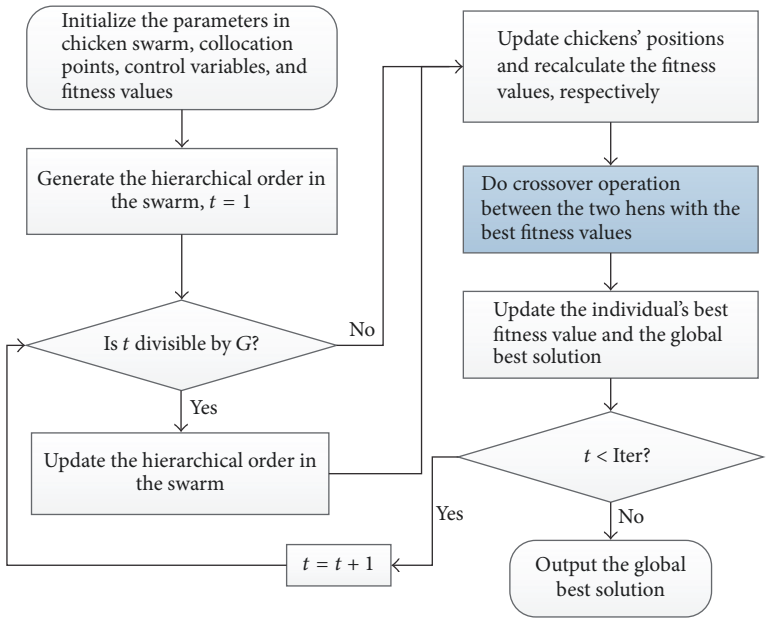

Figure 2: Flow chart of the ICSO method.

Table 1: Parameters setting in CSO method and ICSO method.

Parameter	Iter	N	D	G	N_r	N_h	N_m
Value	100	100	60	5	25	65	50

Table 2: The initial states of CAV-H.

State	$h(t_0)$	$\theta(t_0)$	$\varphi(t_0)$	$V(t_0)$	$\gamma(t_0)$	$\psi(t_0)$
Value	121.9 km	120 deg	−40 deg	7400 m/s	−1 deg	61.39 deg

Table 3: Constraints of reentry.

Trajectory constraints	q (MPa)	n (g)	$\dot{Q}$ (KW/m^2)
	≤70	≤2.5	≤5000
Terminal constraints	$h(t_f)$	$\theta(t_f)$	$\gamma(t_f)$
	28 ± 1 km	230 ± 0.5 deg	50 ± 0.5 deg
Control variable	α	σ	
	10°–15°	−100°–100°	

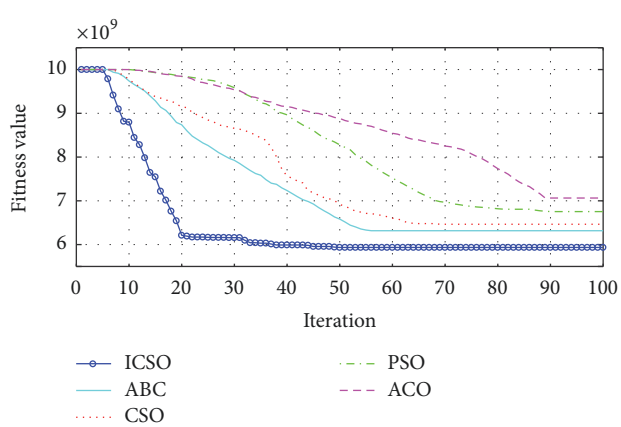

Figure 3: Fitness values of ICSO, ABC, CSO, PSO, and ACO algorithms under the cost function of the minimum aerodynamic heating rate.

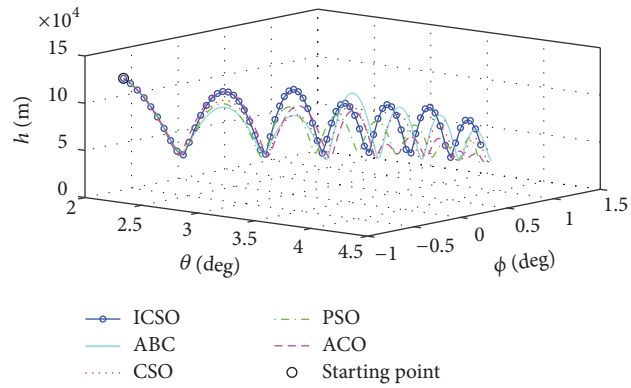

Figure 4: Trajectories of CAV-H with different algorithms under the cost function of the minimum aerodynamic heating rate.

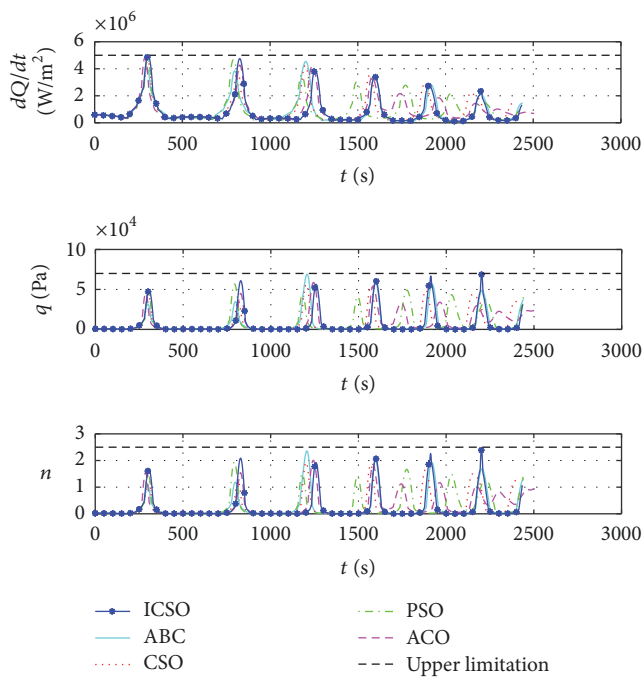

Figure 5: The values of aerodynamic heating rate, dynamic pressure, and overload during flight under the cost function of the minimum aerodynamic heating rate.

with the increase of the iterations, while the value of aerodynamic heating rate of each moment during flight is shown in Figure 5. It is reasonable that the value of aerodynamic heating rate with the ICSO algorithm is not the minimum in some specific moments in Figure 5, and the sum is the minimum in Figure 5 after the reentry flight. The graph about the variation of bank angle σ is shown in Figure 6.

To further explain the superiority of the proposed ICSO algorithm in solving the reentry trajectory optimization problem, the average and standard deviation of the best solution in the last iteration after each algorithm is run 100 times are reported, as presented in Table 5.

The results in Table 5 demonstrate that the proposed ICSO algorithm behaves more stably when solving the reentry trajectory optimization problem. However, the averages and standard deviation only compare the overall performance of the algorithms, and a statistical test considering each run's results is needed to prove that the results are statistically significant. The Wilcoxon rank-sum test is conducted in this study.

The Wilcoxon rank-sum test is a nonparametric test in statics which can be used to verify if two sets of solutions are statistically significant or not [26]. This statistical test returns a parameter called p value. A p value determines the significance level of two algorithms. An algorithm is statistically significant only if it results in a p value less than 0.05. The test results are shown in Table 6.

The p values in Table 6 also show that the superiority of the ICSO algorithm is statistically significant because its

Table 4: Results of experiment in detail.

Algorithm	Fitness value (10^9)	e_r (km)	e_θ (deg)	e_φ (deg)	t_f (s)
ACO	7.0638	0.612	0.3316	−0.4526	2504.2
PSO	6.7538	0.575	−0.2536	−0.4323	2438.9
CSO	6.4643	0.559	0.2969	−0.3491	2402.9
ABC	6.3165	0.525	−0.3742	0.3698	2439.7
ICSO	5.9343	0.413	−0.2668	0.3959	2434.9

Table 5: Statistical comparison of five algorithms.

Algorithm	Best (10^9)	Mean (10^9)	Worst (10^9)	Std.
ACO	7.0638	7.0772	7.0904	782.4
PSO	6.7538	6.7681	6.7704	591.4
CSO	6.4643	6.4684	6.4703	273.5
ABC	6.3165	6.3176	6.3204	203.8
ICSO	5.9343	5.9356	5.9382	126.3

Table 6: p values when the ICSO algorithm is compared against each of the other algorithms.

Algorithm	ACO	PSO	CSO	ABC
p value	0.0283	0.0251	0.0183	0.0148

Table 7: Experimental results with the shortest flight time.

Algorithm	Fitness value (t_f (s))	e_r (km)	e_θ (deg)	e_φ (deg)
ACO	2395.6	0.530	−0.4302	−0.0749
PSO	2361.6	0.525	−0.2820	−0.4632
CSO	2282.6	0.740	−0.3175	0.4145
ABC	2292.8	0.484	−0.3686	−0.4257
ICSO	2254.4	0.588	−0.3410	−0.4632

p value is much less than 0.05 and is the smallest among the five algorithms.

5.2. Reentry Trajectory with the Minimum Flight Time. In the following experiments, the flight time is taken as the fitness function. Like the experiments conducted in Section 5.1, the same algorithms are used to obtain the reentry trajectories and the same parameters settings, initial state of vehicle, and constraints are used. Each algorithm is also run 100 times to get the statistical results. The variation of fitness values of the best solution in 100 trials is shown in Figure 7, and the corresponding trajectories of CAV-H are illustrated in Figure 8.

The fitness values of the five algorithms are all beginning with the value 4000 because no solution satisfies all constraints, and fitness value is set to be a larger value (4000) to denote the abandoned solution. Similar to the results in Figure 3, the smallest fitness value is obtained using the ICSO algorithm, and the ICSO algorithm converges with the minimum iteration times (about 30 iterations). Different from the results in Section 5.1, the iteration times before the algorithm converges are less in general. This is because the flight time has been limited to a certain range in advance to guarantee a feasible flight. Under the known interval which is relatively short, the flight time can converge faster. The results in Figure 7 demonstrate the validity and the superiority of the ICSO algorithm once again.

To explain the feasibility of the generated trajectories, the variation of aerodynamic heating rate, dynamic pressure, overload, and bank angle and the value of fitness function and terminal position error corresponding to the best solution of each algorithm are presented in Figures 9 and 10 and Table 7, respectively.

The results in Table 7 demonstrate that the reentry trajectories obtained by different algorithms can meet all of the proposed constraints that guarantee the flight safety and guidance precision.

To test the stability and statistical significance of the algorithms under the index of the shortest flight time, the average, standard deviation, and p value (in Wilcoxon rank-sum test) of each algorithm after 100 times of independent trials are listed, as shown in Tables 8 and 9.

The results in Tables 8 and 9 demonstrate that the proposed ICSO algorithm performs better than the other four algorithms in terms of stability and statistical significance when the shortest flight time is regarded as the index.

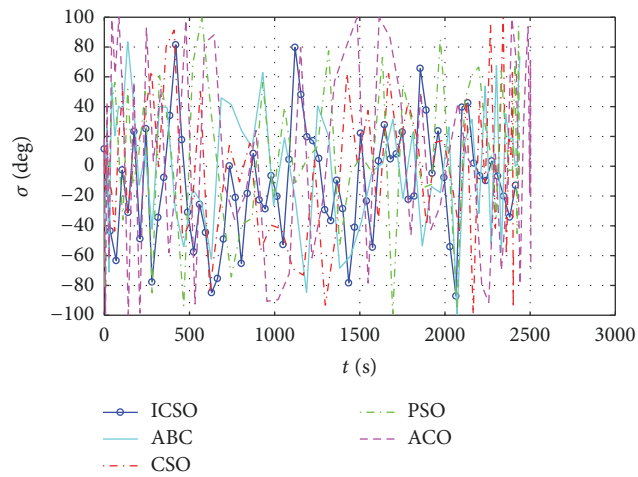

Figure 6: Variation of bank angle with the minimum aerodynamic heating rate.

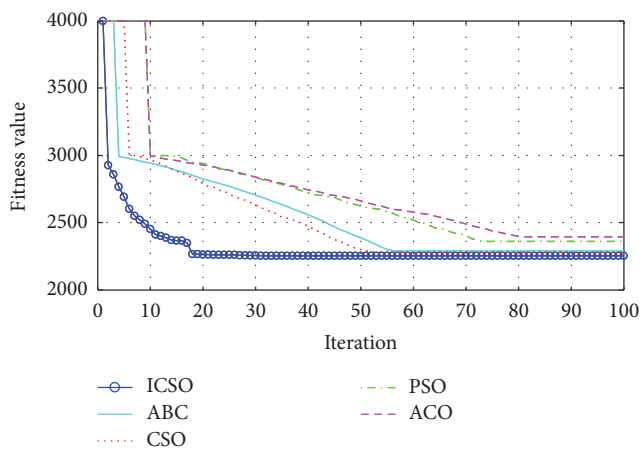

Figure 7: Fitness values under the cost function of the shortest flight time.

5.3. The Influence of Hen on Algorithm's Performance. In the experiments conducted in Sections 5.1 and 5.2, the hen percentage is kept unchanged. Compared to rooster and chick, the hen accounts for the largest number in the chicken swarm, and it has been verified that the performance of CSO algorithm can be improved by adding the crossover operator on hens. In the following experiments shown in Figure 11, the percentages of hen are set as 40%, 50%, 55%, and 65%, respectively, and the cost function in (18) is adopted. Other conditions and constraints are the same as those in Sections 5.1 and 5.2. The experiment results of different hen percentages using the ICSO algorithm are presented in Figure 11.

With the change of hen percentage, the convergence rate does not show a big difference (about 50 iterations). However, the fitness value becomes smaller with the increasing hen percentage. This is because, under an invariant number of roosters, the chicken swarm with more hens has a higher probability of generating good new offspring with the crossover operator, thus improving the probability of the hens to be roosters when the hierarchical order is updated. Therefore, roosters with better fitness value are more likely to reach the global optimum in later iteration process. To make the algorithm perform better, the hen percentage should be as large as possible under a suitable percentage of rooster and chick.

6. Conclusion

The reentry trajectory optimization problem of hypersonic vehicle is studied using the proposed ICSO method in this paper. Literature investigation shows that although the CSO algorithm has better performance than other intelligent optimization algorithms in the benchmark function optimization problem, it is easy to fall into local optimum when dealing with the high-dimensional case, like the reentry trajectory optimization problem. Therefore, an improved CSO algorithm is developed. Firstly, the mathematical model of the problem is established, including the constraints and the cost functions. Then the basics of CSO algorithm are described. Based on that, a crossover operator is introduced to the ICSO algorithm. With this operation, the quality of the best solution can be improved at a certain probability, which makes the roosters move towards the global optimum. In the ICSO method, the control variables are discretized

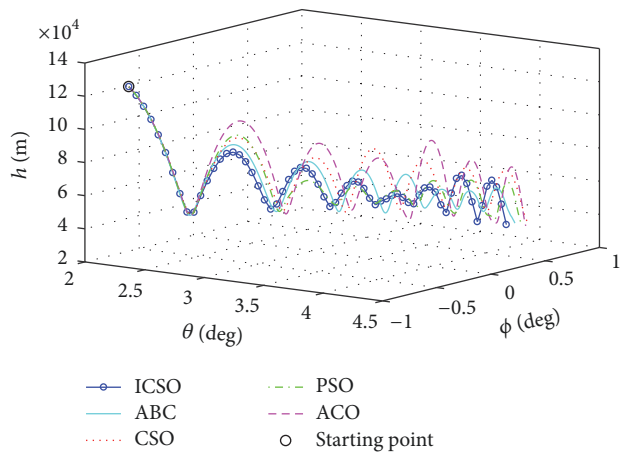

Figure 8: Minimum-time trajectories of CAV-H.

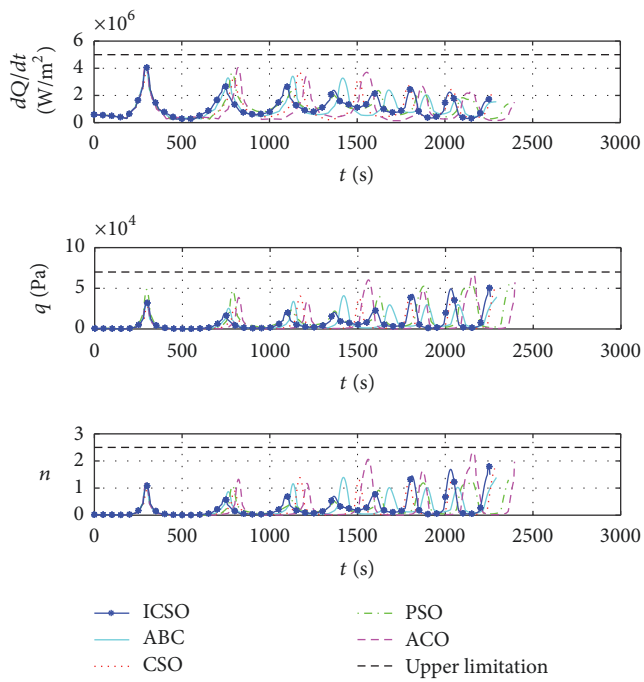

Figure 9: Variation of aerodynamic heating rate, dynamic pressure, and overload with the shortest flight time.

Table 8: Statistical comparison of five algorithms under the index of the shortest flight time.

Algorithm	Best	Mean	Worst	Std.
ACO	2395.6	2406.3	2413.2	0.8395
PSO	2361.6	2369.9	2378.6	0.7293
CSO	2282.6	2290.8	2297.4	0.5382
ABC	2292.8	2299.5	2304.7	0.5135
ICSO	2254.4	2259.0	2268.3	0.1839

Table 9: p values of the Wilcoxon rank-sum test under the index of the shortest flight time.

Algorithm	ACO	PSO	CSO	ABC
p value	0.0226	0.0189	0.0105	0.0117

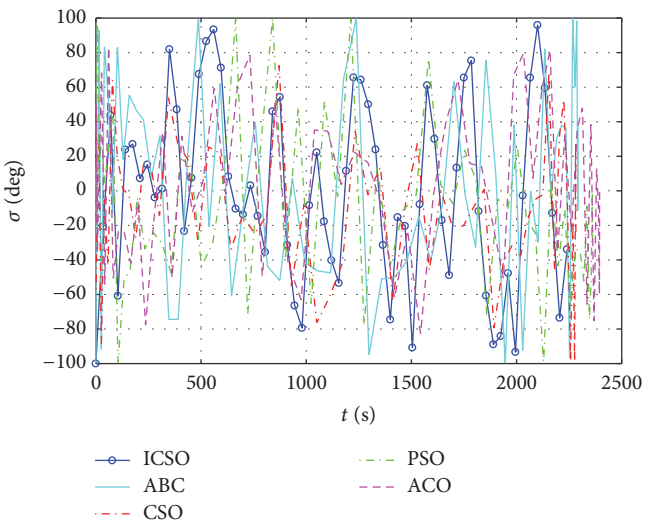

Figure 10: Variation of bank angle with the shortest flight time.

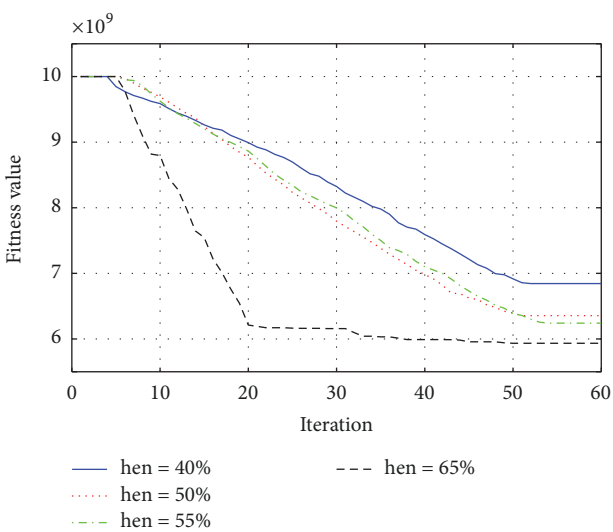

Figure 11: Fitness value with different hen percentage.

at Chebyshev collocation points, thus making it suitable for dealing with the terminal constraints of the problem. In the experiments, under different cost functions, the proposed ICSO approach demonstrates its superiority compared to ACO, PSO, ABC, and CSO approach. The statistical analysis of algorithms is also conducted. To analyze the influence of hen percentage on the algorithm's performance, more experiments are conducted. The experiment results show that, to make the algorithm perform better, the hen percentage can be as large as possible under a suitable percentage of rooster and chick. In the future, other forms of improved CSO method can be further studied to strengthen the robustness of the algorithm so that it can deal with more complicated optimization problems.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research work is financially supported by the Fundamental Research Funds for the Central Universities with the project reference number of 106112017CDJXY320003.

References

[1] J. J. Bertin and R. M. Cummings, "Fifty years of hypersonics: where we've been, where we're going," *Progress in Aerospace Sciences*, vol. 39, no. 6-7, pp. 511–536, 2003.

[2] Y. Xie, L. Liu, G. Tang, and W. Zheng, "Highly constrained entry trajectory generation," *Acta Astronautica*, vol. 88, pp. 44–60, 2013.

[3] W. Zhang, Y. Zhang, W. Li, and Y. Wang, "Path Planning for Rapid Large-Angle Maneuver of Satellites Based on the Gauss Pseudospectral Method," *Mathematical Problems in Engineering*, vol. 2016, Article ID 1081267, 2016.

[4] S. Yang, T. Cui, X. Hao, and D. Yu, "Trajectory optimization for a ramjet-powered vehicle in ascent phase via the Gauss pseudospectral method," *Aerospace Science and Technology*, vol. 67, pp. 88–95, 2017.

[5] Y. Yang and Y. Nan, "Multiphase Return Trajectory Optimization Based on Hybrid Algorithm," *Mathematical Problems in Engineering*, vol. 2016, Article ID 6257050, 2016.

[6] S. Zhang, C. K. M. Lee, H. K. Chan, K. L. Choy, and Z. Wu, "Swarm intelligence applied in green logistics: a literature review," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 154–169, 2015.

[7] C. Gao, Z. Zhen, and H. Gong, "A self-organized search and attack algorithm for multiple unmanned aerial vehicles," *Aerospace Science and Technology*, vol. 54, pp. 229–240, 2016.

[8] S. Lu, J. Zhao, L. Jiang, and H. Liu, "Solving the time-jerk optimal trajectory planning problem of a robot using augmented lagrange constrained particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2017, Article ID 1921479, 10 pages, 2017.

[9] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[10] D. Karaboga and B. Akay, "A comparative study of artificial Bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[11] H. Duan and P. Qiao, "Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning," *International Journal of Intelligent Computing and Cybernetics*, vol. 7, no. 1, pp. 24–37, 2014.

[12] R. Dou and H. Duan, "Lévy flight based pigeon-inspired optimization for control parameters optimization in automatic carrier landing system," *Aerospace Science and Technology*, vol. 61, pp. 11–20, 2017.

[13] X. B. Meng, L. Yu, X. Z. Gao, and H. Z. Zhang, "A new bio-inspired algorithm: chicken swarm optimization," in *Advances in Swarm Intelligence*, vol. 8794 of *Lecture Notes in Computer Science*, pp. 86–94, Springer International, 2014.

[14] D. Wu, F. Kong, W. Gao, Y. Shen, and Z. Ji, "Improved chicken swarm optimization," in *Proceedings of the 5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015*, pp. 681–686, China, June 2015.

[15] J. Zhao and R. Zhou, "Particle swarm optimization applied to hypersonic reentry trajectories," *Chinese Journal of Aeronautics*, vol. 28, no. 3, pp. 823–831, 2015.

[16] X. Jiao and J. Jiang, "Multi-objective optimization of reentry trajectory planning for hypersonic aircraft based on ant colony algorithm," in *Proceedings of the 6th IEEE Chinese Guidance, Navigation and Control Conference, CGNCC 2014*, pp. 1203–1208, China, August 2014.

[17] H. Duan and S. Li, "Artificial bee colony-based direct collocation for reentry trajectory optimization of hypersonic vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 615–626, 2015.

[18] T. H. Phillips, "A common aero vehicle (CAV) model, description, and employment guide," *Schafer Corporation for AFRL and AFSPc*, 2003.

[19] X. Liu, Z. Shen, and P. Lu, "Entry trajectory optimization by second-order cone programming," *Journal of Guidance Control Dynamics*, vol. 39, no. 2, pp. 1–15, 2015.

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

[21] H. Morimoto, "Trajectory optimization for a hypersonic vehicle with constraint," *Dissertation Abstracts International*, vol. 59-03, no. 03, 1997.

[22] K. L. Smith and S. Zielinksi, "The Startling Intelligence of the Common Chicken," *Scientific American*, vol. 2, no. 2, 2014.

[23] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," pp. 3860–3864.

[24] X. Liu and P. Lu, "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 750–765, 2014.

[25] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the International Conference on Evolutionary Programming*, vol. 1447 of *Lecture Notes in Computer Science*, pp. 591–600, Springer, 1998.

[26] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.


Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
The Scientific World Journal
Journal of Probability and Statistics
International Journal of Mathematics and Mathematical Sciences
Hindawi
Submit your manuscripts at
www.hindawi.com
Journal of Optimization
International Journal of Engineering Mathematics
International Journal of Analysis
Journal of Complex Analysis
Advances in Numerical Analysis
Mathematical Problems in Engineering
International Journal of Differential Equations
Discrete Dynamics in Nature and Society
International Journal of Stochastic Analysis
Journal of Mathematics
Journal of Function Spaces
Abstract and Applied Analysis
Advances in Mathematical Physics