

A Hybrid Parallel Harris Hawks Optimization Algorithm for Reusable Launch Vehicle Reentry Trajectory Optimization With No-fly Zones

Ya Su

Tongji University

Ying Dai

Tongji University

Yi Liu (✉ liuyi@tongji.edu.cn)

Tongji University

Research Article

Keywords: Hybrid parallel harris hawks optimization, Oppositional learning, Smoothing technique, Parallel optimization mechanism, Reentry trajectory optimization

Posted Date: May 27th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-554106/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A hybrid parallel harris hawks optimization algorithm for reusable launch vehicle reentry trajectory optimization with no-fly zones

Ya Su · Ying Dai · Yi Liu*

School of Aerospace Engineering and Applied Mechanics, Tongji University,
Shanghai 200092, China

*To whom correspondence should be addressed. Email: liuyi@tongji.edu.cn (Y. Liu)

Abstract

Reentry trajectory optimization is a critical optimal control problem for reusable launch vehicle (RLV) with highly nonlinear dynamic characteristics and complex constraints. In this paper, a hybrid parallel harris hawks optimization (HPHHO) algorithm is proposed to address the problem. HPHHO aims to enhance the performance of existing harris hawks optimization (HHO) algorithm by three strategies including oppositional learning, smoothing technique and parallel optimization mechanism. At the beginning of each iteration, the opposite population is calculated from the current population by the oppositional learning strategy. Following that, the individuals in the two populations are arranged in ascending order on the basis of the fitness function values, and the top half of the resulting population is selected as the initial population. The selected initial population is divided into two equal subpopulations which are assigned to the differential evolution and the HHO algorithm, respectively. The both algorithms operate in parallel to search and update the solutions of each subpopulation simultaneously. Then the solutions are smoothed for each iteration by the smoothing technique to reduce fluctuations. As a result, the optimal solution obtained by the parallel optimization mechanism avoids falling into local optima. The performance of HPHHO is evaluated by 4 CEC 2005 benchmark functions and 3 constrained continuous optimal control problems, showing better efficiency and robustness in terms of performance metrics, convergence rate and stability. Finally, the simulation results show that the proposed algorithm is very effective, practical and feasible in solving the RLV reentry trajectory optimization problem.

Keywords Hybrid parallel harris hawks optimization · Oppositional learning · Smoothing technique · Parallel optimization mechanism · Reentry trajectory optimization

1 Introduction

Reusable launch vehicle (RLV) is a special type of vehicle. After each mission into space, it is used repeatedly by recycling and reuse. It has good prospects in civil and defense applications and is also a goal pursued by different aerospace agencies around the world (Jee et al. 2014). Reentry trajectory optimization is a key part of the conceptual design of RLV and has a significant effect on the design of aerodynamics, structural strength and thermal protection systems, which plays a **critical** role in the safe and effective reentry flight (Zhang et al. 2020).

The philosophy of repeated launching with the same vehicle sounds interesting and fascinating, whereas a major challenging problem posed in flight missions is that of atmospheric reentry (Mao et al. 2018). The RLV reentry dynamics are highly nonlinear with limited control authority. In practice, path constraints such as heating rate, dynamic pressure and aerodynamic load must be strictly restricted **in order to guarantee the structural and thermal safety and reliability of the vehicle. Especially** with the diversification of reentry missions, the trajectories of the current RLV must sometimes meet the constraints of no-fly zones. Consequently, the RLV reentry trajectory optimization problem becomes **highly complicated**, and the gradient calculation becomes **onerous and tricky**. In recent years, nature-inspired algorithms (NIAs) have received widespread attention because they utilize the stochastic operators to **escape from local optima and converge to approximate** global optima in the iterative process. **Another popular reason is due to their generality and ease of use and tune in practice.** More commendable, they are not affected by the increased complexity (Gharehchopogh and Gholizadeh 2019). **Perhaps in view of these advantages, NIAs based on global optimization techniques have, for example, been applied to find orbital transfer trajectories** (Pontani and Conway 2010) **or path planning for solar-powered UAV in urban environment** (Wu et al. 2018).

NIAs can be broadly divided into three categories: evolutionary algorithms (EAs), swarm intelligence (SI) and physics-based (PB) algorithms (Jain et al. 2019). EAs mimic the evolutionary behavior of creatures found in nature. Genetic algorithm (GA) and differential evolution (DE) (Storn and Price 1997) algorithm can be considered as the common standard forms of EAs. The second category is SI based techniques, which imitate the intelligent social behavior of animal groups. Besides they collect and utilize the full information about search space with the progress of algorithm. Particle swarm optimization (PSO) (Eberhart and Kennedy 1995) and **artificial bee colony (ABC)** (Karaboga and Basturk 2007) can be regarded as the representative algorithms of SI. Several recent SI algorithms have also been proposed, such as grey wolf optimizer (GWO) (Mirjalili et al. 2014), moth-flame optimization (MFO) (Mirjalili 2015), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016) and squirrel search algorithm (SSA) (Jain et al. 2019). The PB algorithms are inspired from basic physical laws that exist in the universe, while the

prevailing methods of this category are simulated annealing (SA) (Kirkpatrick et al. 1983), multi-verse optimizer (MVO) (Mirjalili et al. 2016) and gravitational search algorithm (GSA) (Rashedi et al. 2009). Despite many prominent NIAs, there is no single optimization technique that can optimally solve all optimization problems, and be superior to all other optimization algorithms in solving the same optimization problem according to the "No Free Lunch" (NFL) theorem (Wolpert and Macready 1997). Therefore, researchers are still constantly developing more efficient optimization methods to achieve better designs. Consequently, various variants of the existing standard NIAs and hybrid algorithms with specific global and local search strategies have emerged in recent years. **These modified versions can tackle the limitation of the basic NIAs such as big dimensional problems, avoiding trapping in local optima, and some of them can reduce execution time (Ridha et al. 2020). Thus, they have been widely carried out to realize different real-world cases.** For instance, the hybridization of grasshopper and new cat swarm optimization algorithm for feature selection (Bansal et al. 2020), a hybridized version of the existing DE and symbiotic organism search algorithms for OPF problem (Saha et al. 2020), **the incorporation of whale and sine-cosine optimizers with cellular topology for different parameter optimization cases (Turgut et al. 2021), etc.**

Based on literature investigation of the mentioned works, one can feel that the modified versions of NIAs provide us a wider choice of optimization techniques, and their application in the field of spacecraft reentry trajectory optimization **is also dazzling. In the case of multi-objective trajectory optimization, there are many promising achievements.** For example, Kang et al. (2015) developed an improved ABC for multi-objective reentry trajectory optimization design of the RLV that could effectively and flexibly reflect the preference of the designers. **Chai et al. (2017b) designed an adaptive DE algorithm that could generate high quality Pareto front of the multi-objective trajectory optimization problem of space maneuver vehicles (SMV). In their follow-up work (Chai et al. 2018; Chai et al. 2020), they successively developed gradient-based hybrid genetic algorithm (GHGA) and novel NSGA-III-based optimal control solver to address multi-objective spacecraft trajectory optimization problems, and the efficiency and feasibility of these algorithms are particularly attractive and can offer an alternative for optimizing multi-objective trajectory optimization issues. For objective reentry trajectory optimization, which is the focus of this paper, there are likewise many notable achievements.** For instance, Zhang and Liu (2011) put forward an improved GA with sequential quadratic programming algorithm to achieve a fast and accurate optimum design of RLV reentry trajectory. Su and Wang (2015) proposed an improved GSA with gauss pseudospectral method (GPM) to address the RLV approach and landing trajectory optimization issue, which was proven practicable. **Chai et al. (2017a) presented an differential evolution algorithm incorporated with a violation learning constraint handling strategy based on hp-adaptive GPM to solve the space**

maneuver vehicle trajectory optimization issue, and the method was demonstrated feasible and effective. Zhang et al. (2020) hybridized an improved WOA with GPM to solve the hypersonic vehicle entry trajectory optimization problem with no-fly zones, which exhibited faster convergence, higher accuracy and strong robustness. A common feature of these reported algorithms is that they combine the robustness of NIAs with the high accuracy of gradient-based algorithms to form a hybrid two-stage optimization algorithm, thereby generating optimal trajectories with faster convergence speed and efficiency. Meanwhile this combination is a frequently used optimization technique in trajectory optimization, especially when the prior knowledge about the optimal trajectory cannot be predicted in advance. However, there still exists some weakness, such as which level of the accuracy of the first stage optimization results can provide the reliable guesses for the second stage optimization, thus ensuring the best efficiency of the entire optimization process. As a matter of fact, the two-stage optimization algorithm is a complex optimization problem in itself. The first stage of the optimization process is usually set to stop after reaching lower iterations, or when the tolerance set reaches the low-accuracy values in the order of 10^{-1} to 10^{-2} for normalized problems. Of course, from the available literature, it is also a trend to use a single NIA or its variants to address the trajectory optimization problems. The development of a novel stochastic gradient PSO for fast generation of feasible and smooth entry trajectories for hypersonic glide vehicle is a good example (Li et al. 2018), but the algorithm does not address the complex constraints of no-fly zones. Therefore, the point of departure is to develop a modified version of NIA with a few adjustable parameters that not only improves the optimization accuracy but is also quite robust to obtain a solution for RLV single objective reentry trajectory optimization with no-fly zones.

The RLV reentry trajectory optimization with complex constraints on path and no-fly zones is a highly nonlinear optimal control problem, and a priori knowledge of the optimal trajectory is usually unknown beforehand. As a result, the NIA based on global optimization techniques, which offers distinctive benefits, from the perspective of robustness, performance in the presence of uncertainty and unknown search spaces (Jain et al. 2019), is highly preferred. Beautifully yet imperfectly, stagnation at the early phases of the iterations, which leads to entrapment in local optimum points in the search space, is one of the inherent drawbacks of the NIAs (Turgut et al. 2021). Coupled with the fact that the RLV reentry trajectory optimization is a high-dimensional optimization problem that aims to search in a fixed space to obtain the optimal sequences of control variables, the shortcomings of premature convergence and stagnation in addressing this issue are more apparent in the standard version of NIAs. Motivated by these circumstances, this paper aims to develop a hybrid optimization algorithm based on a new NIA harris hawks optimization (HHO), namely, hybrid parallel harris hawks optimization (HPHHO) algorithm, thereby setting up a more stable trade-off between exploration and exploitation trends to improve the search efficiency for the

best solution. In view of an application, the proposed algorithm does not require a priori knowledge of the optimal solution and can automatically detect the regions in the solution. In addition, it can avoid the solution of a secondary optimization problem while ensuring the accuracy. All these reasons make it suitable for addressing the RLV reentry trajectory optimization problem.

The proposed HPHHO algorithm in this paper mainly uses three strategies: oppositional learning, smoothing technique, and parallel optimization mechanism. The oppositional learning in each iteration is used to intensify the diversity of the initial population. On the other hand, the smoothing technique is employed to smooth the solutions to reduce or avoid the fluctuations. In terms of the parallel optimization mechanism, it consists of HHO and DE two algorithms, whereas the core search procedure of the proposed algorithm is the harris hawks optimization (HHO). The HHO is one of the latest SI algorithms mimicking the surprise pounce behavior of harris hawks' chasing style (Heidari et al. 2019). The use of adaptive and time-varying escape energy parameters enables it to perform a smooth transition between exploration and exploitation, which can well address the difficulties of the search space, including local optimal solutions, multi-modality, and deceptive optima. Besides, this technique has been applied to tackle different real-world cases, such as manufacturing industry, environmental quality, image segmentation, power systems (Alabool et al. 2021), etc. Furthermore, in order to make the global exploration capability of HHO significantly strengthened, the logistic chaos map sequence, DE/current-to-best/1 operator and Levy mutation operator have been embedded in the standard version of HHO. When referring to the DE algorithm, it is regarded as a remarkable EA for global optimization over continuous search space, whose performance of simplicity, straightforwardness in implementation and good convergence are outstanding. Moreover, the DE/best/1 operator exhibits relatively faster convergence, which may provide an excellent exploitation ability (Sun et al. 2019). In practice, DE often works as a local search mechanism to improve the exploitation of hybrid algorithms, such as the combination of DE with GA (Mustafi and Sahoo 2019), sine-cosine algorithm (Nenavath and Jatoth 2018), and ABC (Jadon et al. 2017). Inspired by these observations, this paper regards DE as an exploiter of parallel optimization mechanism for promising regions on the search space to fine-tune the current optimal solution. Meanwhile, HHO acts as a global search mechanism to reduce the possibility of premature convergence and to better guide the search toward promising regions. In this way, the DE and HHO operate in parallel to search the global optimal solution more efficiently.

The contributions of this paper have three aspects:

- Propose an algorithm HPHHO by using oppositional learning, smoothing technique and parallel optimization mechanism strategies to improve the search efficiency for the best solution.
- Evaluate the performance of HPHHO on 4 benchmark problems in comparison with other NIAs, including PSO, GWO, MFO, ABC_{imp2}, MVO, WOA, WOASA, HHO and HHODE.

- Apply HPHHO to solve the RLV maximum cross-range problem with no-fly zones constraints.

The paper is outlined as follows: Section 2 presents and formulates the problem of RLV reentry trajectory optimization. The **fundamentals of HHO and DE algorithms are described** in Section 3. Section 4 **provides the detailed discussion of the proposed HPHHO algorithm**. Section 5 investigates the performance of HPHHO on four examples, and applies it to solve the RLV reentry optimization problem with no-fly zones constraints. Finally, the conclusions and future work are given in Section 6.

2 RLV reentry trajectory model

2.1 Reentry dynamics

The motion of RLV reentry problem is defined by the following set of equations (Betts 2009):

$$\begin{cases} \dot{r} = V \sin \gamma \\ \dot{\theta} = V \cos \gamma \sin \chi / (r \cos \phi) \\ \dot{\phi} = V \cos \gamma \cos \chi / r \\ \dot{V} = -D/m - g \sin \gamma \\ \dot{\chi} = L \cos \sigma / (mV) + \cos \gamma (V/r - g/V) \\ \dot{\sigma} = L \sin \sigma / (mV \cos \gamma) + V \cos \gamma \sin \chi \tan \phi / r \end{cases} \quad (1)$$

and the aerodynamic forces are computed according to:

$$L = \rho V^2 S C_L(\alpha) / 2, \quad D = \rho V^2 S C_D(\alpha) / 2 \quad (2)$$

where the states are geocentric radius r , the longitude θ , the latitude ϕ , the Earth-relative velocity V , the flight path angle γ , and the heading angle χ , while the controls are the bank angle σ and the angle of attack α . m is the mass of RLV; ρ is the atmospheric density; S is the reference area; C_L and C_D are aerodynamic lift and drag coefficients, respectively.

2.2 Constraints in the reentry process

2.2.1 Path constraints

The dynamic pressure q , heating rate \dot{Q} and overload n_z should be taken into account during the entire descent phase, and it can be written as follows:

$$\begin{cases} q = 1/2 \rho V^2 \leq q_{\max} \\ \dot{Q} = 17700(c_0 + c_1 \alpha + c_2 \alpha^2 + c_3 \alpha^3) \sqrt{\rho} (10^{-4} V)^{3.07} \leq \dot{Q}_{\max} \\ n_z = (L \cos \alpha + D \sin \alpha) / mg_0 \leq n_{z\max} \end{cases} \quad (3)$$

where c_0, c_1, c_2 and c_3 are fixed values (Graichen and Petit 2008). g_0 is the gravitational acceleration at sea level.

2.2.2 State and control constraints

To ensure the stability and controllability of RLV, it is necessary to limit the magnitude of the state $\mathbf{x} = [r, \theta, \phi, V, \gamma, \chi]^T$ and the control $\mathbf{u} = [\alpha, \sigma]^T$. The entire flight process can be expressed as:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}_{\text{rand}}(t) - r_1 |\mathbf{X}_{\text{rand}}(t) - 2r_2 \mathbf{X}(t)| & q \geq 0.5 \\ (\mathbf{X}_{\text{rabbit}}(t) - \mathbf{X}_m(t)) - r_3 (\mathbf{LB} + r_4 (\mathbf{UB} - \mathbf{LB})) & q < 0.5 \end{cases} \quad (7)$$

$$\mathbf{X}_m(t) = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i(t) \quad (8)$$

where $\mathbf{X}(t+1)$ is the position of hawks in $(t+1)$ th iteration, $\mathbf{X}_{\text{rabbit}}$ represents the position of rabbit or prey, r_1, r_2, r_3 , and r_4 are also random numbers inside $(0, 1)$ updated in each iteration. \mathbf{LB} and \mathbf{UB} are the lower and upper bounds of decision variables with the specific problem, respectively. The average position of hawks \mathbf{X}_m can be calculated using Eq. (8). N denotes the population size.

3.1.2 Transition from exploration to exploitation

An important control parameter to measure the transition between exploration and exploitation of HHO is the escaping energy of the prey E , which decreases with the increasing iteration, $Iter$, defined as follows:

$$E = 2E_0 \left(1 - \frac{Iter}{T}\right) \quad (9)$$

where E_0 denotes the initial energy of prey changing in the interval $[-1, 1]$, and T is the maximum number of iterations. If $|E| \geq 1$, the HHO performs the exploration phase to search the prey, while if $|E| < 1$, the exploitation phase is performed to exploit the promising areas.

3.1.3 Exploitation stage

Once the prey (rabbit) is detected, the harris hawks prefer to attack it with surprise pounce. In reality, different chasing styles occur with the various escaping behaviors of the prey. Four possible strategies are simplified to model the attacking situations (Heidari et al. 2019). A random parameter r is utilized to measure the chance of a prey in successfully escaping. The situation $r < 0.5$ performs successful escape, while $r \geq 0.5$ indicates unsuccessful escape. Whatever the prey does, the harris hawks will perform a hard or soft besiege to catch it depending on its retained energy E . If $|E| \geq 0.5$, the soft besiege occurs, while if $|E| < 0.5$, the hard besiege happens.

- Soft besiege

Assuming that $r \geq 0.5$ and $|E| \geq 0.5$, the rabbit still has enough energy to struggle for escaping. Finally, the rabbit is exhausted and fails to escape by the continuous softly encircling of the harris hawks. This behavior is formulated as:

$$\mathbf{X}(t+1) = \Delta \mathbf{X}(t) - E |J \mathbf{X}_{\text{rabbit}}(t) - \mathbf{X}(t)| \quad (10)$$

$$\Delta \mathbf{X}(t) = \mathbf{X}_{\text{rabbit}}(t) - \mathbf{X}(t) \quad (11)$$

where $\Delta \mathbf{X}(t)$ is the difference between the position vector of the rabbit and the current individual. $J = 2(1 - r_5)$ represents the random jump strength of the rabbit during the escaping process, and r_5 is a random number inside $(0, 1)$.

- Hard besiege

Assuming that $r \geq 0.5$ and $|E| < 0.5$, the rabbit is too exhausted to have enough energy for escaping. At this moment, the harris hawks hardly encircle the rabbit to perform the final surprise pounce. The procedure can be expressed as:

$$\mathbf{X}(t+1) = \mathbf{X}_{\text{rabbit}}(t) - E|\Delta\mathbf{X}(t)| \quad (12)$$

- Soft besiege with progressive rapid dives

Assuming that $r < 0.5$ and $|E| \geq 0.5$, the rabbit still has enough energy to escape successfully. Meanwhile, a more intelligent soft besiege is constructed before the surprise pounce. It must be noting that the updating position of the harris hawks is a two-step procedure with greedy selection strategy. The first step process can be mathematically written as:

$$\mathbf{Y} = \mathbf{X}_{\text{rabbit}}(t) - E|J\mathbf{X}_{\text{rabbit}}(t) - \mathbf{X}(t)| \quad (13)$$

and the second-step, Levy flight is used to perform the rapid, irregular, and abrupt movement of harris hawks when chasing the rabbit. The position is updated as follows:

$$\mathbf{Z} = \mathbf{Y} + \mathbf{S} \times \text{Levy}(D) \quad (14)$$

where D is the dimension of problem and \mathbf{S} is a random vector by size $1 \times D$ and Levy represents the Levy distribution which is calculated as follows:

$$\text{Levy}(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \right)^{\frac{1}{\beta}} \quad (15)$$

where u, v are random values inside $(0, 1)$, β is a constant value set to 1.5.

The final strategy for updating the positions in this procedure is given as follows:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{Y} & \text{if } F(\mathbf{Y}) < F(\mathbf{X}(t)) \\ \mathbf{Z} & \text{if } F(\mathbf{Z}) < F(\mathbf{X}(t)) \end{cases} \quad (16)$$

- Hard besiege with progressive rapid dives

Assuming that $r < 0.5$ and $|E| < 0.5$, the rabbit has not enough energy but succeeds in escaping. At this moment, the hawks construct a hard besiege before the surprise pounce, and they try to bring themselves closer to the prey. Hence, the final strategy for updating the positions is performed as:

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{Y} & \text{if } F(\mathbf{Y}) < F(\mathbf{X}(t)) \\ \mathbf{Z} & \text{if } F(\mathbf{Z}) < F(\mathbf{X}(t)) \end{cases} \quad (17)$$

$$\mathbf{Y} = \mathbf{X}_{\text{rabbit}}(t) - E|J\mathbf{X}_{\text{rabbit}}(t) - \mathbf{X}_m(t)| \quad (18)$$

$$\mathbf{Z} = \mathbf{Y} + \mathbf{S} \times \text{Levy}(D) \quad (19)$$

where \mathbf{Y} and \mathbf{Z} are obtained using the new rules in Eqs. (18) and (19).

3.2 Differential evolution algorithm

DE is a stochastic search algorithm based on population, which simulates the natural evolutionary process via mutation, crossover and selection to move its population toward the global optimum (Sun et al. 2019). **The DE algorithm mainly includes the following three operations.**

3.2.1 Mutation operation

Mutation operation in DE is performed to create three indices randomly in range over $[1, N]$, where N is the population size, and then three solution vectors X_{r1} , X_{r2} , and X_{r3} with a given index will be selected to generate a new solution V_i from the current solutions in the search space. The two most frequently used mutation operators implemented in DE variants are as follows:

- DE/best/1

$$V_i = X_{\text{best}} + F(X_{r1} - X_{r2}) \quad (20)$$

- DE/current-to-best/1

$$V_i = X_i + F(X_{\text{best}} - X_{r1}) + F(X_{r2} - X_{r3}) \quad (21)$$

where X_{best} is the best current solution vector, F is the mutation scaling factor inside $[0, 1]$. In the present work, DE/best/1 is adopted as the mutation operator to accelerate the convergence speed and maintain population diversity.

3.2.2 Crossover operation

Crossover operation in DE is performed to produce a trial vector U_i from the corresponding mutant solution vector V_i and the target vector X_i according to:

$$U_{ij} = \begin{cases} V_{ij} & \text{if } \text{rand}(0,1) \leq CR \text{ or } j = j_{\text{rand}} \\ X_{ij} & \text{otherwise} \end{cases} \quad (22)$$

where CR is the crossover rate, and j_{rand} is random value over $[1, D]$ which represents the randomly chosen index in the solution vector. D is the dimensionality of the problem.

3.2.3 Selection operation

Selection operation in DE is performed to make a choice between X_i or U_i based on their fitness value $f(\cdot)$, by selecting the fitter one for next iteration according to:

$$X_i(t+1) = \begin{cases} U_i(t) & \text{if } f(U_i(t)) \leq f(X_i(t)) \\ X_i(t) & \text{otherwise} \end{cases} \quad (23)$$

4 A hybrid parallel harris hawks optimization algorithm

A hybrid parallel harris hawks optimization HPHHO algorithm mainly consists of HHO and DE two algorithms. It is worth noting that DE/current-to-best/1 operator and Levy mutation operator as well as the logistic chaos map sequence are incorporated in the standard version of HHO, whereas DE adopts the DE/best/1 operator. The realization of HPHHO can be broadly divided into three steps. Firstly, the opposite population is calculated from the current population by the oppositional

learning strategy, and the fitness function values of initial and opposite population are calculated, as well as the N good harris hawks are selected as the new initial population according to the fitness function values. Secondly, the new initial population is divided into two equal subpopulations, which are assigned to DE and HHO, respectively. The both algorithms operate in parallel to search and update the solutions of each subpopulation simultaneously. Finally, the solutions are smoothed by the smoothing technique and the optimal solution can be obtained from the parallel optimization mechanism. The specific strategy and related procedures are as follows.

4.1 Oppositional learning

The opposition-based learning (OBL) is intended to generate the 'opposite' local of solutions in the initialization phase to cover a large area in the feasible domain, thereby improving the diversity of the initial solutions. Even without a priori knowledge, using OBL can still obtain fitter starting candidates and enhance the probability of detecting better regions (Liang et al. 2019). Many NIAs (such as GA, PSO, GSA, SA and krill herd algorithm) use OBL to improve their performance (Mahdavi et al. 2018). Inspired by these observations, the OBL idea is incorporated in the HPHHO initialization to improve diversity and search capabilities. For individual solution X_i , its opposition solution X_{ij}^o can be determined as follows:

$$X_{ij}^o = lb_j + ub_j - X_{ij} \quad (24)$$

where $j = 1, 2, \dots, D$ (the dimension of specific problems) and $i = 1, 2, \dots, N$ (the total number of hawks). Besides, the upper value ub_j and lower value lb_j are used in opposite solutions calculation.

4.2 Chaotic sequence

Another popular strategy to improve the convergence of HHO is to replace random parameters with chaotic map sequences (Abd and Mirjalili 2019). With the characteristics of ergodicity and non-repetitiveness, chaos can perform overall search at a higher speed than stochastic ergodic search that depends on probabilities (Coelho and Mariani 2008). Hence, logistic chaos map sequence is used to tune the random parameter r of HHO, which helps to control the exploitation mechanism. In addition, it can be defined as a discrete-time dynamic system as shown below:

$$lr(t+1) = \mu \cdot lr(t) \cdot (1 - lr(t)) \quad (25)$$

where t indicates the current iteration. Besides, the control parameter $\mu = 4$, $lr(1) \in (0, 1)$ and $lr(1) \neq 0.25, 0.5, 0.75, 1$. Unlike HHO, the r is taken from the chaotic sequence lr , in this way $r(t) = lr(t)$.

4.3 Smoothing technique

In order to reduce or avoid the control function fluctuations and corners in the optimal solution, smoothing technique (Hashemi and Mirjalili 2018) is embedded in the processing phase of HPHHO to smooth the solution. As depicted in Fig. 2, the white points are control values before smoothing while the black points are corresponding smoothed ones. It is worth noting that smoothing technique

is only performed when zigzag shapes in raw data occurs such as a zigzag between u_{k-1} , u_k and u_{k+1} . The average value based on each of three successive points is employed to reduce the magnitude of jumps. λ is a weight coefficient and the formula for smoothing at point k can be expressed as:

$$\bar{u}_k = \begin{cases} \lambda u_{k-1} + (1-\lambda)u_{k+1}, & \text{if } \text{sign}(u_k - u_{k-1})\text{sign}(u_{k+1} - u_k) < 0 \\ u_k, & \text{else} \end{cases} \quad (26)$$

where \bar{u}_k is the smoothed value of u_k and replaced it during the next steps of iterations.

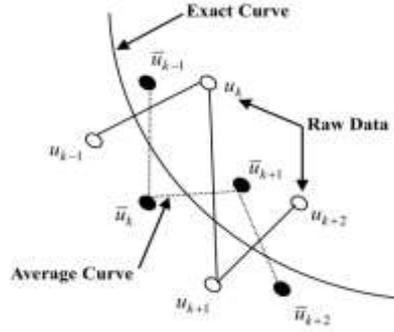


Fig. 2 Application of smoothing technique in a control variable

4.4 Mutation operators

To utilize the best current individual population information, accelerate the search speed, and reduce the risk of trapping in local optima. The famous DE/current-to-best/1 mutation operator is adopted to replace the original position update strategy in Eq. (7). The formulation can be rewritten as:

$$X_i(t+1) = \begin{cases} X_i(t) + F(X_{\text{rabbit}}(t) - X_{r_1}(t)) + F(X_{r_2}(t) - X_{r_3}(t)) & q \geq 0.5 \\ (X_{\text{rabbit}}(t) - X_m(t)) - r_4(LB + r_5(UB - LB)) & q < 0.5 \end{cases} \quad (27)$$

where r_4 and r_5 are also random numbers inside (0, 1).

Once the global search stage has passed, the intensity of exploitation should be strengthened in a promising search space, which benefits to convergence to the global optimum. Levy mutation operator is likely to generate a different offspring due to its heavy-tailed distribution, and can help the individuals escape from local optima easily. Therefore, local Levy mutation is utilized instead of global Levy flight. The Eqs. (14) and (19) can be rewritten as:

$$Z = Y + Y \times \text{Levy}(D) \quad (28)$$

4.5 Pseudo-code and flowchart of the HPHHO algorithm

For clear understanding, the specific pseudo-code of the HPHHO algorithm is depicted in Fig. 3, and the flowchart in Fig. 4 shows the procedures of HPHHO, including OBL strategy, logistic chaos map sequence, smoothing technique and mutation operators. **As can be seen that the HHO works as an exploration-based algorithm aims to search in far reached neighborhoods for finding global optimum, while the DE works as an exploitation-based algorithm focuses the search on already identified potential neighborhood for converging to optimum solution. Therefore, the HPHHO can**

maintain a more stable balance between exploration and exploitation trends.

4.6 Computational complexity

Note that the computational complexity of HHO depends on the number of harris hawks (N), the dimensions (D), and the maximum number of iterations (T). The time complexity of the basic HHO is $O(N \times (1 + T + T \times D))$. Correspondingly, there are seven aspects being mainly responsible for the computational complexity of the HPHHO algorithm, which are initialization, OBL strategy, chaotic sequences, fitness evaluation, smoothing technique, parallel optimization mechanism, and mutation operators. Since the fitness evaluation is related to specific problems to be solved and the mutation operators in HPHHO are only a replacement of the original operators, the calculation complexity of them can be neglected. The time complexity of initialization, OBL and chaotic sequences require $O(N)$, $O(N \times T)$ and $O(T)$, respectively. The smoothing technique requires $O(N \times T \times (D - 2))$. The time complexity of searching the best location and updating of individuals of parallel optimization mechanism are $O(T \times 2N \log 2N)$ and $O(N \times T \times D/2 + N \times T \times D/2)$. Therefore, the total time complexity of HPHHO is $O(N \times T \times (2 \log 2N + 2D - 1) + N + T)$, which is generally a little higher than the basic HHO.

4.7 Constraint handling

The HPHHO was originally proposed to solve the static optimization problem without constraints. Thus, the penalty function method working as a kind of constraint handling techniques was utilized to address the constrained continuous optimal control problem. The equidistant discrete time nodes within user-defined upper (ub) and lower (lb) bounds, are used to parameterize the control variables, which means $X \in [lb, ub]$. If the lower or upper limit is violated, the repair rules will serve as:

$$\begin{cases} X_{ij} = lb_j, & \text{if } X_{ij} < lb_j \\ X_{ij} = ub_j, & \text{if } X_{ij} > ub_j \end{cases} \quad (29)$$

The time history of state variables can be calculated by integrating differential equations with the fourth-order Runge–Kutta method. Decision variables are considered to be inequalities ($g_i(\mathbf{X}) \leq 0$). The dimensions and magnitudes of the objective function $f(\mathbf{X})$ and constraint functions $g_i(\mathbf{X})$ may be very different, which will make the optimization process complicated. Thus, the modified fitness is normalized using positive parameter m_i , which can be rewritten as:

$$\text{Fitness}(\mathbf{X}) = \begin{cases} m_0 \cdot f(\mathbf{X}), & \text{if } g_i(\mathbf{X}) \leq 0 \\ m_0 \cdot f(\mathbf{X}) + q \cdot \sum_1^n m_i \cdot g_i(\mathbf{X}), & \text{if } g_i(\mathbf{X}) > 0 \end{cases} \quad (30)$$

where n is the number of constraints $g_i(\mathbf{X})$ that are not satisfied, and q is a positive constant.

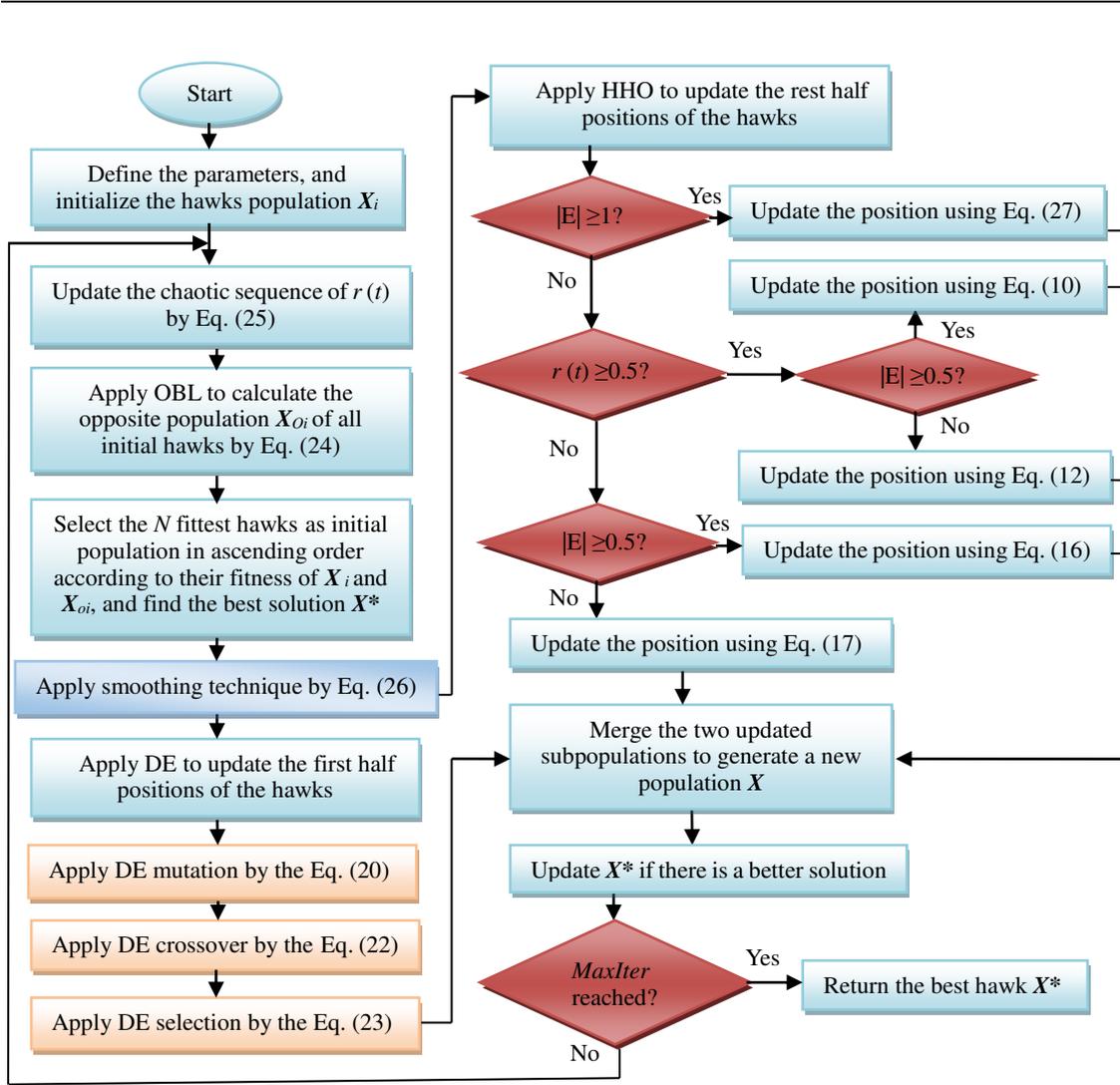


Fig. 4 The flowchart of HPHHO

5 Numerical examples

In order to evaluate and investigate the performance and effectiveness of the proposed algorithm, the HPHHO is compared with some well-known algorithms including PSO (He and Wang 2007), GWO (Mirjalili et al. 2014), MFO (Mirjalili 2015), ABC_{imp2} (Mernik et al. 2015), MVO (Mirjalili et al. 2016), WOA (Mirjalili and Lewis 2016), WOASA (Mafarja and Mirjalili 2017), HHO (Heidari et al. 2019) and HHODE (Bao et al. 2019). Besides, four HHO variants are participated in the comparative experiment. All these algorithms have been performed on MATLAB 2018a on a 3.30 GHz Intel Core i5 CPU with 8GB RAM, and the related parameters are in Table 1.

For three constrained continuous optimal control problems (examples 5.1, 5.3 and 5.4), where the population size $N_{pop}=300$, and the number of collocation points $N=20$ for all algorithms. In addition, the maximum number of iterations $MaxIter=200$. The results are recorded over 20 independent runs for the three examples. The optimization process of NIAs is set to stop when the change in fitness value between 10 iterations e_{stop} is less than $1e-6$. There are two definitions as follows: 1) Number of successful runs (SR): The number of successful runs for which the algorithm reaches the global

minimum (the relative error with the optimal solution is less than 5%). 2) Successful runs rates (*SRR*): *SRR* is equal to *SR*/*20*. One needs special attention is that the parameters of example 5.2 on four benchmark functions are different, while $N_{pop}=30$, $MaxIter=500$ and all algorithms are executed 30 independent runs.

Table 1 Parameters setting of the used algorithms

Algorithm	Parameter setting	Year
PSO	Learning factors $c_1=c_2=2$, Inertia factor $\omega_{max}=0.9$, $\omega_{min}=0.4$, Maximum velocity=0.2	2007
GWO	$a= [2,0]$	2014
MFO	Convergence constant $r= [-1, -2]$, $b=1$	2015
ABC _{imp2}	Trial limit parameter=20	2015
MVO	Wormhole existence probability $WEP_{min}=0.2$, $WEP_{max}=1$; $p=6$	2016
WOA	$a= [2,0]$, $b=1$	2016
WOASA	$a= [2,0]$, $b=1$; Initial temperature $T_0=0.1$, Cooling schedule is $0.99T$	2017
HHO	Constant $\beta=1.5$; Random jump strength $J \in [0,2]$	2019
HHODE	Constant $\beta=1.5$; Random jump strength $J \in [0,2]$; Mutation scaling factor $F=0.5$; Crossover rate $CR=0.9$	2019
HPHHO	Mutation scaling factor $F=0.5$; Crossover rate $CR=0.9$; Weight coefficient $\lambda=0.5$; Constant $\beta=1.5$; Random jump strength $J \in [0,2]$	—

5.1 Multiple solution example

This is an optimal control problem with one state and one control, $x(t), u(t) \in \mathbb{R}$, as well as two distinct local minima (Floudas et al. 1999). The objective function and constraints can be expressed as follows:

$$\begin{cases} \min & J = -x^2(t_f) \\ \text{subject to} & \dot{x} = -x^2 + u \\ & -12 \leq x(t) \leq 9; \quad -5 \leq u(t) \leq 5 \\ & x(0) = 9, \quad t_f = 1 \end{cases} \quad (31)$$

The exact objective function solution for the global minimum of the problem is -8.23623 with control variable $u = -5$. Note that there also exists another local minimum with control variable $u = 5$.

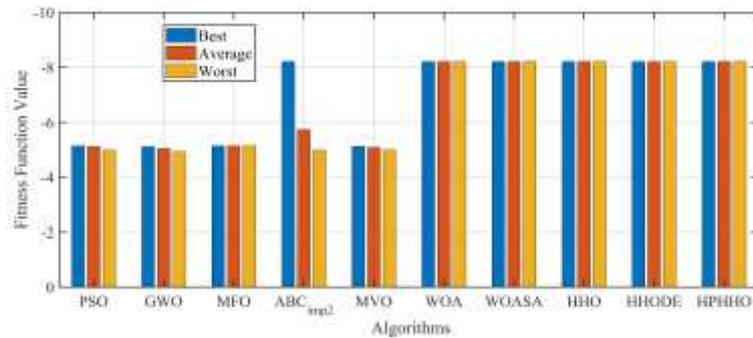


Fig. 5 The experimental results of different algorithms on multiple solution example

As shown in Fig. 5, the PSO, GWO, MFO, and MVO fail to find the global optimal solution in

this optimal control problem, **whereas the ABC_{imp2} is not available on all successful runs.** Thus, they have worse local optima avoidance. **Apart from these algorithms,** the WOA, WOASA, HHO, HHODE and HPHHO have **managed to find** the global optimum with higher accuracy. Therefore, we would not discuss the algorithms of PSO, GWO, MFO, ABC_{imp2} and MVO as comparison in the following sections.

5.2 Performance on benchmark functions

The **performance of HPHHO** is evaluated **and investigated** on 4 mathematical test functions with different characteristics, such as unimodal, multi-modal and composite. The standard benchmark functions are taken from the IEEE CEC 2005 set, which are tabulated in Table 2. In the average value (AVG) and standard deviation (STD), the comparative results of the HHO variants as well as the 4 optimizers are obtained to evaluate the performance, of which the results shown in **bold** are the best. In order to further investigate the statistically significant differences between the HPHHO and its competitors, a nonparametric Wilcoxon sign rank test at 5% significance level (Derrac et al. 2011) is used herein. The symbols of "+", "=", and "-" indicate that the HPHHO is superior, equal to and inferior to its counterparts, respectively. Moreover, the Friedman test is also employed to rank the average performance of all competitors, and lists the average ranking value (ARV) in comparison results.

Table 2 Description of four benchmark functions

Function	Dim	Range	f_{\min}
$F_1(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	[-100,100]	0
$F_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]$	30	[-30,30]	0
$F_3(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	[-32,32]	0
$F_4(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532

In order to deeply investigate the influences of OBL, logistic chaos map sequence, two mutation operators and population distribution order on the performance of the algorithm. **This section tests** the 4 HHO variants **based on 4 benchmark functions, namely** OPHHODE, PHHCDE, PHHLDE and OPHHCLDE. The first three variants imply combining only with OBL, logistic chaos map sequence and two mutation operators, respectively. For OPHHCLDE, it incorporates all the three strategies. Notice that OPHHCLDE means that the top half of the initial population **is allocated to** HHO and the latter half of the initial population **is allocated to** DE, while the distribution order of population is opposite to the proposed HPHHO algorithm.

5.2.1 Performance metrics

The comparative results of HHO variants as well as 4 optimizers on 4 functions are presented in

Table 3 and Table 4. It can be seen that the HPHHO outperforms other competitors in almost all cases in terms of AVG and STD values, **indicating its higher accuracy and robustness**. In addition, when it refers to the average rankings obtained by Friedman test, the proposed algorithm HPHHO ranks first in all cases with the lowest ARV. From a statistical point of view, it has achieved the best performance **among all these competitors**. According to the Wilcoxon's sign rank test and the meaning of the symbol of "+/=-", the HPHHO is superior to PHHCDE, PHHLDE, WOASA, HHO and HHODE on 3 out of 4 cases, and equal to them on 1 out of 4 ones. Moreover, the results of HPHHO performs significantly better than those obtained by WOA in tackling all the functions. Furthermore, the OPHHODE and OPHHCLDE have shown relatively good performance due to the inclusion of the OBL strategy. Both are equal and inferior to HPHHO on 2 out of 4 cases, respectively. Given that the program responsible for exploration and exploitation in OPHHCLDE is exactly the opposite of HPHHO, and the ARVs of the two algorithms are 2.38 and 2.25 respectively, it is clear to see that the performance difference between them is not particularly **obvious**. The evaluation of these two algorithms will be discussed further in section 5.3.

Table 3 Comparison of the results of HHO variants

F	Metric	OPHHODE	PHHCDE	PHHLDE	OPHHCLDE	HPHHO
F1	AVG	2.3491e-313	1.6712e-55	9.9909e-55	2.4752e-125	0.00e+00
	STD	0.00e+00	9.1503e-55	5.4462e-54	1.3556e-124	0.00e+00
F2	AVG	0.086026	9.4422	1.8788	0.008728	0.00019157
	STD	0.1468	11.9705	6.6536	0.017189	0.00038651
F3	AVG	8.8818e-16	8.8818e-16	8.8818e-16	8.8818e-16	8.8818e-16
	STD	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
F4	AVG	-10.1532	-5.5612	-5.994	-10.1532	-10.1532
	STD	0.0026908	1.5568	2.1606	1.2411e-09	3.6193e-07
	+/=-	2/2/0	3/1/0	3/1/0	2/2/0	~
	ARV	2.88	3.88	3.63	2.38	2.25
Overall Rank		3	5	4	2	1

Table 4 Comparison of the results with 4 optimizers

F	Metric	WOA	WOASA	HHO	HHODE	HPHHO
F1	AVG	46786.2771	42394.3924	1.598e-78	3.5891e-54	0.00e+00
	STD	12954.6013	12401.4274	6.2359e-78	1.9658e-53	0.00e+00
F2	AVG	27.8885	1.4731	0.011505	5.3078	0.00019157
	STD	0.47783	1.4179	0.021909	9.4002	0.00038651
F3	AVG	4.4409e-15	4.204e-15	8.8818e-16	8.8818e-16	8.8818e-16
	STD	2.9504e-15	2.6279e-15	0.00e+00	0.00e+00	0.00e+00
F4	AVG	-7.5963	-10.1532	-5.2089	-5.2173	-10.1532
	STD	2.8818	1.0266e-06	0.86207	0.93247	3.6193e-07
	+/=-	4/0/0	3/1/0	3/1/0	3/1/0	~
	ARV	4.38	3.44	2.50	3.13	1.56
Overall Rank		5	4	2	3	1

5.2.2 Convergence rate analysis

This section mainly evaluates the convergence property of HPHHO versus the 4 well-developed optimizers. The convergence curves of the WOA, WOASA, HHO, HHODE and HPHHO on all the functions are plotted in Fig. 6 to observe the convergence rate of the algorithms. The WOA and WOASA are prone to premature convergence on F1, whereas the WOA, HHO and HHODE have trapped into local optima on F4. None of them can obtain a good solution on all cases. As expected, the HPHHO shows a fast convergence speed from the initial steps of iterations and a strong capability to obtain a high-quality solution and this behavior is evident on all functions.

As a summary, the comparative results reveal the different features of the proposed HPHHO algorithm. The OBL strategy can enhance the diversity of the population, thereby ensuring a better solution in the initial stage of the iteration. Besides HHO, which is integrated mutation operators, can jump out of the local optima and search in far reached neighborhoods for finding global optimum, while greatly reducing the likelihood of stagnation. Meanwhile, the DE is committed to enhancing the exploitation by searching the most promising regions identified by HHO. Both DE and HHO in the parallel optimization mechanism operate in coordination with each other and can quickly converge to the optimal solution. However, WOA and HHO do not have additional special operators to devote iterations to searches in the promising regions. As a result, they are not completely free from the problems of premature convergence and stagnation. If most individuals fall into local optima, especially if the optimal solution also falls into it, the whole algorithm will stagnate, and these conditions are where the algorithm will diverge.

5.2.3 Stability analysis

This section analyzes and discusses the stability, including two performance indicators, namely STD value and boxplot. STD represents the degree of dispersion of a dataset. The smaller the STD, the more stable the algorithm is. The competitive results of HPHHO and other four algorithms run independently for 30 times are shown in Table 4, from which can be found that HPHHO has the lowest STD value among all functions over its counterparts, showing remarkable stability. Since the 4 tested functions present unimodal, multi-modal and composite characteristics, respectively, the boxplots of each algorithm and function are shown in Figure 7 for a better understanding of the distribution of the results. It can be noticed that HHO and HHODE present better performance on F1, F2 and F3. Once referring to composite F4, they are not so stable enough to obtain the global optimum. Moreover, WOA and WOASA are not available on all sets. The distributions of HPHHO results are all concentrated on the global optimum. Thus, the boxplots vividly reveal the stability and better performance of the proposed algorithm compared with other competitors.

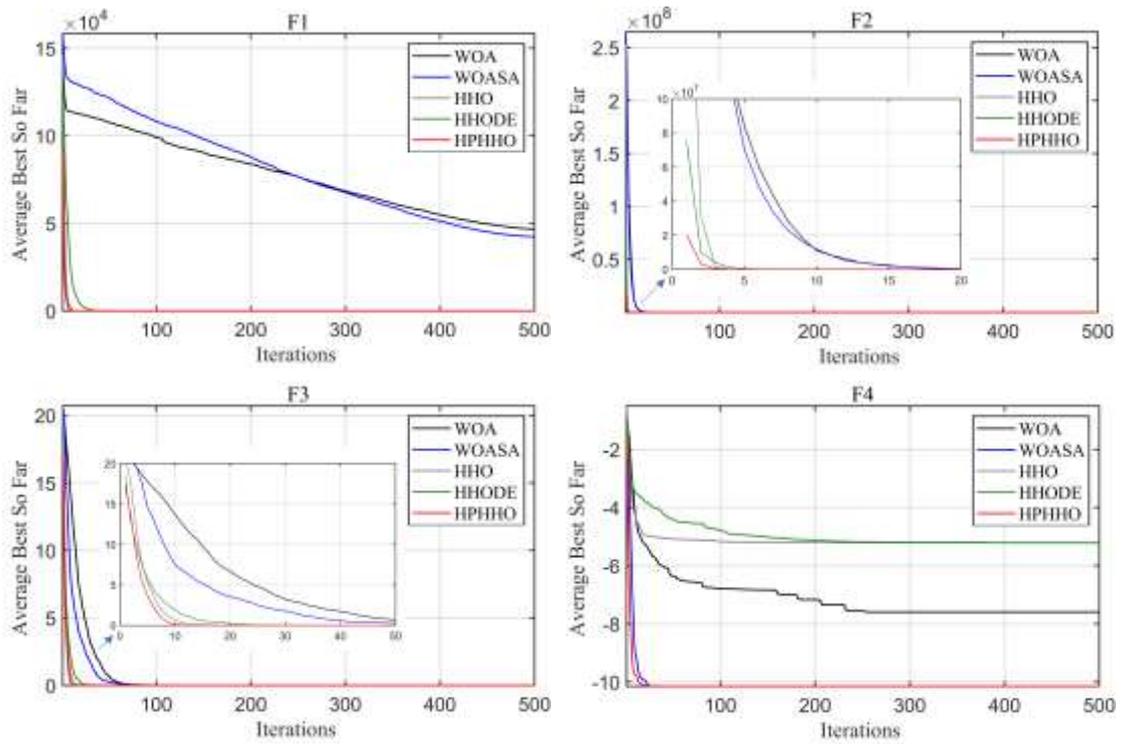


Fig. 6 Convergence rate of HPHHO versus other optimizers

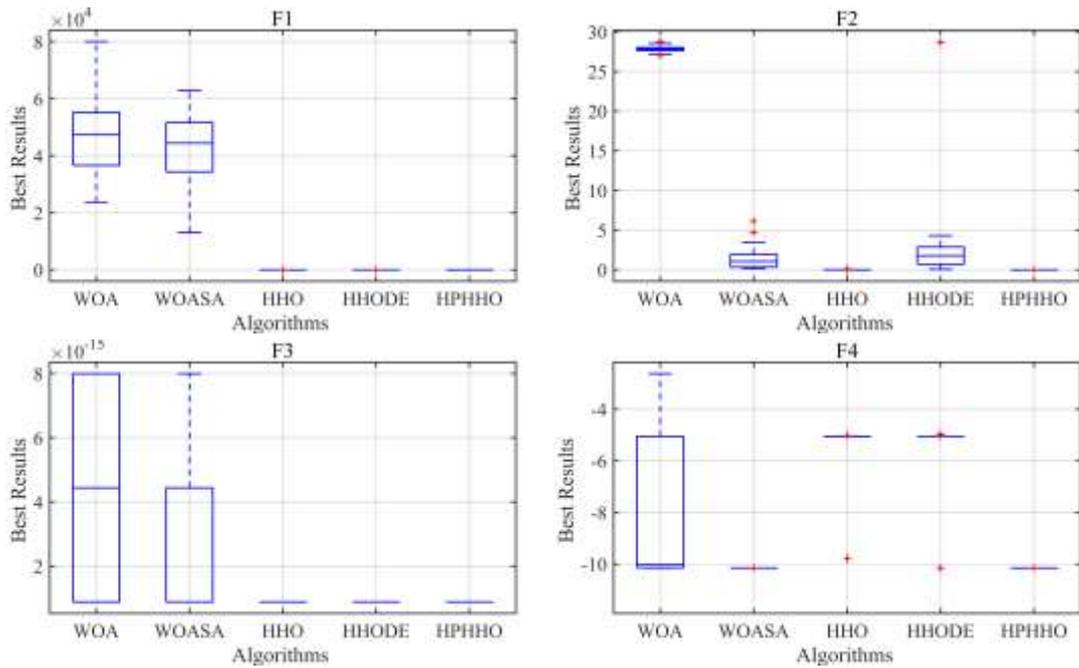


Fig. 7 The boxplot of each method on four benchmark functions

5.3 Minimum energy

The influence of smoothing technique is studied in this section. Consider the minimum energy problem of linear dynamical systems with analytical solutions (Hashemi and Mirjalili 2018). The objective function and constraints can be expressed as follows:

$$\left\{ \begin{array}{l} \min \quad J = \frac{1}{2} \int_0^{t_f} u^2(t) dt \\ \text{subject to} \quad \dot{x}_1(t) = x_2(t) \\ \quad \quad \quad \dot{x}_2(t) = -x_2 + u(t) \\ \quad \quad \quad x_1(0) = 1, x_2(0) = 1, x_1(t_f) = 0, x_2(t_f) = 0 \\ \quad \quad \quad |u| \leq 18, t_f = 1 \end{array} \right. \quad (32)$$

while the analytical solution can be defined as:

$$x_1(t) = c_1 e^t + c_2 e^{-t} + c_3 + c_4 t, \quad x_2(t) = c_1 e^t - c_2 e^{-t} + c_4 \quad (33)$$

where $c_1 = \frac{3-2e}{e^2-4e+3}$, $c_2 = \frac{e^2}{e^2-4e+3}$, $c_3 = \frac{-2e}{e^2-4e+3}$, $c_4 = \frac{2e}{e-3}$.

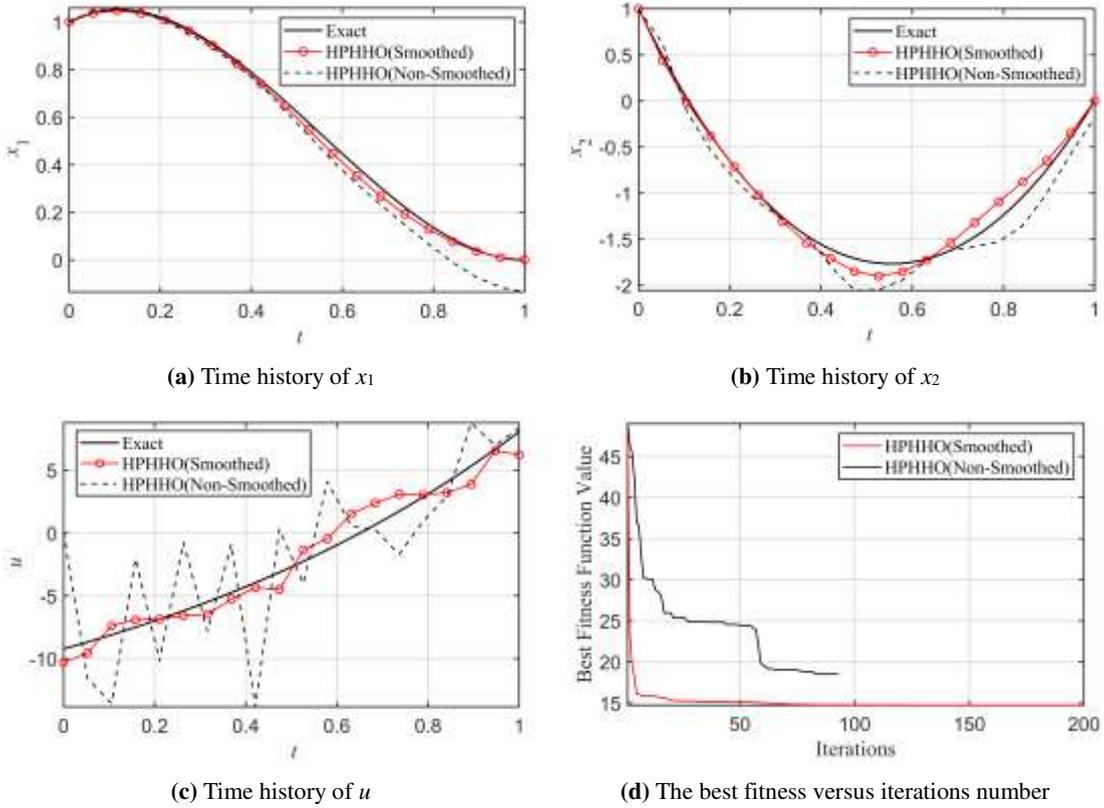


Fig. 8 The exact and numerical solutions of the minimum energy solution

The results of the system responses are described in Figs. 8 (a)-(c). Obviously, compared with the non-smoothed solutions without the smoothing technique, the HPHHO (Smoothed) time history curves are more consistent with the exact solution. Through the depiction of the control variable time history curve in Fig. 8 (c), it can be found that the non-smoothed solution fluctuates greatly and is jagged, while the smoothed solution is smoother and closer to the actual control variable curve. In addition, the best fitness function value with respect to iterations is depicted in Fig. 8 (d), which represents the convergence rate. As expected, HPHHO requires fewer iterations than non-smoothed, reflecting the competitiveness of the smoothing technique for the algorithm in terms of time and computational resources. Obviously, the smoothing technique can significantly

improve the performance of the proposed algorithm in solving continuous optimal control issues.

5.4 RLV maximum cross-range with heating rate constraint

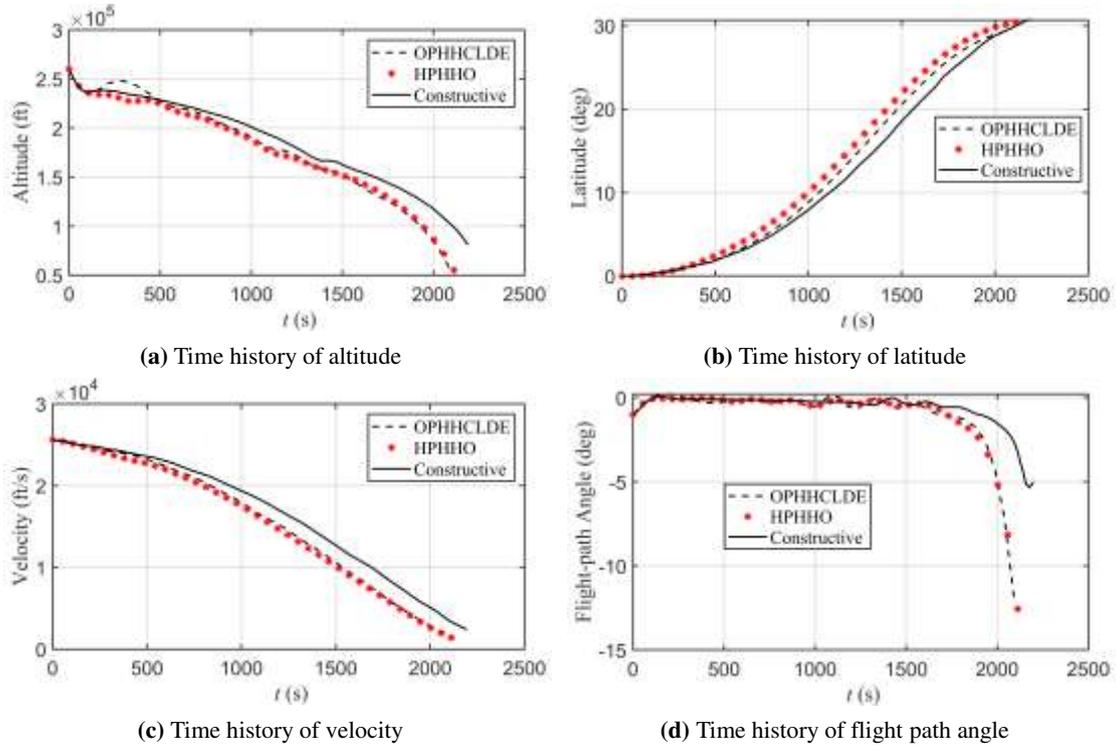
In this section, the HPHHO and the variant OPHHCLDE are used to solve the RLV maximum cross-range with heating rate constraint problem. The influence of the population distribution order is investigated. The objective function of the reentry problem is to maximize the cross-range, which is equivalent to maximizing the terminal latitude $\phi(T_{\text{end}})$. The specific values of initial and terminal conditions as well as the state constraints follow (Graichen and Petit 2008). In addition, the control constraints can be formulated as follows:

$$10^\circ \leq \alpha \leq 30^\circ, \quad -89^\circ \leq \sigma \leq 0^\circ, \quad 2000\text{s} \leq T_{\text{end}} \leq 2200\text{s} \quad (34)$$

There is another single path constraint satisfying $\dot{Q} \leq 70 \text{ Btu/ft}^2/\text{s}$, and the exact solution of $\phi(T)$ is 30.6255° . A comparison between the results obtained from OPHHCLDE and HPHHO is listed in Table 5. In addition, the best results of the two methods and a constructive method (Graichen and Petit 2008) are depicted in Fig. 9.

Table 5 Comparison for RLV maximum cross-range problem with heating rate constraint

Algorithms	SR	SRR	Average results of $\phi(T)$, $^\circ$	Best results of $\phi(T)$, $^\circ$	Worst results of $\phi(T)$, $^\circ$	$ \phi(T)_{\text{average}} - \phi(T)^* / \phi(T)^*$
OPHHCLDE	11	55%	27.86455	29.54669	23.82476	9.02%
HPHHO	20	100%	29.96602	30.40323	29.2593	2.15%



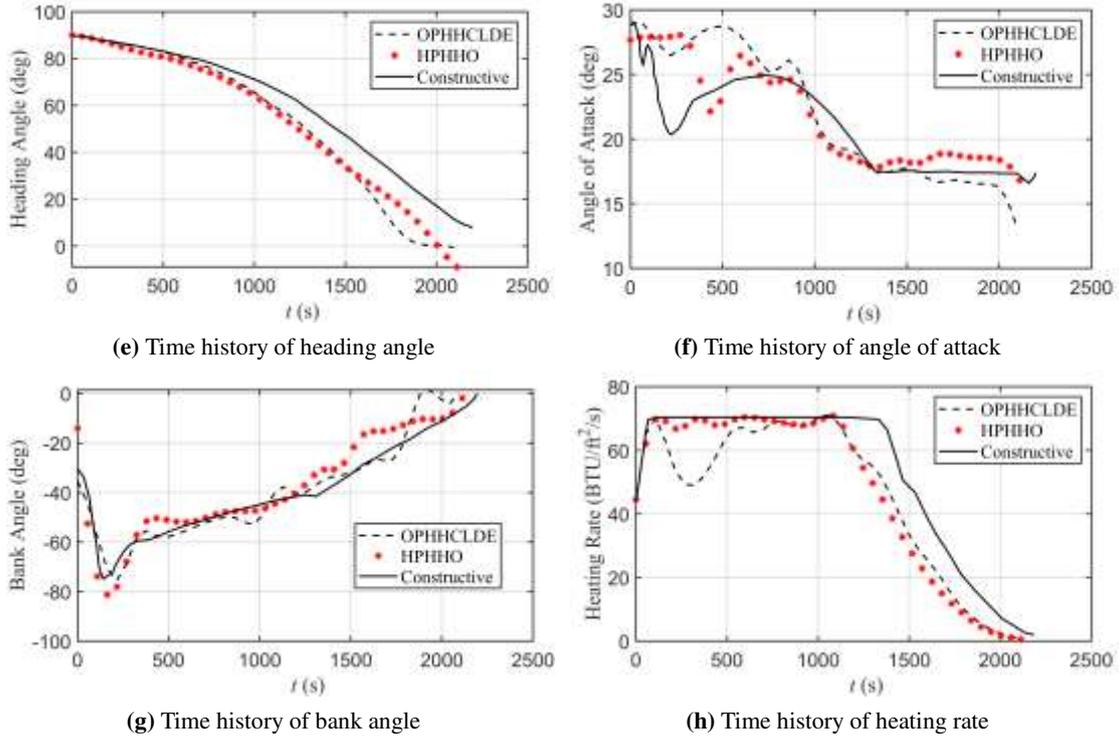


Fig. 9 Optimal trajectories for the constrained reentry problem with heating rate constraint

As shown in Table 5, the *SRR* of OPHHCLDE is 55%, which indicates that it has poor robustness and tends to fall into local optima. In addition, the quality of its solution is not good, with relative error 9.02%. Compared with OPHHCLDE, HPHHO has a relative error of 2.15% and an *SRR* of 100%. It has a higher accuracy and strong robustness.

Figure. 9 shows the best results obtained from OPHHCLDE and HPHHO, both are close to the global optimal solution obtained by the constructive method. Besides, it can be observed that the curves of states, controls and path constraints of HPHHO fit better. Obviously, in terms of the quality of the optimal results, HPHHO is superior to OPHHCLDE. In case of solving the same type of problems, HPHHO can find the best solution at one time regardless of the increasing scale and complexity of the problem, thus showing good effectiveness.

5.5 RLV maximum cross-range with different constraints

Based on the study of the above simulations, it can be found that HPHHO can solve RLV maximum cross-range with heating rate constraint problem with better robustness and effectiveness. To further evaluate the applicability and feasibility of HPHHO in dealing with the RLV maximum cross-range optimization problem, this section tests it by adding different path constraints, involving uncertainty analysis as well as no-fly zones constraints.

5.5.1 Performance of HPHHO in addressing different constraints

A comparative study with respect to the performance of HPHHO in solving the RLV maximum cross-range with different constraints is carried out in this section. Three mission cases are tested to demonstrate the capability and performance of HPHHO in handling different constraints, and

the three mission cases are summarized as follows.

- 1) Case 1: $[\mathcal{Q}_{\max}, q_{\max}, n_{z\max}] = [200 \text{ Btu/ft}^2/\text{s}, 13406 \text{ Pa}, 2.5 g_0]$.
- 2) Case 2: $[\mathcal{Q}_{\max}, q_{\max}, n_{z\max}] = [100 \text{ Btu/ft}^2/\text{s}, 12500 \text{ Pa}, 2.0 g_0]$.
- 3) Case 3: $[\mathcal{Q}_{\max}, q_{\max}, n_{z\max}] = [70 \text{ Btu/ft}^2/\text{s}, 11970 \text{ Pa}, 1.25 g_0]$.

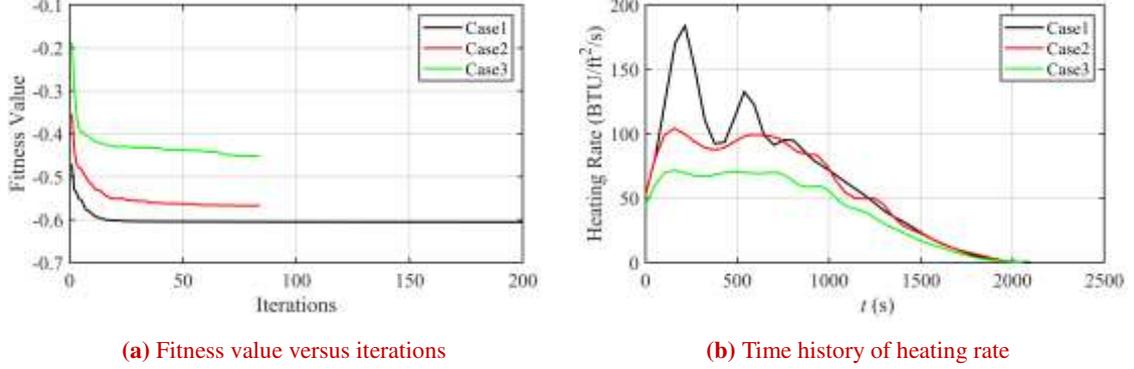


Fig. 10 Results of HPHHO for three different cases

The obtained fitness values versus iterations and the heating rate with respect to time curves are plotted in Fig.10, from which it can be observed that the HPHHO can generally obtain satisfactory solutions for different mission cases. The peak heating rates and the number of iterations stops amount to 184.1, 104.3, 71.7 Btu/ft²/s, and 200, 84, 84 for the three cases, respectively. It is worth noting that the heating rate constraint is active at only one collocation point in Case 2, while there are multiple heating rate constraints active in Case 3. In other words, when the constraint conditions become tighter, it is evident that the HPHHO might fail to drive all candidate solutions into the feasible region and the optimization convergence accuracy is somewhat limited. However, the reduced feasible domain space seems to have a positive influence on the population evolution and convergence process. Overall, the HPHHO tends to have more potential in dealing with the investigated problem with various mission constraints.

5.5.2 Impact of uncertain variables

This section analyzes the impact of different variables on the terminal latitude $\phi(T_{\text{end}})$. As stated in (Chai et al. 2020), the mass m of RLV may experience a fraction, especially when the RLV enters the atmosphere. Thus, an uncertain evaluation is performed in terms of the uncertain mass value. Take mission case 2 as an example and the mass value uncertainties are set as $\delta m = +5\%$ and $\delta m = +10\%$. The obtained uncertainty-perturbed fitness values are plotted in Fig.11(a), from which it can be observed that as δm increase, the fitness values become -0.58 and -0.57, which is deviated from the original solution -0.59. As an increase in RLV mass results in a decrease in velocity, the lower speed of the vehicle also has negative influences with respect to the latitudinal acceleration. The variation of latitude is lower than the one without considering the mass uncertainty, and this phenomenon can be reflected by the curve in Fig.11(a). Therefore, it is necessary to consider the mass uncertainty in the actual design of the spacecraft reentry trajectory.

Another uncertain evaluation is also performed to study the impact of the uncertain variable γ . Note that the flight path angle is one of the most critical parameters with respect to RLV trajectory optimization reentry conditions. The flight path angle value uncertainties are set as $\gamma = 0$ deg and $\gamma = -0.5$ deg. Fig.11(b) displays the obtained uncertainty-perturbed fitness values versus iterations, and it can be seen that as γ increase, the fitness values become -0.58 and -0.53, moving away from the original solution -0.59. The terminal latitude exhibits a high sensitivity to the flight path angle, whose uncertainty should be considered in the actual design of the spacecraft reentry trajectory.

It is clear from Fig.11 that the results obtained by HPHHO for the mass uncertainty perturbation can match well with the theoretical analysis compared with the normal results, which indirectly reflects the effectiveness of the proposed algorithm. In addition, the RLV flight reentry trajectory is highly sensitive to the flight path angle and the performance of the algorithm depends on its specific initial value. In general, HPHHO can handle these cases effectively.

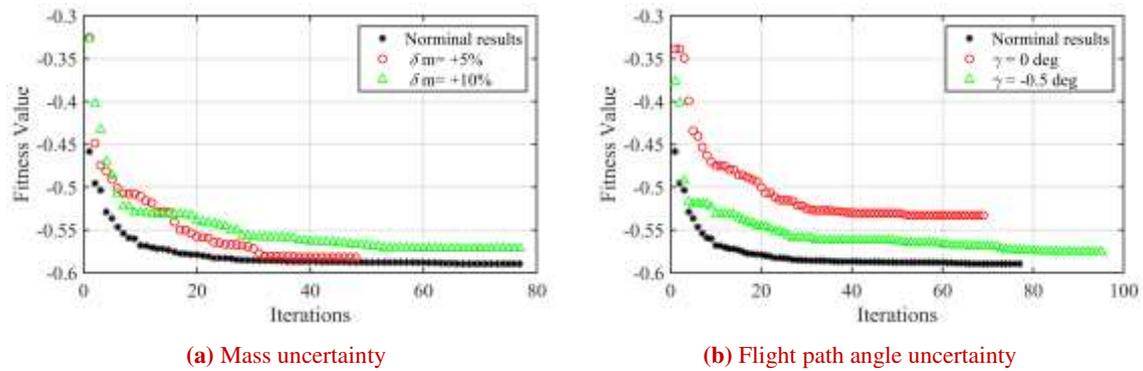


Fig. 11 Results of fitness value with uncertainty

5.5.3 Reentry trajectory optimization problem with no-fly constraints

In this simulation, the no-fly zones constraints are considered during the reentry phase (case 2). The result of the test is to determine how well the numerical solution obtained by the HPHHO avoids different kinds of constraints and obtains the objective solution, thus reflecting the applicability and feasibility of the proposed algorithm. No-fly zones parameter constraints can be found in Table 6. The open-source software GPOPS (Rao et al. 2010) based on GPM is used for comparative analysis.

Table 6 Parameters of no-fly zones constraints

Constraint	Parameter
No-fly zone 1	Center: $\theta_1 = 20^\circ$, $\phi_1 = 5^\circ$; Radius: $R_{Z1} = 445.3$ km
No-fly zone 2	Center: $\theta_2 = 70^\circ$, $\phi_2 = 15^\circ$; Radius: $R_{Z2} = 334.0$ km

As depicted in Fig. 12, the trajectories of GPOPS and HPHHO are very consistent, and all the path constraints are not violated, which proves the practicality and feasibility of HPHHO in solving this case. In particular, the trajectories with and without no-fly zones constraints shown in Fig. 12(d) indicate that the proposed algorithm is effective for this mission. Moreover, the heating rate plays a major role in the early stages of the reentry phase, while the dynamic pressure and overload gradually dominate in the subsequent gliding stage as the altitude decreases and the air density

increases. The two optimal trajectories and the location of 3D no-fly zones are displayed Fig. 13, from which it can be seen that both optimal trajectories smoothly avoid the no-fly zones.

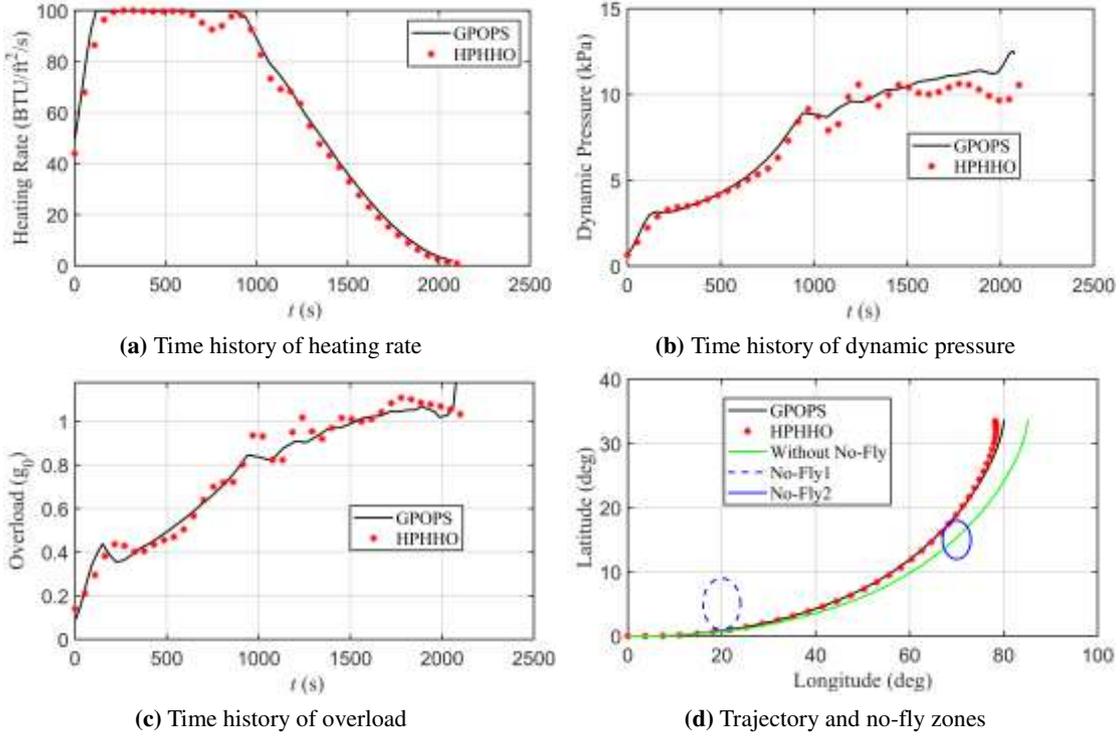


Fig. 12 Optimal path trajectories with no-fly zones constraints of HPHHO and GPOPS

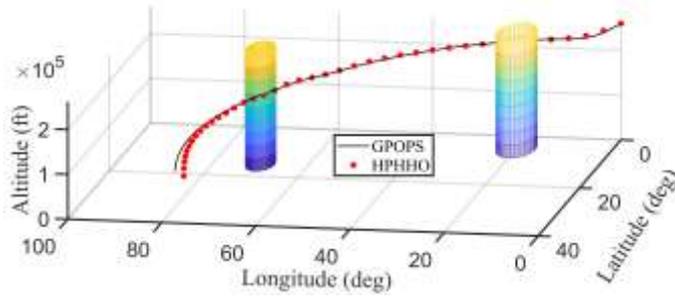


Fig. 13 Three-dimensional optimal trajectory

Given the lack of a priori knowledge of the optimal trajectory, this case uses linear interpolation of the initial and terminal conditions as the initial guesses of GPOPS, which achieves trajectory optimization at the number of collocation points $N=40$, and requires iterative training. Unsuitable guesses and parameter choices may lead to the use of a large amount of computational resources, and even some optimization results are difficult to satisfy the dynamic model and path constraints, such as the results of GPM optimization in (Zhang et al. 2020). The efficiency of GPOPS largely depends on the specific optimization problem. In contrast, the proposed algorithm HPHHO does not require a priori knowledge of the optimal solution and can automatically detect the region in the solution within the interval of user-defined control variables. In addition, it can avoid solving the quadratic optimization problem while guaranteeing satisfactory accuracy compared with the hybrid NIAs based and gradient-based algorithms.

While admiring its potential and superiority, one has to acknowledge the fact that the algorithm proposed in this paper is somehow based on a continuous iterative process of stochastic operators, and the processing time of the optimization procedure does not currently support the use of the algorithm for online optimal trajectory design. It is important to remark that for practical spacecraft guidance and control systems, the design of optimal flight trajectories is usually carried out offline (Chai et al. 2020). In fact, if parallel computing or high-performance computers can be implemented to optimize the flight trajectory, the optimization processing can be further reduced. In addition, the solution obtained using HPHHO can not only provide high-quality reference trajectories for online tracking algorithms, but also provide an alternative for constructing trajectory databases.

6 Conclusions

This work proposes a HPHHO algorithm for RLV reentry trajectory optimization problems. The main purpose of HPHHO is to set up a more stable trade-off between exploration and exploitation trends. To obtain a global optimization solution, three strategies are adopted including oppositional learning, smoothing technique and parallel optimization mechanism. Oppositional learning is used to intensify the diversity of the initial population, whereas the smoothing technique is employed to smooth the solutions of the redistributed initial population. In the strategy of parallel optimization mechanism, the DE algorithm enabled by DE/best/1 operator exhibits relatively faster convergence. Simultaneously, incorporating with the logistic chaos map sequence, DE/current-to-best/1 operator and Levy mutation operator, the HHO exhibits a super exploration capability. DE and HHO operate in parallel to search the global optimal solution more efficiently. Testing on 4 standard benchmark functions and compared with 4 well-established optimizers as well as 4 HHO variants, the HPHHO gets superior results in terms of average value, standard deviation, and convergence rate. Through 3 constrained continuous optimal control problems, the convergence and robustness of the HPHHO have significant advantages. Finally, the proposed algorithm is successfully applied to solve the critical problem for RLV trajectory reentry optimization with no-fly zones. And its effectiveness, applicability and feasibility are further demonstrated by simulation results. This research can be extended in many research areas, including multi-objective optimization problems, initial guess generator for gradient-based methods, and optimal control problems.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abd M, Mirjalili S (2019) A hyper-heuristic for improving the initial population of whale optimization algorithm. *Knowledge-Based Syst* 172:42–63. <https://doi.org/10.1016/j.knosys.2019.02.010>

-
- Alabool HM, Alarabiat D, Abualigah L et al (2021) Harris hawks optimization: a comprehensive review of recent variants and applications. *Neural Comput Appl*.
<https://doi.org/10.1007/s00521-021-05720-5>
- Bansal P, Kumar S, Pasrija S et al (2020) A hybrid grasshopper and new cat swarm optimization algorithm for feature selection and optimization of multi-layer perceptron. *Soft Comput* 24:15463–15489. <https://doi.org/10.1007/s00500-020-04877-w>
- Bao XL, Jia HM, Lang CB (2019) A novel hybrid harris hawks optimization for color image multilevel thresholding segmentation. *IEEE Access* 7:76529–76546.
<https://doi.org/10.1109/ACCESS.2019.2921545>
- Betts JT (2009) *Practical methods for optimal control and estimation using nonlinear programming*, 2nd ed. SIAM Press, Philadelphia
- Chai R, Savvaris A, Tsourdos A et al (2018) Unified multiobjective optimization scheme for aeroassisted vehicle trajectory planning. *J Guid Control Dyn* 41:1521–1530. <https://doi.org/10.2514/1.G003189>
- Chai R, Savvaris A, Tsourdos A et al (2020) Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm. *IEEE Trans Cybern* 50:1630–1643.
<https://doi.org/10.1109/TCYB.2018.2881190>
- Chai R, Savvaris A, Tsourdos A (2017a) Violation learning differential evolution-based hp-adaptive pseudospectral method for trajectory optimization of space maneuver vehicle. *IEEE Trans Aerosp Electron Syst* 53:2031–2044. <https://doi.org/10.1109/taes.2017.2680698>
- Chai R, Savvaris A, Tsourdos A et al (2017b) Multi-objective trajectory optimization of space manoeuvre vehicle using adaptive differential evolution and modified game theory. *Acta Astronaut* 136:273–280. <https://doi.org/10.1016/j.actaastro.2017.02.023>
- Coelho L dos S, Mariani VC (2008) Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Syst Appl* 34:1905–1913.
<https://doi.org/10.1016/j.eswa.2007.02.002>
- Derrac J, García S, Molina D et al (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1:3–18. <https://doi.org/10.1016/j.swevo.2011.02.002>
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. pp 39–43.
<https://doi.org/10.1109/MHS.1995.494215>
- Floudas CA, Pardalos PM, Adjiman C (1999) *Handbook of test problems in local and global optimization*. Kluwer Academic Publishers
- Gharehchopogh FS, Gholizadeh H (2019) A comprehensive survey: whale optimization algorithm and its applications. *Swarm Evol Comput* 48:1–24. <https://doi.org/10.1016/j.swevo.2019.03.004>
- Graichen K, Petit N (2008) Constructive methods for initialization and handling mixed state-input constraints in optimal control. *J Guid Control Dyn* 31:1334–1343. <https://doi.org/10.2514/1.33870>
- Hashemi H, Mirjalili S (2018) A parallel numerical method for solving optimal control problems based on whale optimization algorithm. *Knowledge-Based Syst* 151:114–123.
<https://doi.org/10.1016/j.knsys.2018.03.024>
- He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng Appl Artif Intell* 20:89–99.
<https://doi.org/10.1016/j.engappai.2006.03.003>
- Heidari AA, Mirjalili S, Faris H et al (2019) Harris hawks optimization: algorithm and applications.

-
- Futur Gener Comput Syst 97:849–872. <https://doi.org/10.1016/j.future.2019.02.028>
- Jadon SS, Tiwari R, Sharma H et al (2017) Hybrid artificial bee colony algorithm with differential evolution. *Appl Soft Comput* 58:11–24. <https://doi.org/10.1016/j.asoc.2017.04.018>
- Jain M, Singh V, Rani A (2019) A novel nature-inspired algorithm for optimization: squirrel search algorithm. *Swarm Evol Comput* 44:148–175. <https://doi.org/10.1016/j.swevo.2018.02.013>
- Jee G, Sharma KK, Rao KK et al (2014) Evolution of attitude control law of an Indian re-entry launch vehicle. *Int J Adv Eng Sci Appl Math* 6:148–157
- Kang YL, Cheng L, Zhang QZ et al (2015) Data-driven RLV multi-objective reentry trajectory optimization based on new QABC algorithm. *Int J Adv Manuf Technol* 84:453–471. <https://doi.org/10.1007/s00170-015-8124-9>
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim* 39:459–471. <https://doi.org/10.1007/s10898-007-9149-x>
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220:671–680
- Li ZH, Hu C, Ding CB et al (2018) Stochastic gradient particle swarm optimization based entry trajectory rapid planning for hypersonic glide vehicles. *Aerosp Sci Technol* 76:176–186. <https://doi.org/10.1016/j.ast.2018.01.033>
- Liang W, Kun W, Wang Z et al (2019) Opposition-based multi-objective whale optimization algorithm with global grid ranking. *Neurocomputing* 341:41–59. <https://doi.org/10.1016/j.neucom.2019.02.054>
- Mafarja MM, Mirjalili S (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing* 260:302–312. <https://doi.org/10.1016/j.neucom.2017.04.053>
- Mahdavi S, Rahnamayan S, Deb K (2018) Opposition based learning: a literature review. *Swarm Evol Comput* 39:1–23. <https://doi.org/10.1016/j.swevo.2017.09.010>
- Mao Q, Dou L, Zong Q et al (2018) Attitude controller design for reusable launch vehicles during reentry phase via compound adaptive fuzzy H-infinity control. *Aerosp Sci Technol* 72:36–48. <https://doi.org/10.1016/j.ast.2017.10.012>
- Mernik M, Liu S, Karaboga D et al (2015) On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation. *Inf Sci* 291:115–127. <https://doi.org/10.1016/j.ins.2014.08.040>
- Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowledge-Based Syst* 89:228–249. <https://doi.org/10.1016/j.knosys.2015.07.006>
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27:495–513. <https://doi.org/10.1007/s00521-015-1870-7>
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Mustafi D, Sahoo G (2019) A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the k-means algorithm with applications in text clustering. *Soft Comput* 23:6361–6378. <https://doi.org/10.1007/s00500-018-3289-4>
- Nenavath H, Jatoth RK (2018) Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. *Appl Soft Comput* 62:1019–1043. <https://doi.org/10.1016/j.asoc.2017.09.039>

-
- Pontani M, Conway BA (2010) Particle swarm optimization applied to space trajectories. *J Guid Control Dyn* 33:1429–1441. <https://doi.org/10.2514/1.48475>
- Rao AV, Benson DA, Darby C et al (2010) Algorithm 902: GPOPS, a matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method. *ACM Trans Math Softw* 37:1–39. <https://doi.org/10.1145/1731022.1731032>
- Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179:2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- Ridha HM, Heidari AA, Wang M et al (2020) Boosted mutation-based harris hawks optimizer for parameters identification of single-diode solar cell models. *Energy Convers Manag* 209:112660. <https://doi.org/10.1016/j.enconman.2020.112660>
- Saha A, Bhattacharya A, Das P et al (2020) HSOS: a novel hybrid algorithm for solving the transient-stability-constrained OPF problem. *Soft Comput* 24:7481–7510. <https://doi.org/10.1007/s00500-019-04374-9>
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359. <https://doi.org/10.1023/A:1008202821328>
- Su ZK, Wang HL (2015) A novel robust hybrid gravitational search algorithm for reusable launch vehicle approach and landing trajectory optimization. *Neurocomputing* 162:116–127. <https://doi.org/10.1016/j.neucom.2015.03.063>
- Sun GJ, Lan YF, Zhao RQ (2019) Differential evolution with gaussian mutation and dynamic parameter adjustment. *Soft Comput* 23:1615–1642. <https://doi.org/10.1007/s00500-017-2885-z>
- Turgut MS, Sağban HM, Turgut OE et al (2021) Whale optimization and sine–cosine optimization algorithms with cellular topology for parameter identification of chaotic systems and Schottky barrier diode models. *Soft Comput* 25:1365–1409. <https://doi.org/10.1007/s00500-020-05227-6>
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82. <http://dx.doi.org/10.1109/4235.585893>
- Wu J, Wang H, Li N et al (2018) Path planning for solar-powered UAV in urban environment. *Neurocomputing* 275:2055–2065. <https://doi.org/10.1016/j.neucom.2017.10.037>
- Zhang DN, Liu Y (2011) RLV reentry trajectory optimization through hybridization of an improved GA and a SQP algorithm. In: *AIAA Guidance, Navigation, and Control Conference*. <https://doi.org/10.2514/6.2011-6658>
- Zhang HP, Wang HL, Li N et al (2020) Time-optimal memetic whale optimization algorithm for hypersonic vehicle reentry trajectory optimization with no-fly zones. *Neural Comput Appl* 32:2735–2749. <https://doi.org/10.1007/s00521-018-3764-y>

Figures

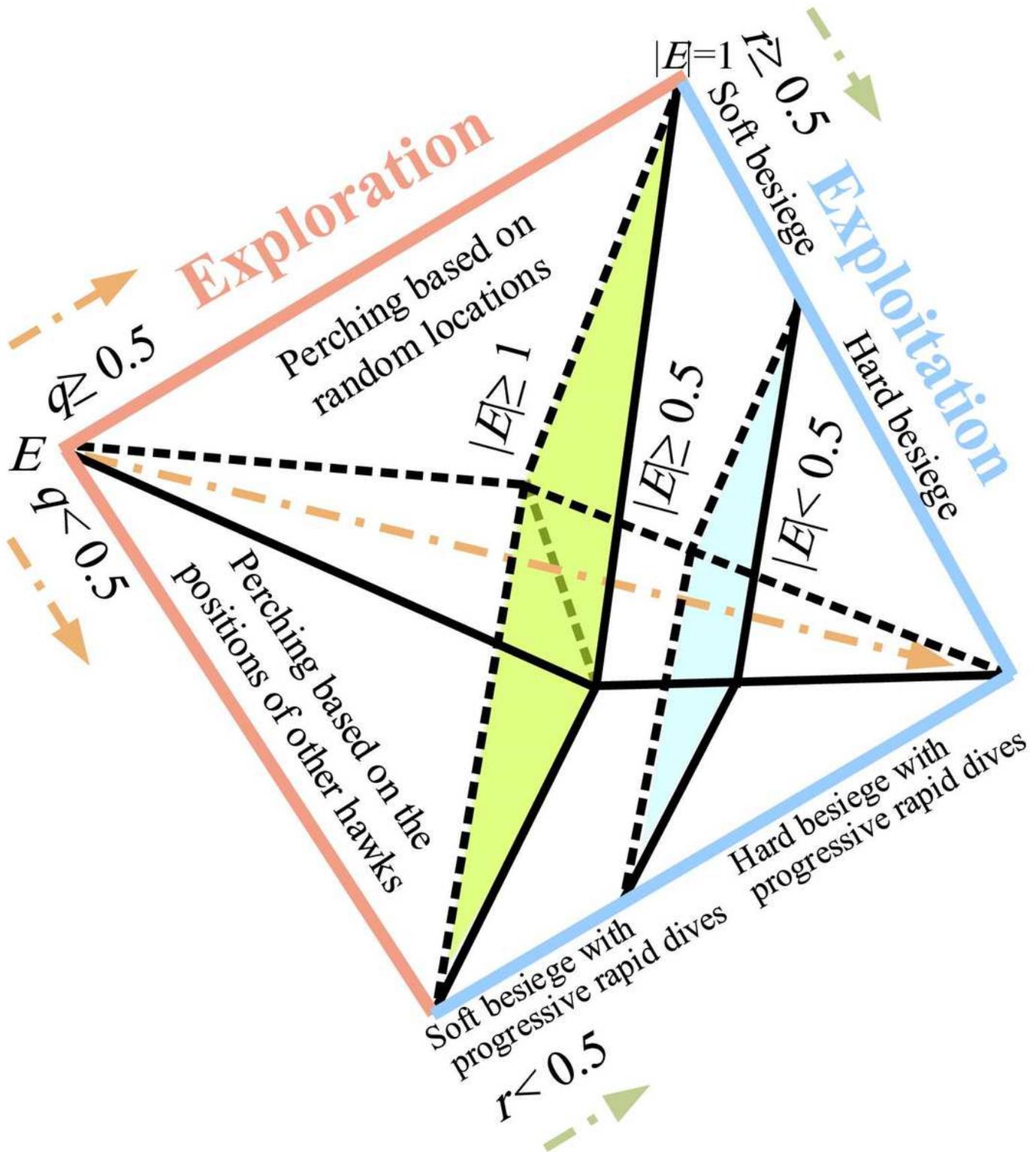


Figure 1

Different phases of harris hawks optimization

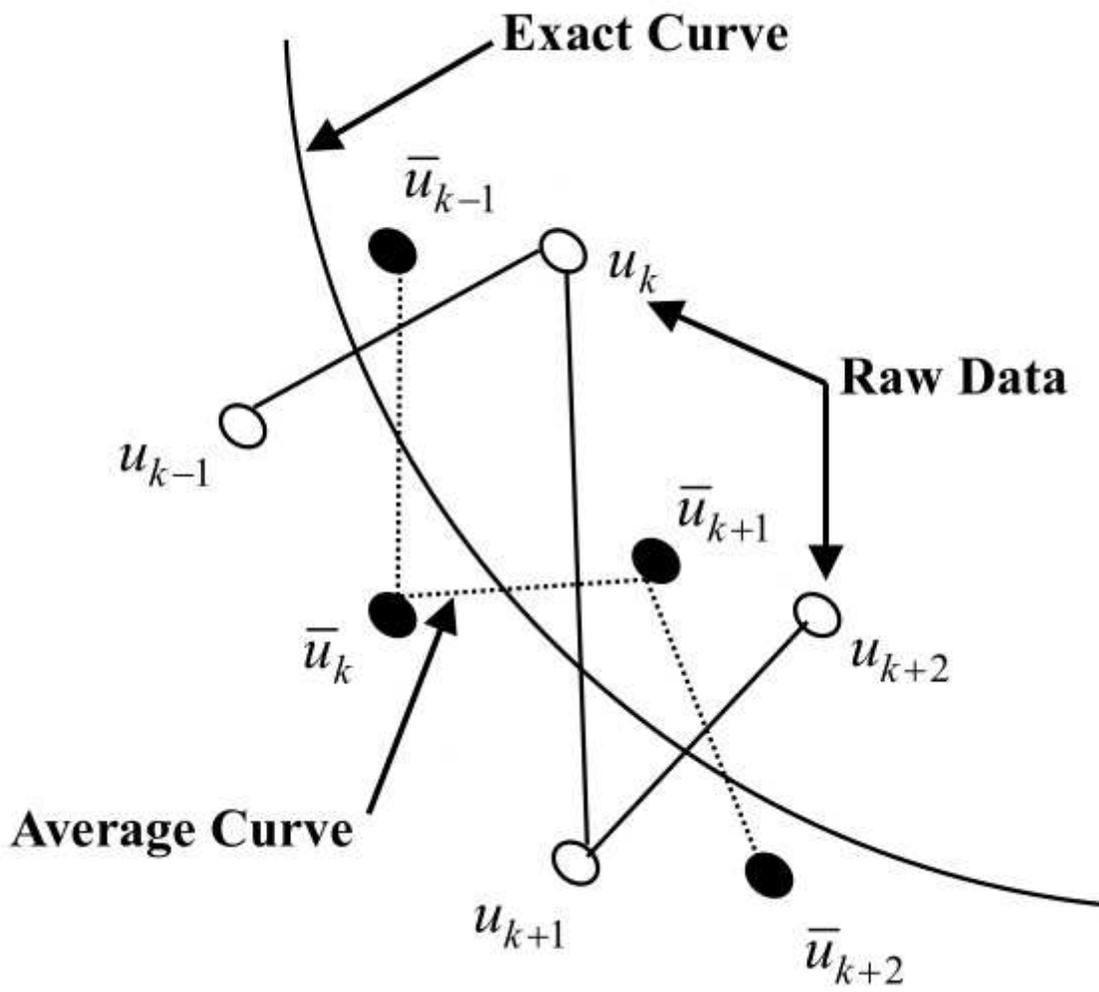


Figure 2

Application of smoothing technique in a control variable

Algorithm: Pseudo-code of the HPHHO algorithm

Input: The population size N , maximum number of iterations T

Output: The position of the rabbit and the corresponding fitness function value

Initialize the hawks population X_i ($i = 1, 2, \dots, N$)

while ($t < T$)

 Update chaotic sequence of r (t) by Eq. (25)

 Apply OBL to calculate the opposite population X_{oi} of all initial hawks by Eq. (24)

 Check if any new opposite population goes beyond the search space and amend it

 Calculate the fitness of each hawk in X_i and its corresponding opposite hawk X_{oi}

 Select the N fittest hawks as initial population in ascending order according to their fitness of X_i and X_{oi}

X^* = the best hawk

 Update hawks with smoothing technique by Eq. (26)

 % Apply DE algorithm to update the positions of the hawks

 for (each hawk of the first half ($i=1: N/2$))

 Apply DE mutation by the Eq. (20)

 Apply DE crossover by the Eq. (22)

 Apply DE selection by the Eq. (23)

 end for

 % Apply HHO algorithm to update the positions of the hawks

 for (each hawk of the rest half ($i=1+N/2: N$))

 Update the energy of the rabbit E using Eq. (9)

 if ($|E| = 1$) then % Exploration Stage

 Update the position using Eq. (27)

 else if ($|E| < 1$) then % Exploitation Stage

 if ($r = 0.5$ and $|E| = 0.5$) then

 Update the position using Eq. (10)

 else if ($r = 0.5$ and $|E| < 0.5$) then

 Update the position using Eq. (12)

 else if ($r < 0.5$ and $|E| = 0.5$) then % Soft besiege with progressive rapid dives

 Update the position using Eq. (16)

 else if ($r < 0.5$ and $|E| < 0.5$) then % Hard besiege with progressive rapid dives

 Update the position using Eq. (17)

 end if

 end if

 end for

 Merge the two updated subpopulations to generate a new population

 Check if any hawk goes beyond the search space and amend it

 Update X^* if there is a better solution

$t=t+1$

end while

return X^*

Figure 3

Pseudo-code of HPHHO

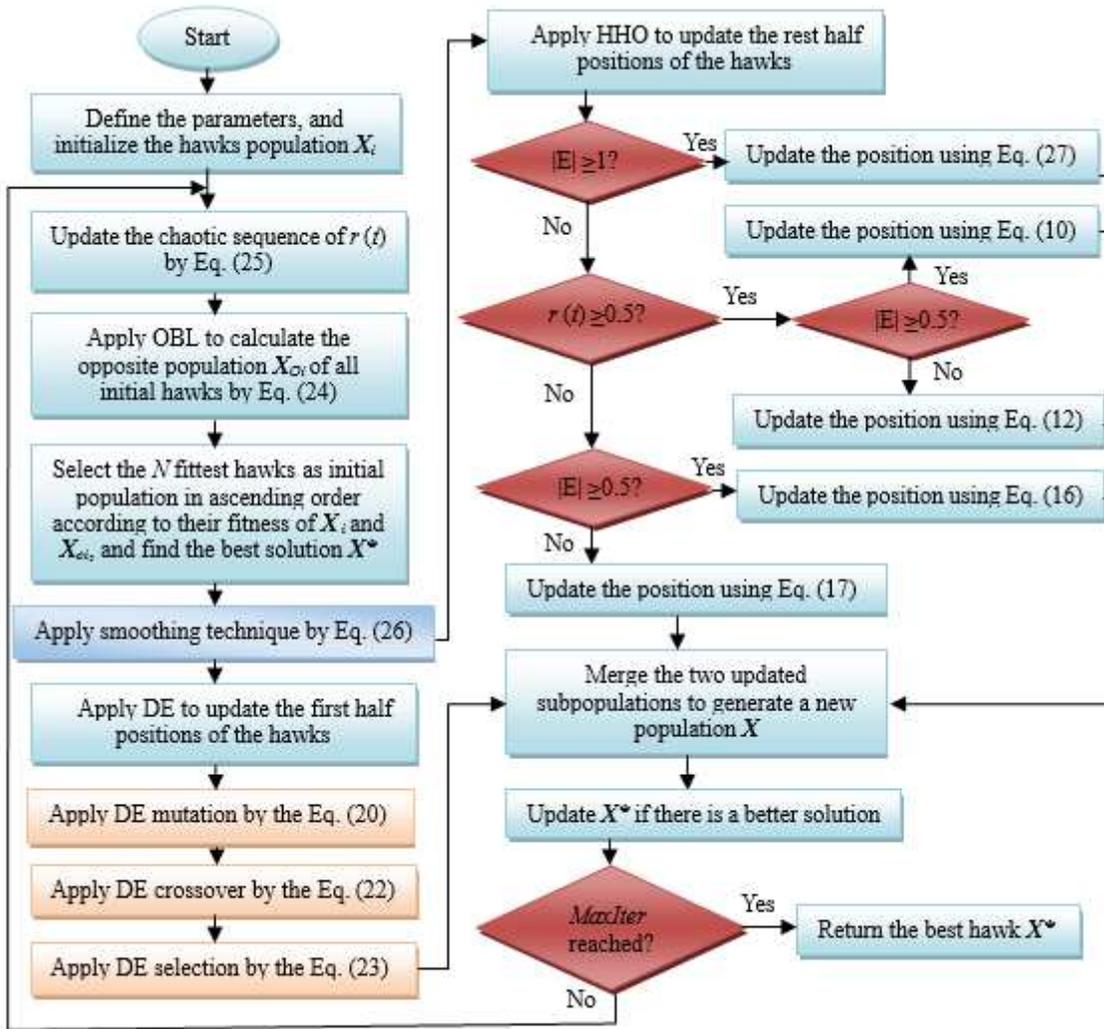


Figure 4

The flowchart of HPHHO

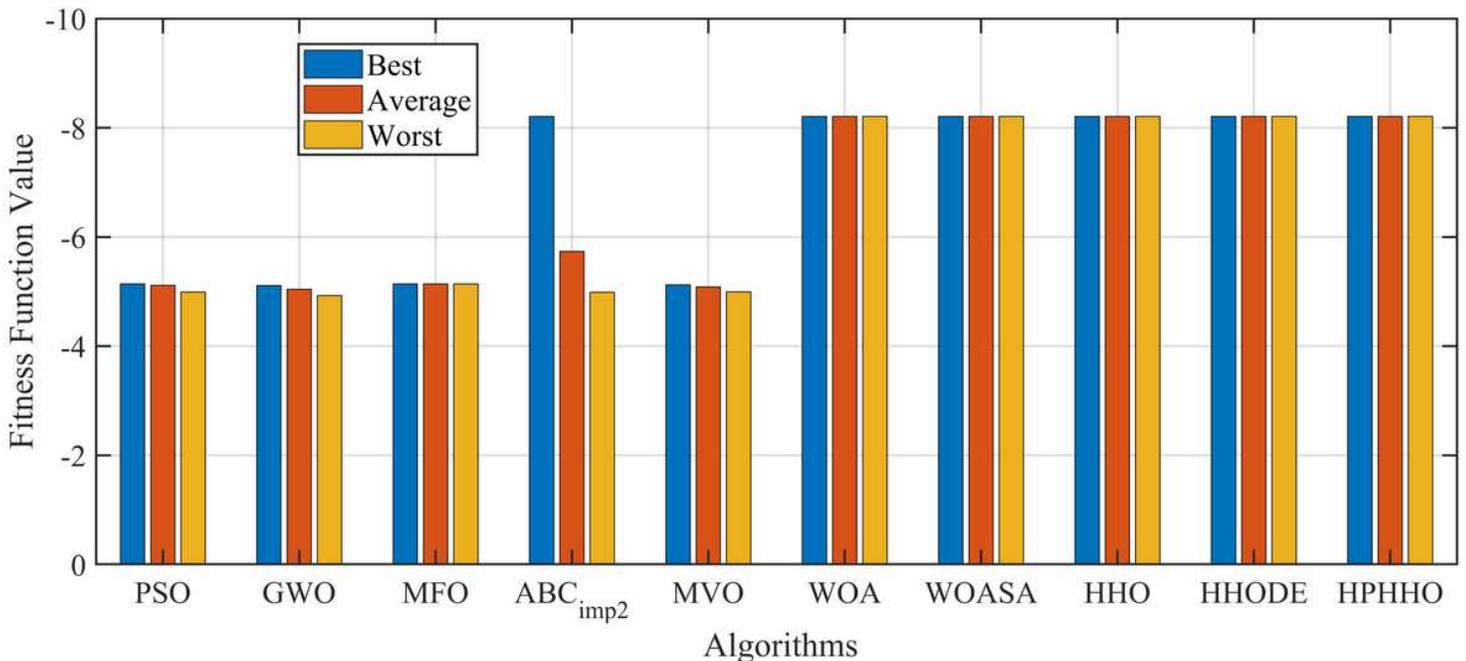


Figure 5

The experimental results of different algorithms on multiple solution example

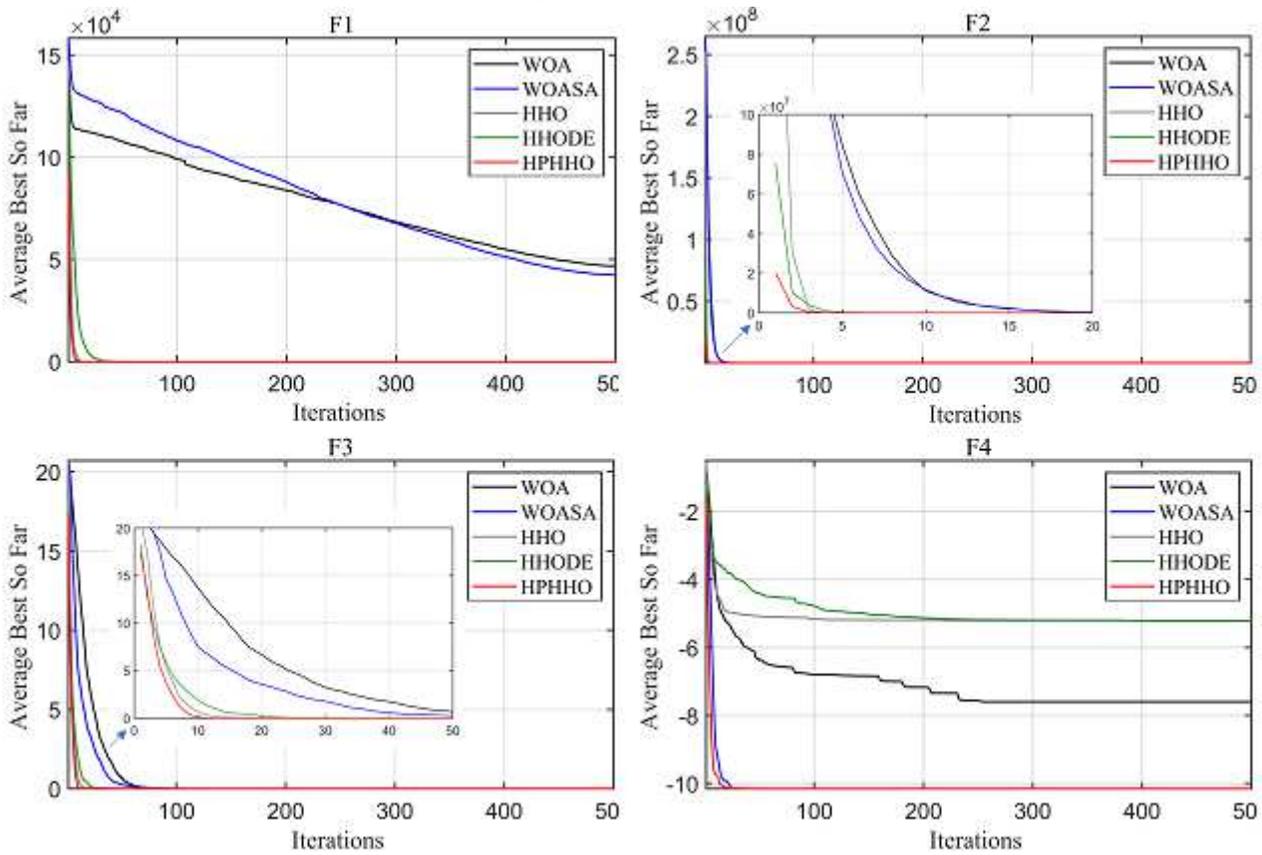


Figure 6

Convergence rate of HPHHO versus other optimizers

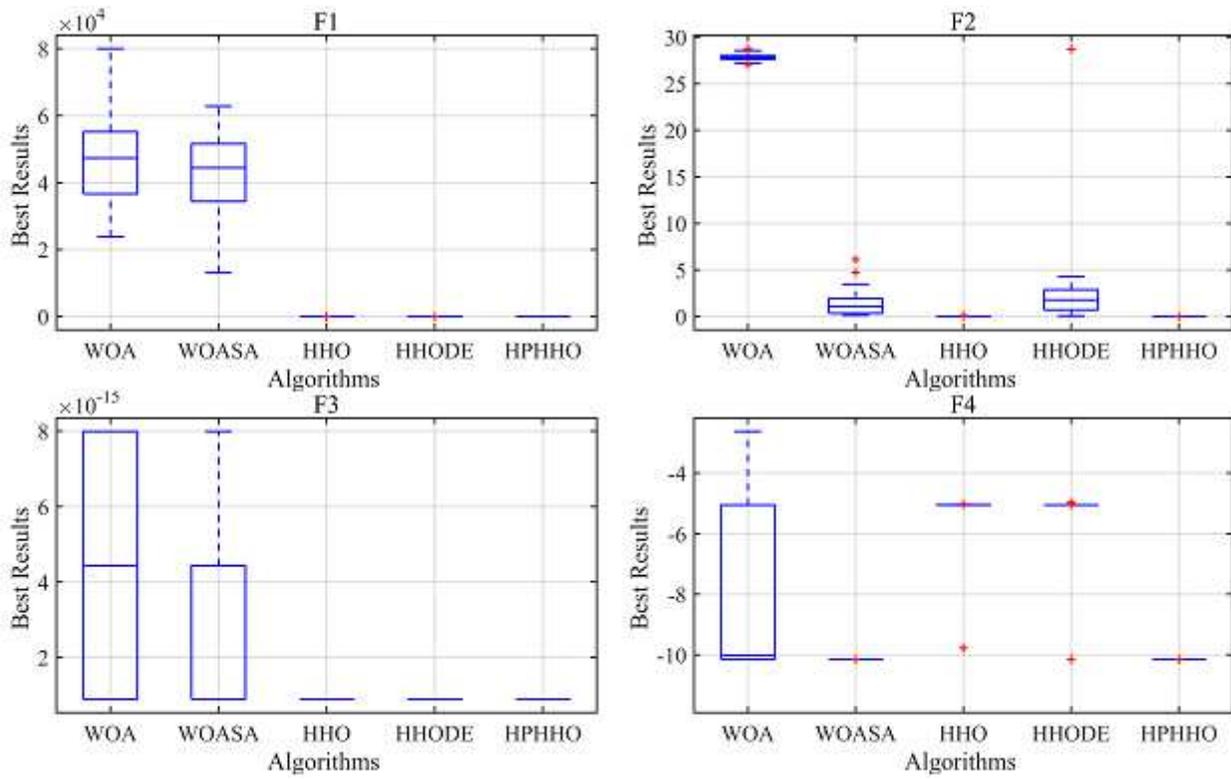


Figure 7

The boxplot of each method on four benchmark functions

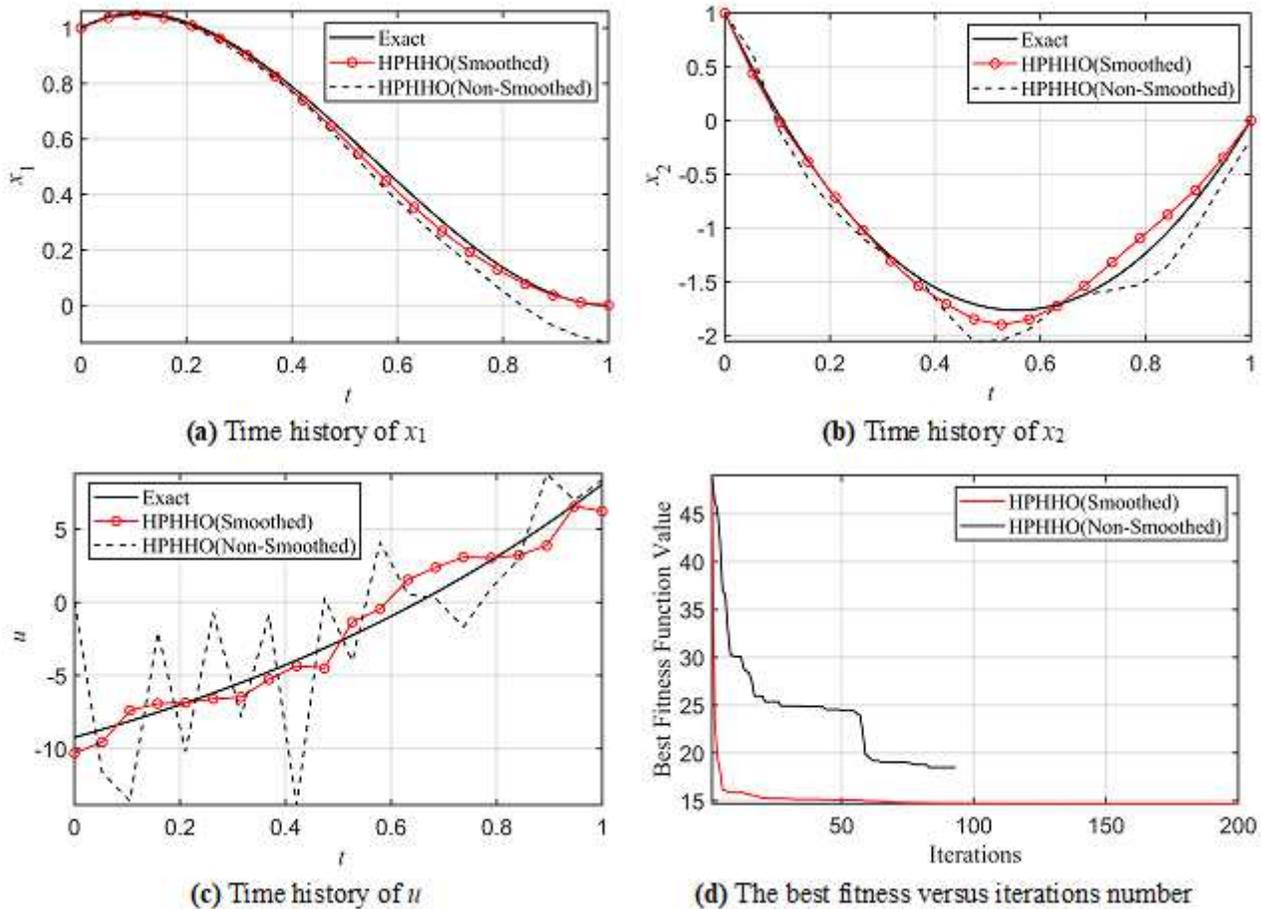


Figure 8

The exact and numerical solutions of the minimum energy solution

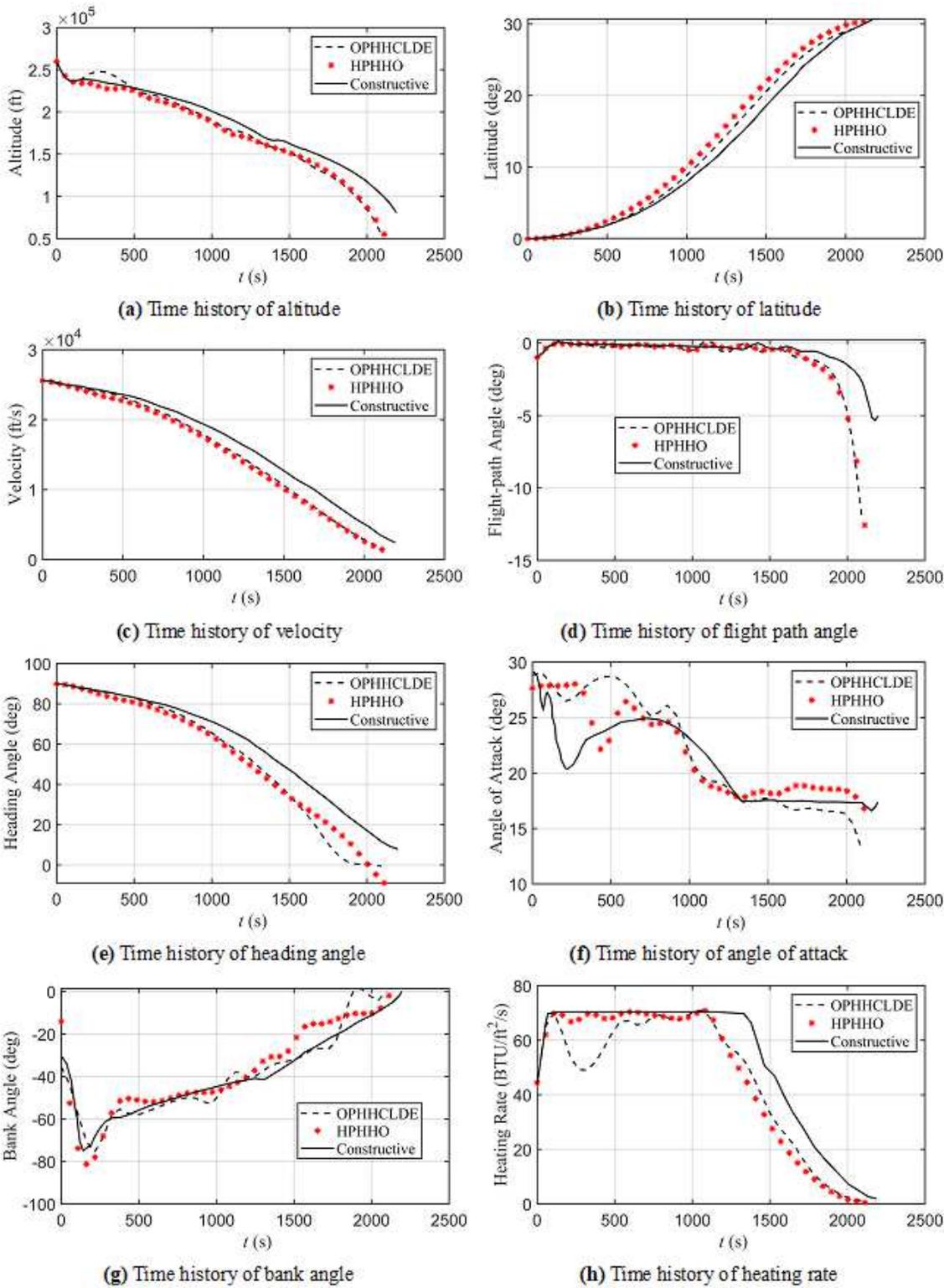
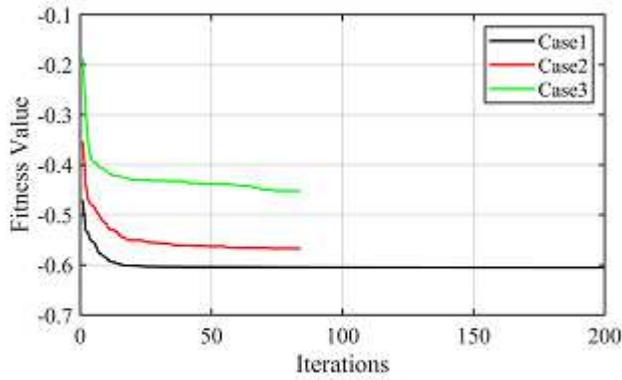
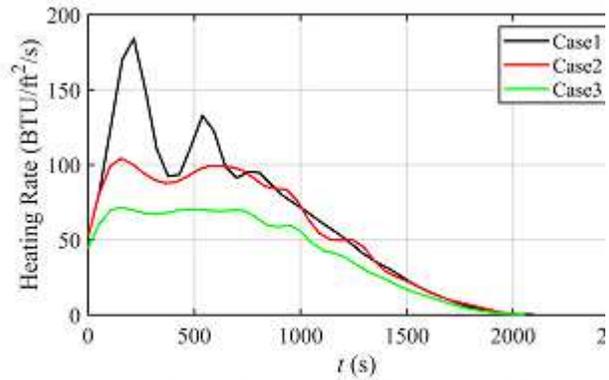


Figure 9

Optimal trajectories for the constrained reentry problem with heating rate constraint



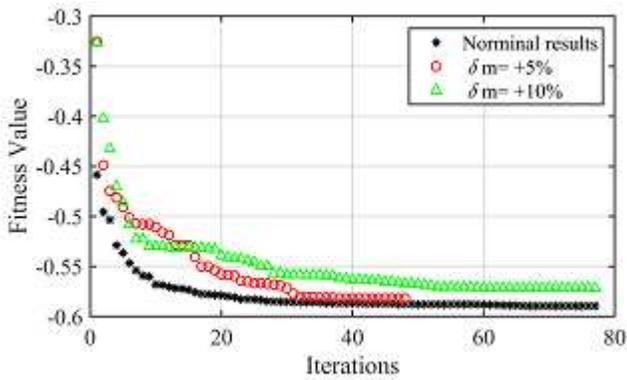
(a) Fitness value versus iterations



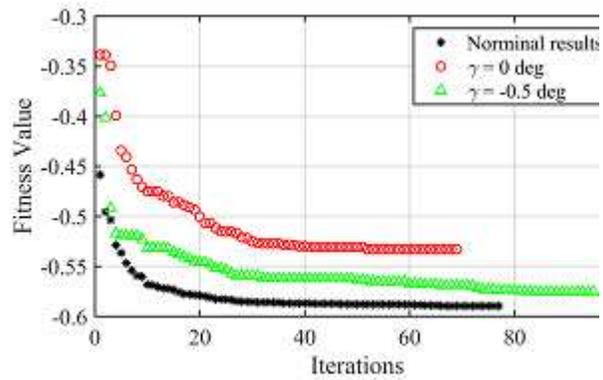
(b) Time history of heating rate

Figure 10

Results of HPHHO for three different cases



(a) Mass uncertainty



(b) Flight path angle uncertainty

Figure 11

Results of fitness value with uncertainty

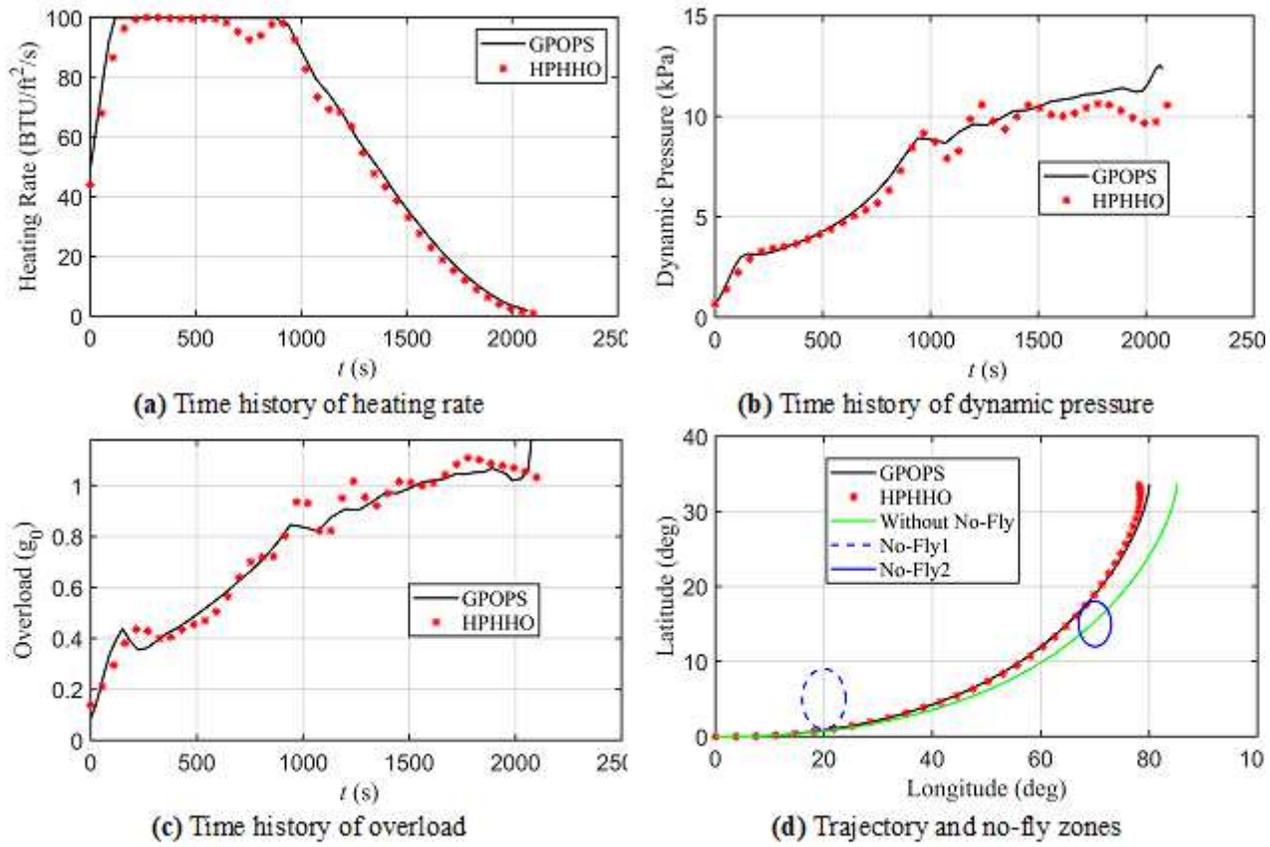


Figure 12

Optimal path trajectories with no-fly zones constraints of HPHHO and GPOPS

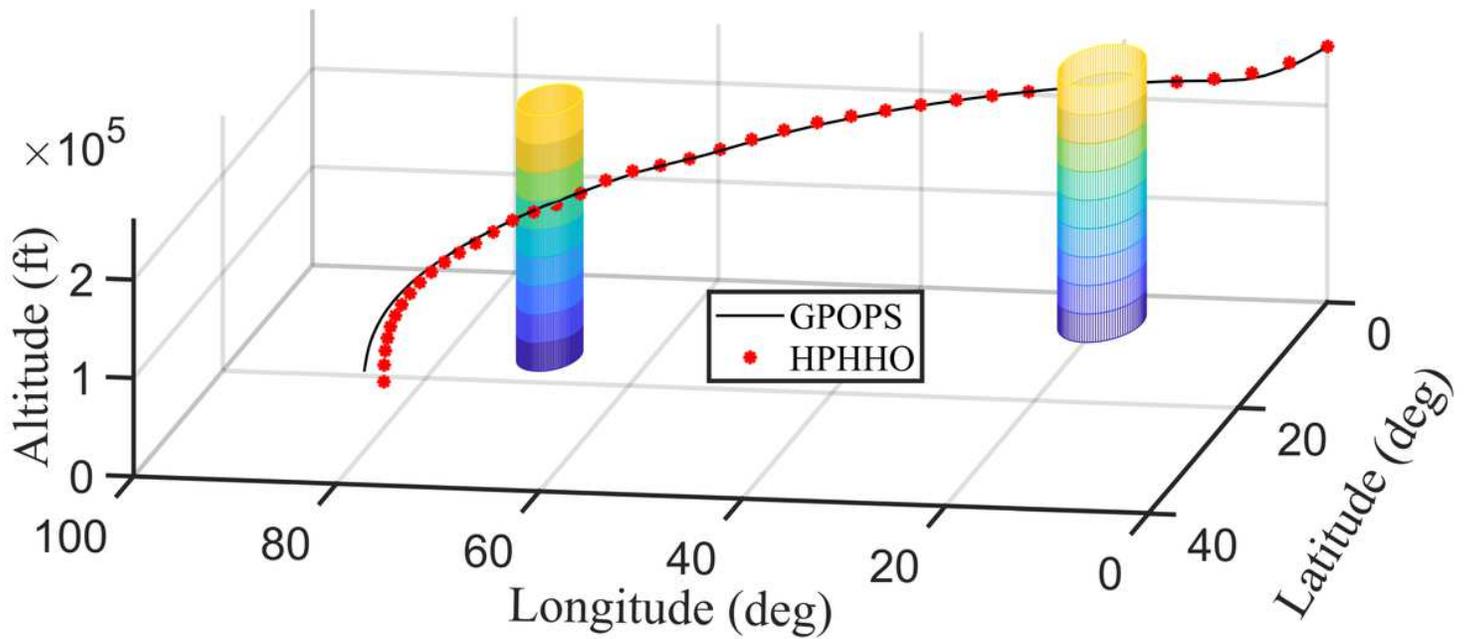


Figure 13

Three-dimensional optimal trajectory